Lab Manual

Data Structures using C

1. OBJECTIVE AND RELEVANCE

- To review basic computer systems concepts
- To be able to understand the different computing environments and their components
- To review the history of computer languages
- To be able to list and describe the classifications of computer languages
- To understand the steps in the development of a computer Program To review the system development life cycle
- To write Programs in C++ to solve the problems.
- To implement linear data structures such as Lists, Stacks, Queues.
- To implement non-linear data structures such asTrees and Graphs.
- To implement searching trees like Binary Search Trees, AVL Trees, B-Trees.
- To implement different types of sorting techniques
- To implement hash functions, collision resolution techniques.

2. OUTCOMES

- This subject contributes to developing student critical thinking through tutorial and lab exercises on solving problems. They will also practice more in written assignments, Programming exercises, and project.
- This subject contributes to having students practice their writing skills with project document and report writing.
- This subject allows te users to collect and organize the data in such a way that they can perform operations on these data in an effective way.
- This subject contributes to team work with group-based project for students to practice team spirit.

3. EQUIPMENT REQUIRED

Hardware

No. of System	:	60(IBM)
Processor	:	PIV™ 1.67 GH$_z$
RAM	:	512 MB
Hard Disk	:	40 GB
Mouse	:	Optical Mouse
Network Interface card	:	Present

Software

Operating System	:	Window XP
Software	:	Turbo C

4. CODE OF CONDUCT

- Students should report to the concerned lab as per the time table.
- Students who turn up late to the labs will in no case be permitted to do the program schedule for the day.
- After completion of the program, certification of the concerned staff in-charge in the observation book is necessary.
- Student should bring a notebook of 100 pages and should enter the readings/observations into the notebook while performing the experiment.
- The record of observations along with the detailed experimental procedure of the EXPERIMENTin the immediate last session should be submitted and certified staff member in-charge.
- Not more than 2-students in a group are permitted to perform the EXPERIMENTon the set.
- Any damage of the systmes will be viewed seriously either by putting penalty or by dismissing the total group of students from the lab for the semester/year.
- Students should be present in the labs for total scheduled duration

5. SYLLABUS

1. Basic C Programs – looping, data manipulations, arrays
2. Prgrams using strings – string function implementation
3. Programs using structures and pointers
4. Programs involving dynamic memory allocations
5. Array implementation of stacks and queues
6. Application of stacks and queues
7. Implementation of Trees and tree traversal
8. Implementation of BST
9. Implementation of linear and binary search
10. Implementation of insertion sort, bubble sort, quick sort and merge sort
11. Implementation of hash functions, Collision resolution techniques.

7. List of text book (OU Prescribed Text Books)

C++ AND DATA STRUCTURES

TEXT BOOKS

T1: Walter Savich, Problem Solving with C++, 6th Edition, Pears Education Publishing, 2009.

T2: Bjarne Stroustrup, The C++ Programming Language, 3rd, Edition, earson Education, 2013.

T3: Sartaj A Sahani, Data structures and Algoritms, Tata McGraw Hill, 3rd Edition, 2013.

REFERENCE BOOKS

R1: K.R. Venugopal, Rajkumar, T. Ravishankar, Masternig C++, Tata McGraw Hill, 2008.

R2: Yashavant Kanetkar, Data Structures through C++, BPB Publications, 1st Edition, 2008.

Ex.No.-1a ARMSTRONG NUMBER

AIM:

To write a C program for finding whether a given number is Armstrong number or not using loop control statement.

ALGORITHM:

```
1. Start the program.
2. Take number input from user store it in 'number'.
3. Assign 'temp' as number;
4. while temp!=0 do,
        remainder:=temp%10;
        sum:=sum+remainder*remainder*remainder;
        temp:=temp/10
   End while
5. if number==sum then,
        print number is Armstrong
   else
        print number is not armstrong
6. Stop the program.
```

PROGRAM:

```
/*Armstrong Number*/
#include <stdio.h>
int main()
{

    int n, n1, rem, num=0;
    printf("Enter a positive integer: ");
    scanf("%d", &n);
    n1= n;
    while(n1!=0)
    {
        rem=n1%10;

        num+=rem*rem*rem;
        n1/=10;
    }
    if(num==n)
        printf("%d is an Armstrong number.",n);
    else
        printf("%d is not an Armstrong number.",n);
}
```

OUTPUT:

```
Enter a positive integer: 371

371 is Armstrong number.
```

Ex.No.-1b MATRIX MULTIPLICATION

AIM:

To write a C program to compute matrix multiplication using the concept of arrays.

ALGORITHM:

1. Start the program.
2. To multiply two matrixes sufficient and necessary condition is "number of columns in matrix A = number of rows in matrix B".
3. Loop for each row in matrix A.
4. Loop for each columns in matrix B and initialize output matrix C to 0.
5. This loop will run for each rows of matrix A.
6. Loop for each columns in matrix A.
7. Multiply A[i,k] to B[k,j] and add this value to C[i,j]
8. Return output matrix C.
9. Stop the program.

PROGRAM:

```
/*Matrix Multiplication*/
#include<stdio.h>
#include<conioh>
int main()
{
    int a[10][10], b[10][10], c[10][10],m,n, i, j, k, l,g;
    printf("Enter the number of rows and column of first matrix\n");
    scanf("%d%d",&m,&n);
    printf("\nEnter the elements of first %dx%d matrix\n",m,n);
    for (i=0; i< m; i++)
    {
            for(j=0; j<n; j++)
            {
                    scanf("%d",&a[i][j]);
            }
    }
    s1 :
    printf("\nEnter the number of rows and column of second matrix\n");
    scanf("%d%d",&g,&l);
    if(n!=g)
    {
            printf("\nIn matrix multiplication first column and second row number should be the
            same \nRenter ");
            getch();
            goto s1;
        }
    printf("\nEnter the elements of second %dx%d matrix",n,l);
    for(i = 0; i <n; i++)
    {
        for (j = 0; j < l; j++)
        {
                scanf("%d", &b[i][j]);
        }
    }
    printf("\nThe first matrix is :-\n");
    for (i = 0; i < m; i++)
    {
        for (j = 0; j < n; j++)
        {
                printf("\t%d", a[i][j]);
        }
        printf("\n");
    }
    printf("\nThe second matrix is :-\n");
    for (i = 0; i < n; i++)
    {
        for (j = 0; j < l; j++)
        {
            printf("\t%d", b[i][j]);
        }
        printf("\n");
    }

    printf("\nMultiplication of the two matrices is as follows:\n");
    for (i = 0;i < m; i++)
    {
        printf("\n");
        for (j = 0; j < l; j++)
        {
                c[i][j]=0;
                for(k=0;k<n;k++)
                        c[i][j] = c[i][j]+a[i][k] * b[k][j];
                        printf("\t%d", c[i][j]);
        }
    }
    getch();
    }
```

OUTPUT:

```
Enter the number of rows and column of first matrix 2 2

Enter the elements of first 2x2 matrix 3 2 3 2


Enter the number of rows and column of second matrix 2 2


Enter the elements of second 2x2 matrix 4 1 4 1

The first matrix is :-
        3    2
        3    2

The second matrix is :-
        4    1
        4    1

Multiplication of the two matrices is as follows:

      20     5
      20     5
```

Ex.No.-2 STRING FUNCTIONS

AIM:

To write a C program to illustrate the concept of string handling functions.

ALGORITHM:

1. Start the program.
2. Get two strings as input.
3. Use the string compare function for checking the two strings are equal.
4. Use the string concatenation function for joining first string with second string.
5. Find the length of the strings using the strlen function.
6. Print the reverse of a string using strrev function.
7. Show the upper case and lower case of a string using strupr and strlwr functions.
8. Stop the program.

PROGRAM:

```
/*String Handling*/
#include <string.h>
void main()
{
    char s1[20], s2[20], s3[20];
    int x, l1, l2, l3;
    clrscr();
    printf("\n\nEnter two string constants: \n");
    scanf("%s %s", s1, s2);
    x = strcmp(s1, s2);
     if(x != 0)
    {
        printf("\n\nStrings are not equal \n");
        strcat(s1, s2);
    }
    else
        printf("\n\nStrings are equal \n");

    strcpy(s3, s1);

    l1 = strlen(s1);
    l2 = strlen(s2);
    l3 = strlen(s3);
    printf("\ns1 = %s\t length = %d characters\n", s1, l1);
    printf("s2 = %s\t length = %d characters\n", s2, l2);
    printf("s3 = %s\t length = %d characters\n", s3, l3);
    printf("Reverse of S1 is %s\n",strrev(s1));
    printf("Uppercase of S2 is %s\n",strupr(s2));
    printf("Lowercase of S2 is %s\n",strlwr(s2));
}
```

OUTPUT:

```
Enter two string constants:
Sachein
Ashwath

Strings are not equal

 s1 = SacheinAshwath      length = 14 characters
 s2 = Ashwath  length = 7 characters
 s3 = SacheinAshwath      length = 14 characters

    Reverse of S1 is htawhsAniehcaS
    Uppercase of S2 is ASHWATH
    Lowercase of S2 is ashwath
```

Ex.No.-3a EMPLOYEE DETAILS

AIM:

To write a C program to store employee details using the concept of structures.

ALGORITHM:

1. Start the program.
2. Create a structure details and declare its members.
3. Get the values for the structure members as input.
4. Display them in output.
5. Stop the program.

PROGRAM:

```
#include  <stdio.h>
#include <conio.h>
struct details
{
    char name[30];
    int age;
    char address[500];
    float salary;
};

int main( )
{
    struct details detail;
    clrscr( );
    printf("\nEnter name:\n");
    scanf("%s",detail.name);
    printf("\nEnter age:\n");
    scanf("%d",&detail.age);
    printf("\nEnter Address:\n");
    scanf("%s",detail.address);
    printf("\nEnter Salary:\n");
    scanf("%f",&detail.salary);
    printf("\n\n\n");
    printf("Name of the Employee : %s \n",detail.name);    printf("Age
    of the Employee : %d \n",detail.age); printf("Address of the
    Employee : %s \n",detail.address); printf("Salary of the Employee :
    %f \n",detail.salary); getch( );
}
```

OUTPUT:

```
Enter name: Sachein Enter age:
21
Enter Address: Perambalur Enter
Salary: 28000

Name of the Employee : Sachein
Age of the Employee : 21
Address of the Employee : Ariyalur
Salary of the Employee : 28000.00
```

Ex.No.-3b ADDING OF TWO NUMBERS

AIM:

To write a C program for adding the two numbers using pointers.

ALGORITHM:

1. Start the program.
2. Declare the variable p and q as pointers.
3. Read the values of two pointer variables p and q.
4. Calculate the sum as p and q.
5. Display the sum.
6. Stop the program.

PROGRAM:

```
#include <stdio.h>
int main()
{
    int first, second, *p, *q, sum;
    printf("\n Enter two integers to add : ");
    scanf("%d%d", &first, &second);
    p =  &first;
   q = &second;
    sum = *p + *q;
    printf("Sum of the numbers = %d\n", sum);
    return 0;
}
```

OUTPUT:

```
Enter two integers to add : 5 7
Sum of the numbers = 12
```

Ex.No.-4 LARGEST ELEMENT IN ARRAY

AIM:

To write a C program to find the largest element using dynamic memory allocation.

ALGORITHM:

1. Create an initial array with some elements in it.
2. Allocate memory for the elements using calloc function.
3. Check first element is greater than the second element of the array using pointers and so on.
4. Print the largest element.
5. Stop the program.

PROGRAM:

/*Largest Element in Array*/

```
#include  <stdio.h>
#include <stdlib.h>
int main( )
{

        int  i,n;
        float *data;

        printf("Enter total number of elements(1 to 100): ");
        scanf("%d",&n);
        data=(float*)calloc(n,sizeof(float));
        if(data==NULL)
        {
                printf("Error!!! memory not allocated.");
                exit(0);
        }
        printf("\n");
        for(i=0;i<n;++i)
        {
                printf("Enter Number %d: ",i+1);
                scanf("%f",data+i);
        }
        for(i=1;i<n;++i)
        {
                if(*data<*(data+i))
                        *data=*(data+i);
        }

        printf("Largest element = %.2f",*data);
        return 0;
}
```

OUTPUT:

```
Enter the total number of elements(1 to 100): 5

Enter Number 1: 10.5
Enter Number 2: 12.6
Enter Number 3: 39.78
Enter Number 4: 9.6
Enter Number 5: 42.5
Largest element = 42.50
```

Ex.No.-5a STACK – ARRAY IMPLEMENTATION

AIM:

To write a C program to implement the stack using arrays.

ALGORITHM:

(i) Push Operation:

- To push an element into the stack, check whether the top of the stack is greater than or equal to the maximum size of the stack.
- If so, then return stack is full and element cannot be pushed into the stack.
- Else, increment the top by one and push the new element in the new position of top.

(ii) Pop Operation:

- To pop an element from the stack, check whether the top of the stack is equal to -1.
- If so, then return stack is empty and element cannot be popped from the stack.
- Else, decrement the top by one.

PROGRAM:

/*Stack Implementation using arrays*/

```
# include <stdio.h>
# include <conio.h>
# include <stdlib.h>
# define size 5
struct stack
{
        int s[size]; int top;
}st;

int stfull()
{
   if(st.top>=size-1)
         return 1;
   else return 0;
}

void push(int item)
{
         st.top++;
         st.s[st.top] =item;
}
int stempty()
{
 if(st.top==-1)
   return 1;
 else return 0;
}

int pop()
{
            int item; item=st.s[st.top];
            st.top--;
            return(item);
 }

void display()
{
    int i;
    if(stempty())
            printf("Stack Is Empty!\n");
  else
  {
            for(i=st.top;i>=0;i--)
                        printf("\n%d",st.s[i]);
  }
}
void main(void)
{
            int item,ch;
            char ans;
            st.top=-1;
            clrscr();
            printf("<-----Stack using Array---- >\n"); while(1)
            {
                       printf("\n1.Push\n2.Pop\n3.Display\n4.exit\n"); printf("Enter
                       Your Choice:\n"); scanf("%d",&ch);
                       switch(ch)
                       {
                                   case 1:
                                               printf("Enter The item to be pushed:\n"); scanf("%d",&item);
                                               if(stfull())
                                                       printf("Stack is Full!\n"); else
                                                       push(item);
                                               break;
                                    case 2:
                                               if(stempty())
                                                           printf("Empty stack!\n");
                                               else
                                               {
                                                           item=pop();
                                                           printf("The popped element is %d\n",item);
```

```
                                        }
        break;
        case 3:
                                        display();
                                        break;
                              case 4:
                                        exit(0);
                    }
            }
            getch();
}
```

OUTPUT:

```
<-----Stack using Array---- >

1.Push
2.Pop
3.Display
4.exit
Enter Your Choice: 1
Enter The item to be pushed: 10

1.Push
2.Pop
3.Display
4.exit
Enter Your Choice: 1
Enter The item to be pushed: 20

1.Push
2.Pop
3.Display
4.exit
Enter Your Choice: 3

20
10

1.Push
2.Pop
3.Display
4.exit
Enter Your Choice: 2
The popped element is 20
```

```
1.Push
2.Pop
3.Display
4.exit
Enter Your Choice: 4
```

Ex.No.-5b QUEUE – ARRAY IMPLEMENTATION

AIM:

To write a C program to implement the queue using arrays.

ALGORITHM:

Enqueue:

1. Check if queue is not full.
2. If it is full then return queue overflow and item cannot be inserted.
3. If not, check if rear value is -1, if so then increment rear and by 1; if not increment front by 1.
4. Store the item in the new value of front.

Dequeue:

1. Check if queue is not empty.
2. If it is empty, return queue underflow and dequeue operation cannot be done.
3. If queue is not empty , check if rear and front are equal.
    - If so assign -1 to front and rear.
    - If not decrement front by 1.

PROGRAM:

```
/*Queue using Array*/
#include<stdio.h>
#include<conio.h>
#define MAX 10
int queue[MAX],front=-1,rear=-1;

void insert_element();
void    delete_element();    void
display_queue();

int main()
{
        int option;
        printf(">>> c program to implement queue operations <<<");
        do
        {
                printf("\n\n 1.Enqueue an element"); printf("\n
                2.Dequeue an element");
                printf("\n 3.Display queue");
                printf("\n 4.Exit");
                printf("\n Enter your choice: ");
                scanf("%d",&option);
                switch(option)
                {
                  case 1:
                          insert_element();
                          break;
                  case 2:
                          delete_element(); break;
                  case 3:
                          display_queue();
                          break;
                  case 4:
                          return 0;
            }
        }while(option!=4);
}

void insert_element()
{
        int num;
        printf("\n Enter the number to be Enqueued: "); scanf("%d",&num);
        if(front==0 && rear==MAX-1)
                 printf("\n Queue OverFlow Occured");
        else if(front==-1&&rear==-1)
        {
                front=rear=0;
                queue[rear]=num;
        }
        else if(rear==MAX-1 && front!=0)
        {
                rear=0;
                queue[rear]=num;
        }
        else
        {
                rear++;
                queue[rear]=num;
        }
}

void delete_element()
{
        int element;
        if(front==-1)
        {
                printf("\n Underflow");
        }
        element=queue[front];
        if(front==rear)
                front=rear=-1;
        else
        {
                if(front==MAX-1)
                        front=0;
                else front++;
                printf("\n The dequeued element is: %d",element);
        }

}

void display_queue()
{
        int i;
        if(front==-1)
              printf("\n No elements to display");
        else
        {
            printf("\n The queue elements are:\n ");
            for(i=front;i<=rear;i++)
            {
                printf("\t %d",queue[i]);
           }
   }
}
```

OUTPUT:

```
>>> c program to implement queue operations <<<

 1.Enqueue an element
 2.Dequeue an element
 3.Display queue
 4.Exit
 Enter your choice: 1

 Enter the number to be Enqueued: 10


 1.Enqueue an element
 2.Dequeue an element
 3.Display queue
```

```
4.Exit
Enter your choice: 1

Enter the number to be Enqueued: 20

1.Enqueue an element
2.Dequeue an element
3.Display queue
4.Exit
Enter your choice: 3

The queue elements are: 10 20

1.Enqueue an element
2.Dequeue an element
3.Display queue
4.Exit
Enter your choice: 2

The dequeued element is: 10

1.Enqueue an element
2.Dequeue an element
3.Display queue
4.Exit
Enter your choice: 4
```

Ex.No.-6a STACK – LINKED LIST IMPLEMENTATION

AIM:

To write a C program to implement the stack using linked list.

ALGORITHM:

A) Push Operation:

1. To push an element into the stack, copy the element to be inserted in the data field of the new node.
2. Assign the reference field of the new node as NULL.
3. Check if top is not NULL, if so, then assign the value of top in the reference field of new node.
4. Assign the address of the new node to the top.

B) Pop Operation:

1. To pop an element from the stack, check whether the top of the stack is NULL.
2. If so, then return stack is empty and element cannot be popped from the stack.
3. Else, assign the top value to a temporary node.
4. Now assign the value in the reference field of the node pointed by top to the top value.
5. Return the value in the data field of the temporary node as the element deleted and delete the temporary node.

PROGRAM:

/*Stack Implementation using Linked List*/

```
#include<stdio.h>
void push();
void pop();
void display();
main()
{
        int n;
        printf("STACK USING LINKED LIST\n1.PUSH\n2.POP\n3.DISPLAY\n4.
        EXIT\n");
        do
        {
                printf("\nEnter your choice\n"); scanf("%d",&n);
                switch(n)
                {
                        case 1:

push();
                         break;
                        case 2:
                          pop();
                        break;
                        case 3:

                        display(
);
                          break;
                        case 4:
                          break;
                        default:
                          printf("Invalid choice\n");
                          break;
                }
        }while(n!=4);
}

typedef struct node
{
        int data;
        struct node *link;
}n;
n *top=NULL;

void push()
{
        int item; n *temp;
        printf("Enter the item\n");
        scanf("%d",&item);
        temp=(n*)malloc(sizeof(n));
        temp->data=item;
        temp->link=top;
        top=temp;
}

void pop()
{
        n *temp;
        if(top==NULL)
                printf("Stack is empty\n"); else
        {
                temp=top;
                printf("The element deleted = %d\n",temp->data); free(temp);
                top=top->link;
        }
}
void display()
{
        n *save;
        if(top==NULL)
                printf("Stack is empty\n");
        else
        {
                save=top;
                printf("The elements of the stack are :");
                while(save!=NULL)
                {
                        printf("%d\t",save->data);
                        save=save->link;
                }
                printf("\nTopmost element = %d\n",top->data);
        }
}
```

OUTPUT:

```
STACK USING LINKED LIST
1.PUSH
2.POP
3.DISPLAY
4.EXIT

Enter your choice 1
Enter the item 10

Enter your choice 1
Enter the item 20

Enter your choice 3
The elements of the stack are :20 10
Topmost element = 20

Enter your choice 2
The element deleted = 20
Enter your choice 4
```

Ex.No.-6b QUEUE – LINKED LIST IMPLEMENTATION

AIM:

To write a C program to implement the queue using linked list.

ALGORITHM:

Enqueue:

1. Create a new node and allocate memory space for the new node.
2. Assign the element to be inserted in the data field of the new node.
3. Assign NULL to the address field of the new node.
4. Check if rear and front pointers are NULL.
5. If so, then make the front and rear pointers to point to new node.
6. If not, then assign address of the new node as the rear pointer value.

Dequeue:

1. Check if queue is not empty.
2. If it is empty, return queue underflow and dequeue operation cannot be done.
3. If not, assign the front->next value as the new front pointer and free the deleted node.

PROGRAM:

```
//Queue using linked list
#include<stdio.h>
#include<conio.h>
 #include<stdlib.h> struct
 node
 {
       int data;
       struct node* next; }*rear,
 *front;
```

```
void dequeue()
{
       struct node *temp, *var=rear;
       if(var==rear)
      {
             rear = rear->next;
             free(var);
      }
      else
            printf("\nQueue Empty");
}


void enqueue(int value)
{
     struct node *temp;
     temp=(struct node *)malloc(sizeof(struct node));
     temp->data=value;
     if (front == NULL)
     {
           front=temp; front-
           >next=NULL; rear=front;
     }
     else
     {
           front->next=temp; front=temp;
           front->next=NULL;
     }
}

void display()
{
     struct node *var=rear;
     if(var!=NULL)
     {
           printf("\nElements in Queue: ");
           while(var!=NULL)
           {
                printf("\t%d",var->data);
                var=var->next;
           }
           printf("\n");
     }
     else
            printf("\nQueue is Empty");
}
```

```
int main()
{
    int ch; clrscr();
    front=NULL;
    printf("<-----Queue using Linked List ---- >\n"); printf("
    \n1. Enqueue an element");
    printf(" \n2. Dequeue an element");
    printf(" \n3. Display Queue");
    printf(" \n4. Exit\n");
    while(1)
    {
        printf(" \nEnter your choice: ");
        scanf("%d",&ch);
        switch(ch)
        {
             case 1:
             {
                 int value;
                 printf("\nEnter a value to Enqueue: ");
                 scanf("%d",&value);
                 enqueue(value);
                 display();
                 break;
             }
             case 2:
             {
                 dequeue();
                 display();
                 break;
             }
             case 3:
             {
                 display();
                 break;
             }
             case 4:
             {
                 exit(0);
             }
             default:
             {
                 printf("\nwrong choice for operation");
             }
        }
    }
}
```

OUTPUT:

```
<-----Queue using Linked List----->

1. Enqueue an element
2. Dequeue an element
3. Display Queue
4. Exit

Enter your choice: 1

Enter a value to Enqueue: 10

Elements in Queue:      10

Enter your choice: 1

Enter a value to Enqueue: 20

Elements in Queue:      10      20

Enter your choice: 1

Enter a value to Enqueue: 30

Elements in Queue:      10      20     30

Enter your choice: 2

Elements in Queue:      20      30

Enter your choice: 4
```

.

Ex.No.-7a INFIX TO POSTFIX CONVERSION

AIM:

To write a C program to perform infix to postfix conversion using stack.

ALGORITHM:

```
1. Define a stack
2. Go through each character in the string
3. If it is between 0 to 9, append it to output string.
4. If it is left brace push to stack
5. If it is operator *+-/ then
    a. If the stack is empty push it to the stack
    b. If the stack is not empty then start a loop:
        i. If the top of the stack has higher precedence
        ii. Then pop and append to output string
        iii. Else break
        iv. Push to the stack
6. If it is right brace then
    a. While stack not empty and top not equal to left brace
    b. Pop from stack and append to output string
    c. Finally pop out the left brace.
7. If there is any input in the stack pop and append to the output string.
```

PROGRAM:

/*Stack application - Infix to Postfix Conversion*/

```
# define SIZE 50
# include <ctype.h>

char s[SIZE];
int top = -1; push(char
elem)
{
        s[++top] = elem;
}
```

```
char pop()
{
        return (s[top--]);
}

int pr(char elem)
{
        switch (elem)
        {
                case '#':
                        return 0;
                case '(':
                        return 1;
                 case '+':
                 case '-':
                        return 2;
                 case '*':
                 case '/':
                        return 3;
        }
}

main()
{
        char infx[50], pofx[50], ch, elem;
        int i = 0, k = 0;
        printf("<-----Stack Application: Infix to Postfix Conversion ---- >\n");
        printf("\n\nRead the Infix Expression ? ");
        scanf("%s", infx);
        push('#');
        while ((ch = infx[i++]) != '\0')
        {
              if (ch == '(')
                 push(ch);
               else if (isalnum(ch))
                pofx[k++] = ch;
        else if (ch == ' )' )
        {
                        while (s[top] != ' ( ' )
                                pofx[k++] = pop();
                                elem = pop();
          }
                else
                {
                        while (pr(s[top]) >= pr(ch))
                                pofx[k++] = pop();
                                push(ch);
                }
        }
        while (s[top] != '#')
                pofx[k++] = pop();
                pofx[k] = '\0';
                printf("\n\nGiven Infix Expn: %s Postfix Expn: %s\n", infx, pofx);
}
```

OUTPUT:

```
<-----Stack Application: Infix to Postfix Conversion ---->

Read the Infix Expression ? a+b*c-d

Given Infix Expn: a+b*c-d Postfix Expn: abc*+d-
```

Ex.No.-7b EVALUATING POSTFIX EXPRESSION

AIM:

To write a C program to evaluate postfix expression using stack.

ALGORITHM:

1. Start the program.
2. Scan the Postfix string from left to right.
3. Initialise an empty stack.
4. If the scannned character is an operand, add it to the stack. If the scanned character is an operator, there will be atleast two operands in the stack.
5. If the scanned character is an Operator, then we store the top most element of the stack(topStack) in a variable temp. Pop the stack. Now evaluate topStack(Operator)temp. Pop the stack and Push result into the stack.
6. Repeat this step till all the characters are scanned.
7. After all characters are scanned, we will have only one element in the stack. Return topStack.
8. Stop the program.

PROGRAM:

```
#define SIZE 50
#include <ctype.h>

int s[SIZE];
int top=-1;

push(int elem)
{
        s[++top]=elem;
}

int pop()
{
        return(s[top--]);
}
 main()
{
        char pofx[50],ch;
        int i=0,op1,op2;
        printf("<-----Stack Application: Evaluating Postfix Expression ---->\n"); printf("\n\nRead the
          Postfix Expression ? ");
          scanf("%s",pofx);
          while( (ch=pofx[i++]) != '\0')
          {
                    if(isdigit(ch))
                    push(ch-'0');
                    else
                    {
                              op2=pop();
                              op1=pop();
                              switch(ch)
                              {
                                        case '+':
                                                  push(op1+op2);
                                                  break;
                                        case '-':
                                                  push(op1-op2);
                                                  break;
                                        case '*':
                                                  push(op1*op2);
                                                  break;
                                        case '/':
                                                  push(op1/op2);
                                                  break;
                              }
                    }
          }
          printf("\n Given Postfix Expn: %s\n",pofx);
          printf("\n Result after Evaluation: %d\n",s[top]);
}
```

OUTPUT:

```
<-----Stack Application: Evaluating Postfix Expression----->

Read the Postfix Expression ? 456*+7-

 Given Postfix Expn: 456*+7-

 Result after Evaluation: 27
```

Ex.No.-8 TREE TRAVERSALS

AIM:

To write a C program to implement tree and tree traversals.

ALGORITHM:

1. Start the program.
2. Declare the node.
3. Create the binary tree by inserting elements into it.
4. Traverse the binary tree by inorder and display the nodes.
5. Traverse the binary tree by preorder and display the nodes.
6. Traverse the binary tree by postorder and display the nodes.
7. Stop the program.

PROGRAM:

```
#  include  <stdio.h>
#  include  <alloc.h>
# include <conio.h>

typedef struct bin
{
        int data;
        struct bin *left;
        struct bin *right;
}node;

void insert(node *,node *);
void inorder(node *);
void preorder(node *);
void postorder(node *);
node *getnode();

void main()
{
        int choice;
        char ans='n';
        node *newnode,*root;
        root=NULL;
        clrscr ( );
```

```
do
{
        printf("\n\t Program for Binary Tree Travesal");
        printf("\n\t 1.Create");
        printf("\n\t 2.Inorder");
        printf("\n\t 3.Preorder");
        printf("\n\t 4.Postorder");
        printf("\n\t 5.Exit");
        printf("\n\t Enter your Choice:");
        scanf("%d",&choice);
        switch(choice)
        {
                case 1:
                        root=NULL;
                        do
                        {
                                newnode=getnode();
                                printf("\n\tEnter the element:");
                                scanf("%d",&newnode->data);
                                if(root==NULL)
                                root=newnode;
                                else
                                insert(root,newnode);
                                printf("\n\tDo u want to enter more elements?(y/n): ");
                                ans=getche();
                        }while(ans=='y' || ans=='Y');
                        clrscr();
                        break;

                case 2:
                        if(root==NULL)
                                printf("\n\t Tree is not created.");
                        else
                                inorder(root);
                        break;

                case 3:
                        if(root==NULL)
                                printf("\n\t Tree is not created.");
                        else
                                preorder(root);
                        break;
                case 4:
                        if(root==NULL)
                                printf("\n\t Tree is not created.");
                        else
                                postorder(root);
                        break;

                }
        }while(choice!=5);
}

node *getnode()
{
        node *temp;
        temp=(node *)malloc(sizeof(node));
        temp->left=NULL;
        temp->right=NULL;
        return temp;
}

void insert(node *root, node *newnode)
{
        char ch;
        printf("\n\t Where to insert LEFT/RIGHT of %d: ",root->data);
        ch=getche();
        if((ch=='r')||(ch=='R'))
        {
                if(root->right==NULL)
                {
                        root->right=newnode;
                }
                else
                        insert(root->right,newnode);;
        }
        else
        {
                if(root->left==NULL)
                {
                        root->left=newnode;
                }
                else
                        insert(root->left,newnode);
        }
}

void inorder(node *temp)
{
        if(temp!=NULL)
        {
                inorder(temp->left);
                printf(" %d",temp->data);
                inorder(temp->right);
        }
}

void preorder(node *temp)
{
        if(temp!=NULL)
        {
                printf(" %d",temp->data);
                preorder(temp->left);
                preorder(temp->right);
        }
}

void postorder(node *temp)
{
        if(temp!=NULL)
        {
                postorder(temp->left);
                postorder(temp->right);
                printf("  %d",temp->data);

        }
}
```

OUTPUT:

```
Program for Binary Tree Travesal
1.Create
2.Inorder
3.Preorder
4.Postorder
5.Exit
Enter your Choice:1
Enter the element:10
Do u want to enter more elements?(y/n):y

Enter the element:12
Where to insert LEFT/RIGHT of 10: l
Do u want to enter more elements?(y/n):y

Enter the element:17
Where to insert LEFT/RIGHT of 10:r
Do u want to enter more elements?(y/n):y

Enter the element:8
Where to insert LEFT/RIGHT of 10: l
Where to insert LEFT/RIGHT of 12:r
Do u want to enter more elements?(y/n):n
```

```
Program for Binary Tree Travesal
1.Create
2.Inorder
3.Preorder
4.Postorder
5.Exit
Enter your Choice:2
12      8       10      17

Program for Binary Tree Travesal
1.Create
2.Inorder
3.Preorder
4.Postorder
5.Exit
Enter your Choice:3
10      12      8       17

Program for Binary Tree Travesal
1.Create
2.Inorder
3.Preorder
4.Postorder
5.Exit
Enter your Choice:4
8       12      17      10
Program for Binary Tree Travesal
1.Create
2.Inorder
3.Preorder
4.Postorder
5.Exit
Enter your Choice:5
```

Ex.No.-9 IMPLEMENTATION OF BINARY SEARCH TREE

AIM:

To write a C program to implement binary search tree.

ALGORITHM:

1. Start the program.
2. Declare the node.
3. Read the elements to be inserted.
4. Create the binary search tree.
5. Read the element to be searched.
6. Visit the nodes by inorder.
7. Find the searching node and display if it is present with parent node.
8. Read the element to be removed from BST.
9. Delete that node from BST.
10. Display the binary search tree by inorder.
11. Stop the program.

PROGRAM:

```
# include  <stdio.h>
# include  <alloc.h>
# include <conio.h>
# include <stdlib.h>

typedef struct bst
{
        int data;
        struct bst *left,*right;
}node;

void insert(node *,node *);
void inorder(node *);
node *search(node *,int,node **);
void del(node *,int);


void main()
{
        int choice;
        char ans='N';
```

```
int key;
node *newnode,*root,*temp,*parent;
node *getnode();
root=NULL;
clrscr();
do
{
        printf("\n\t Program for Binary Search Tree");
        printf("\n\t 1.Create");
        printf("\n\t 2.Search");
        printf("\n\t 3.Delete");
        printf("\n\t 4.Display");
        printf("\n\t 5.Exit");
        printf("\n\t Enter your Choice:");
        scanf("%d",&choice);
        switch(choice)
        {
                case 1:
                        do
                        {
                                newnode=getnode();
                                printf("\n\tEnter the element:");
                                scanf("%d",&newnode->data);
                                if(root==NULL)
                                root=newnode;
                                else
                                insert(root,newnode);
                                printf("\n\tDo u want to enter more elements?(y/n): ");
                                ans=getche();
                        }while(ans=='y' || ans=='Y');
                        break;

                case 2:
                        printf("\n\tEnter the element to be searched:");
                        scanf("%d",&key);
                        temp=search(root,key,&parent);
                        printf("\n\tParent of node %d is %d",temp->data,parent->data);
                        break;

                case 3:
                        printf("\n\tEnter the element to be deleted:");
                        scanf("%d",&key);
                        del(root,key);
                        break;
                case 4:
                        if(root==NULL)
                                printf("\n\t Tree is not created.");
                else
                {
                    printf("\n The Tree is:");
                    inorder(root);
                }
                break;
            }
    }while(choice!=5);
}

node *getnode()
{
    node *temp;
    temp=(node *)malloc(sizeof(node));
    temp->left=NULL;
    temp->right=NULL;
    return temp;
}

void insert(node *root, node *newnode)
{
    if(newnode->data > root->data)
    {
        if(root->right==NULL)
        {
            root->right=newnode;
        }
        else
            insert(root->right,newnode);;
    }
    if(newnode->data < root->data)
    {
        if(root->left==NULL)
        {
            root->left=newnode;
        }
        else
            insert(root->left,newnode);
    }
}

void inorder(node *temp)
{
    if(temp!=NULL)
    {
        inorder(temp->left);
        printf("  %d",temp->data);
```

```
                inorder(temp->right);
        }
}

node *search(node *root,int key,node **parent)
{
        node *temp;
        temp=root;
        while(temp!=NULL)
        {
             if(temp->data==key)
             {
                        printf("\n\tThe %d element is present",temp->data);
                        return(temp);
             }
             *parent=temp;
             if(temp->data > key)
                 temp=temp->left;
             else
                 temp=temp->right;
        }
        return NULL;
}

void del(node *root, int key)
{
        node *temp,*parent,*tempsucc;
        temp=search(root,key,&parent);
        if(temp->left!=NULL&&temp->right!=NULL)
        {
                parent=temp;
                tempsucc=temp->right;
                while(tempsucc->left!=NULL)
                {
                        parent=tempsucc;
                        tempsucc=tempsucc->left;
                }
                temp->data=tempsucc->data;
                parent->right=NULL;
                printf("Now Deleted it!");
                return;
        }

        if(temp->left!=NULL&&temp->right!=NULL)
        {
                if(parent->left==temp)
                        parent->left=temp->left;
                else
                        parent->right=temp->left;
                        temp=NULL;
                        free(temp);
                        printf("Now deleted it!");
                        return;
        }
        if(temp->left!=NULL&&temp->right!=NULL)
        {
                if(parent->left==temp)
                        parent->left=temp->right;
                else
                        parent->right=temp->right;
                temp=NULL;
                free(temp);
                printf("Now deleted it!");
                return;
        }

                if(temp->left!=NULL&&temp->right!=NULL)
        {
                if(parent->left==temp)
                        parent->left=NULL;
                else
                        parent->right=NULL;
                        printf("Now deleted it!");
                        return;
        }
}
```

OUTPUT:

```
Program for Binary Search Tree
1.Create
2.Search
3.Delete
4.Display
5.Exit
Enter your Choice:1

Enter the element:10
Do u want to enter more elements?(y/n):y

Enter the element:8
Do u want to enter more elements?(y/n):y

Enter the element:9
```

```
Do u want to enter more elements?(y/n):y

Enter the element:7
Do u want to enter more elements?(y/n):y

Enter the element:15
Do u want to enter more elements?(y/n):y

Enter the element:13
Do u want to enter more elements?(y/n):y

Enter the element:14
Do u want to enter more elements?(y/n):y

Enter the element:12
Do u want to enter more elements?(y/n):y

Enter the element:16
Do u want to enter more elements?(y/n):n

1.Create
2.Search
3.Delete
4.Display
5.Exit
Enter your Choice:4

The Tree is: 7        8      9      10      12      13      14      15      16

1.Create
2.Search
3.Delete
4.Display
5.Exit
Enter your Choice:2

Enter the element to be searched:16
The 16 element is present
Parent of node 16 is 15

1.Create
2.Search
3.Delete
4.Display
5.Exit
Enter your Choice:5
```

Ex.No.-10 LINEAR & BINARY SEARCH

AIM:

To write a C program to perform linear search and binary search.

ALGORITHM:

Linear Search

1. Read n numbers and search value.
2. If search value is equal to first element then print value is found.
3. Else search with the second element and so on.

Binary Search

1. Read n numbers and search value.
2. If search value is equal to middle element then print value is found.
3. If search value is less than middle element then search left half of list with the same method.
4. Else search right half of list with the same method.

PROGRAM:

/*Searching*/

```
# include <stdio.h>
# include <stdlib.h>
# include <conio.h>

void main()
{
        int a[100],i,n,item,s=0,ch,beg,end,mid; clrscr();
        printf("Enter No. of Elements:");
        scanf("%d",&n);
        printf("\nEnter Elements:\n");
        for(i=1;i<=n;i++)
        {
                scanf("%d",&a[i]);
        }
        while(1)
        {
                printf("\n1.Linear Search\n2.Binary Search\n3.Exit\n"); printf("Enter
                your choice:");
                scanf("%d",&ch);
        switch(ch)
        {
            case 1:
            printf("<-----LINEAR SEARCH----->\n"); printf("\nEnter
            Element you want to Search:"); scanf("%d",&item);
            for(i=1;i<=n;i++)
            {
                if(a[i]==item)
                {
                    printf("\nData is Found at Location : %d",i); s=1;
                    break;
                }
            }
            if(s==0)
            {
                printf("Data is Not Found");
            }
break;
case 2:
            printf("<-----BINARY SEARCH ---->\n");
            printf("\nEnter Item you want to Search:");
            scanf("%d",&item);
            beg=1;
            end=n;
            mid=(beg+end)/2;
            while(beg<=end && a[mid]!=item)
            {
                if(a[mid]<item)
                beg=mid+1;
                else end=mid-1;
                mid=(beg+end)/2;
            }
            if(a[mid]==item)
            {
                printf("\nData is Found at Location : %d",mid);
            }
            else
            {
                printf("Data is Not Found");
            }
            break;
            case3:
                default: exit(0);
        }
    }
getch();
}
```

OUTPUT:

```
Enter No. of Elements: 5

Enter Elements: 2 4 3 5
1

1.Linear          Search
2.Binary Search 3.Exit
Enter your choice: 1

<-----LINEAR SEARCH---- >

Enter Element you want to Search: 1
Data is Found at Location : 5

1.Linear Search
2.Binary Search 3.Exit
Enter your choice: 2

<-----BINARY SEARCH ---->

Enter Item you want to Search: 3
Data is Found at Location : 3

1.Linear Search
2.Binary Search 3.Exit
Enter your choice: 3
```

Ex.No.-11 SORTING

AIM:

To write a C program to perform insertion sort, quick sort and bubble sort**.**

ALGORITHM:

Insertion Sort

1. Get the n elements to be sorted.
2. The ith element is compared from (i-1)th to 0th element and placed in proper position according to ascending value.
3. Repeat the above step until the last element.

Quick Sort

1. Pick an element, called a pivot, from the list.
2. Reorder the list so that all elements which are less than the pivot come before the pivot and so that all elements greater than the pivot come after it.
3. After this partitioning, the pivot is in its final position. This is called the partition operation.
4. Recursively sort the sub-list of lesser elements and the sub-list of greater elements

Bubble Sort

1. Get the n elements to be sorted.
2. Compare the first two elements of the array and swap if necessary.
3. Then, again second and third elements are compared and swapped if it is necessary and continue this process until last and second last element is compared and swapped.
4. Repeat the above two steps n-1 times and print the result.

PROGRAM:

```
/*Sorting*/
#include<conio.h>
#include<stdio.h>
#include<process.h>
```

```
void quickSort(int numbers[], int array_size); void
q_sort(int numbers[], int left, int right); void bubble(int
*array,int length);
void insertion(int a[], int n);

void insertion(int a[], int n)
{
  int i,j,temp; for(i=1;i<n;i++)
  {
    temp=a[i]; j=i-1;
    while((temp<a[j])&&(j>=0))
    {
         a[j+1]=a[j]; j=j-1;
}
    a[j+1]=temp;
  }
}

void display(int a[],int n)
{
        int i;
        printf("\n\t\t\tSorted List\n");
        for(i=0;i<n;++i) printf("\t%d",a[i]);
}

void q_sort(int a[], int left, int right)
{
        int pivot, l_hold, r_hold;
        l_hold = left;
        r_hold = right;
        pivot = a[left];
        while (left < right)
        {
                while ((a[right] >= pivot) && (left < right))
                 right--;
                if (left != right)
                {
                        a[left] = a[right];
                        left++;
                }
                while ((a[left] <= pivot) && (left < right))
                                left++;
                                if (left != right)
                                {
                                        a[right] = a[left]; right--;
                                }
                        }
                        a[left] = pivot;
                        pivot = left;
                        left = l_hold;
                        right = r_hold;
                        if (left < pivot)q_sort(a, left, pivot-1);
                        if (right > pivot) q_sort(a, pivot+1, right);
}

void bubble(int *array,int length)
{
        int i,j; for(i=0;i<length;i++)
        {
                for(j=0;j<i;j++)
                {
                        if(array[i]>array[j])
                        {
                                inttemp=array[i];
                                array[i]=array[j];
                                array[j]=temp;
                        }
                }
        }
}

void main()
{
        int a[100],n,i,ch;
        clrscr( );
        printf("\nEnter The Number Of Elements\t: ");
        scanf("%d",&n);
        printf("\nEnter Elements\n");
        for(i=0;i< n;++i)
                scanf("%d",&a[i]);
        while(1)
        {
                printf("\n1.Insertion sort\n2.Quick sort\n3.Bubble sort\n4.Exit\n"); printf("Enter your
                choice:");
                scanf("%d",&ch);
                switch(ch)
                {
                        case 1:
                                printf("<-----Insertion SORT ---- >\n");
                                insertion(a,n);
                                display(a,n);
                                break;
                        case 2:
                                printf("<-----Quick SORT ---->\n");
```

```
                                        q_sort(a,0,n-1);
                                        display(a,n);

                                        break;
case 3:
                                        bubble(a,n);
                                        printf("<-----Bubble SORT ---- >\n"); printf("\n\t\t\tSorted
                                        List\n");
                                        for (i=n-1;i>=0;i--)
                                                  printf("\t%d",a[i]);
                                        break;
                              case 4:
                                        exit(0); default:
                              printf("Enter a Valid Choice!");
                    }
          }
getch();
}
```

OUTPUT:

```
Enter The Number Of Elements : 5

Enter Elements 2 4 1 3
5

1.Insertion sort 2.Quick
sort 3.Bubble sort
4.Exit
Enter your choice: 1

<-----Insertion SORT ---->
                    Sorted List
      1       2      3      4    5

1.Insertion sort
2.Quick sort
3.Bubble sort
4.Exit
Enter your choice:
2
<-----Quick SORT---- >

                    Sorted List
```

```
       1        2       3       4     5

1.Insertion sort
2.Quick sort
3.Bubble sort
4.Exit
Enter your choice:
3
<-----Bubble SORT ---->

                    Sorted List
       1        2       3       4     5
```

12 MERGE SORT

AIM:

To write a c program to implement merge sort.

```
#include <stdio.h>

#define max 10

int a[11] = { 10, 14, 19, 26, 27, 31, 33, 35, 42, 44, 0 };
int b[10];

void merging(int low, int mid, int high) {
   int l1, l2, i;

   for(l1 = low, l2 = mid + 1, i = low; l1 <= mid && l2 <= high; i++) {
      if(a[l1] <= a[l2])
         b[i] = a[l1++];
      else
         b[i] = a[l2++];
   }

   while(l1 <= mid)
      b[i++] = a[l1++];

   while(l2 <= high)
      b[i++] = a[l2++];

   for(i = low; i <= high; i++)
      a[i] = b[i];
}

void sort(int low, int high) {
   int mid;

   if(low < high) {
      mid = (low + high) / 2;
```

```
    sort(low, mid);
    sort(mid+1, high);
    merging(low, mid, high);
  } else {
    return;
  }
}

int main() {
  int i;

  printf("List before sorting\n");

  for(i = 0; i <= max; i++)
    printf("%d ", a[i]);

  sort(0, max);

  printf("\nList after sorting\n");

  for(i = 0; i <= max; i++)
    printf("%d ", a[i]);
}
```

RESULT:

```
List before sorting
10 14 19 26 27 31 33 35 42 44 0
List after sorting
0 10 14 19 26 27 31 33 35 42 44
```

Ex.No.-13 LINEAR PROBING

AIM:

To write a c program to create hash table and collision handling by linear probing.

ALGORITHM:

1. Start the program.
2. Read the numbers to be stored in hash table.
3. Create the hash function by generating the hash key.
4. If the location indicated by hash key is empty, then place the number in the hash table.
5. If collision occurs, then search for empty location.
6. If found, place the number at that location.
7. Display the hash table.

PROGRAM:

```
#  include  <stdio.h>
# include <conio.h>
# include <stdlib.h>
# define MAX 10

void main()
{
        int a[MAX],num,key,i;
        char ans;
        int create(int);
        void linearprob(int[], int,int),display(int[]);
        clrscr();

        printf("\nCOLLISION HANDLING BY LINEAR PROBING");
        for(i=0;i<MAX;i++)
        a[i]=-1;
        do
        {
                printf("\nEnter the number:");
                scanf("%d",&num);
                key=create(num);
                linearprob(a,key,num);
                printf("\n Do U wish to continue? (y/n)");
                ans=getche();
        }
                while(ans=='y');
                display(a);
                getch();
}

int create(int num)
{
        int key;
        key=num%10;
        return key;
}

void linearprob(int a[MAX],int key,int num)
{
        int flag,i,count=0;
        void display(int a[]);
        flag=0;
        if(a[key]==-1)
                a[key]=num;
        else
        {
                i=0;
                while(i<MAX)
                {
                        if(a[i]!=-1)
                                count++;
                                i++;
                }
                if(count==MAX)
                {
                        printf("\n Hash Table is full");
                        display(a);
                        getch();
                        exit(1);
                }
                for(i=key+1;i<MAX;i++)
                        if(a[i]==-1)
```

```
                    {
                            a[i]=num;
                            flag=1;
                            break;
                    }
            for(i=0;i<key&&flag==0;i++)
                    if(a[i]==-1)
                    {
                            a[i]=num;
                            flag=1;
                            break;
                    }
        }
}

void display(int a[MAX])
{
        int i;
        printf("\n Hash Table is..\n");
        for(i=0;i<MAX;i++)
        printf("\n %d  %d  ",i,a[i]);
}
```

OUTPUT:

```
COLLISION HANDLING BY LINEAR PROBING
Enter the number:131
Do U wish to continue? (y/n)y
Enter the number:21
Do U wish to continue? (y/n)y
Enter the number:3
Do U wish to continue? (y/n)y
Enter the number:4
Do U wish to continue? (y/n)y
Enter the number:8
Do U wish to continue? (y/n)y
Enter the number:9
Do U wish to continue? (y/n)y
```

```
Enter the number:18
Do U wish to continue? (y/n)n
Hash Table is..
0       18
1       131
2       21
3       3
4       4
5       5
6       -1
7       -1
8       8
9       9
```

Lab Manual

<u>Java Programming</u>

1. OBJECTIVE AND RELEVANCE

The objective of this course is to provide students with hands on knowledge on the core java programming environment, practical knowledge about the basic programming language concepts, practical knowledge of object-oriented concepts, practical knowledge about java's exception-handling mechanism, multithreading programming, string handling, applets and hands on experience in java development environment.

a. To introduce java compiler and eclipse platform.
b. To impart hand on experience with java programming

1. Learning JAVA Programming Language.
2. Learning object-oriented programming through Java
3. Learning the rich API provided with Java
4. Learning object-oriented design patterns used in Java API Design.
5. Learning and practicing problems which would help touch upon most of the JAVA API packages with a view to get maximum possible exposure to JAVA API Library.
6. Learning how to use JAVA API documentation to quickly find the API that can be used as a part of a given problem solution.
7. Learning how JAVA supports defensive programming with exceptions.
8. Learning JAVA support for building rich user interfaces, both web-based and in regular desk-top applications.
9. Learning JAVA support for parallel programming using multiple threads.
10. Learning JAVA support for writing client-server applications

2. OUTCOMES

1. Basics of java programming, multi-threaded programs and Exception handling.
2. The skills to apply OOP in Java programming in problem solving
3. Designing GUI components Using AWT.
4. Grouping the individual objects into single entity Using Collection Frame work
5. Working with the Files using java.io Package

3. EQUIPMENT REQUIRED

Product	Eclipse IDE &JDK1.5
Operating System	**Server:** Microsoft Windows 2000 Server, or later **Client:** Microsoft Windows 2000 / XP / Vista
Processor	**Server**: Dual Intel Pentium 4 for 10 or more users, single Pentium 4 for less than 10 users **Client**: Intel Pentium or AMD Athlon, 233 MHz or faster 32 and 64-bit versions
RAM	**Server:** 1 GB **Client:** 1 GB
Drive space	**Server**: 20 GB available after installation of the operating system, RAID 5 or RAID 1+0 configuration **Client:** 1.5 GB available after installation of the operating system and Oracle Client software
Drives	Runs from hard disk

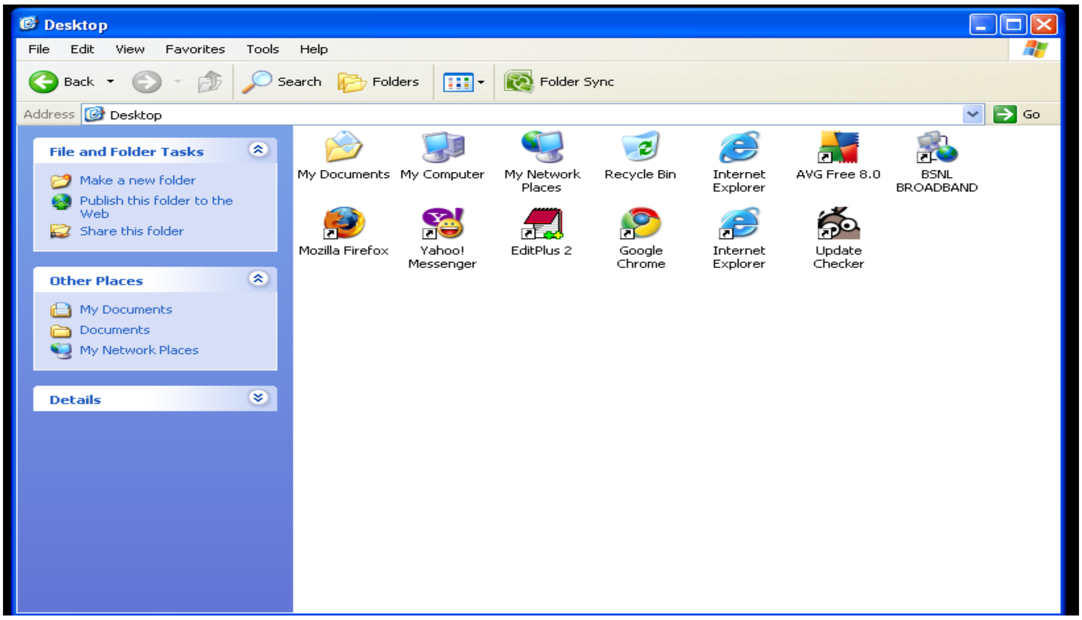
4. PROCEDURE FOR INSTALLING JAVA

STEP 1:

- First install the java version (1.5/1.6) Ur having..

STEP 2:

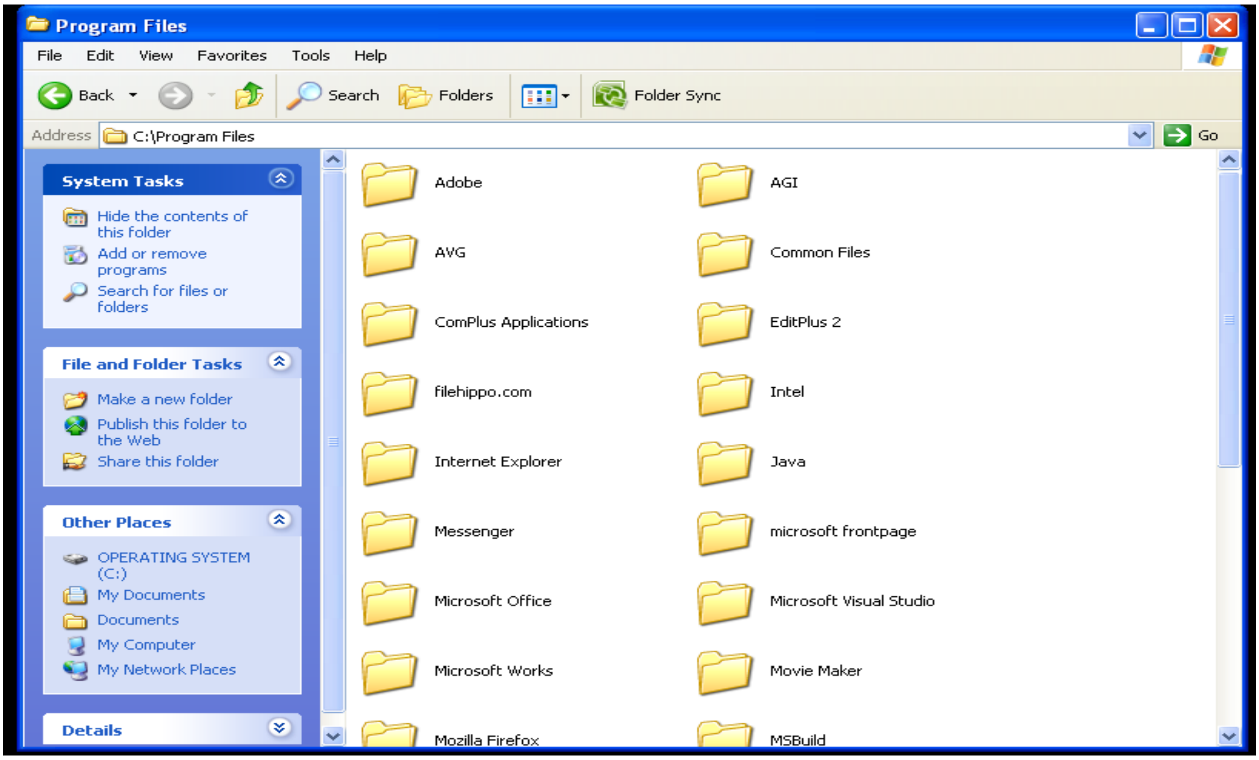
- Go To My Computer



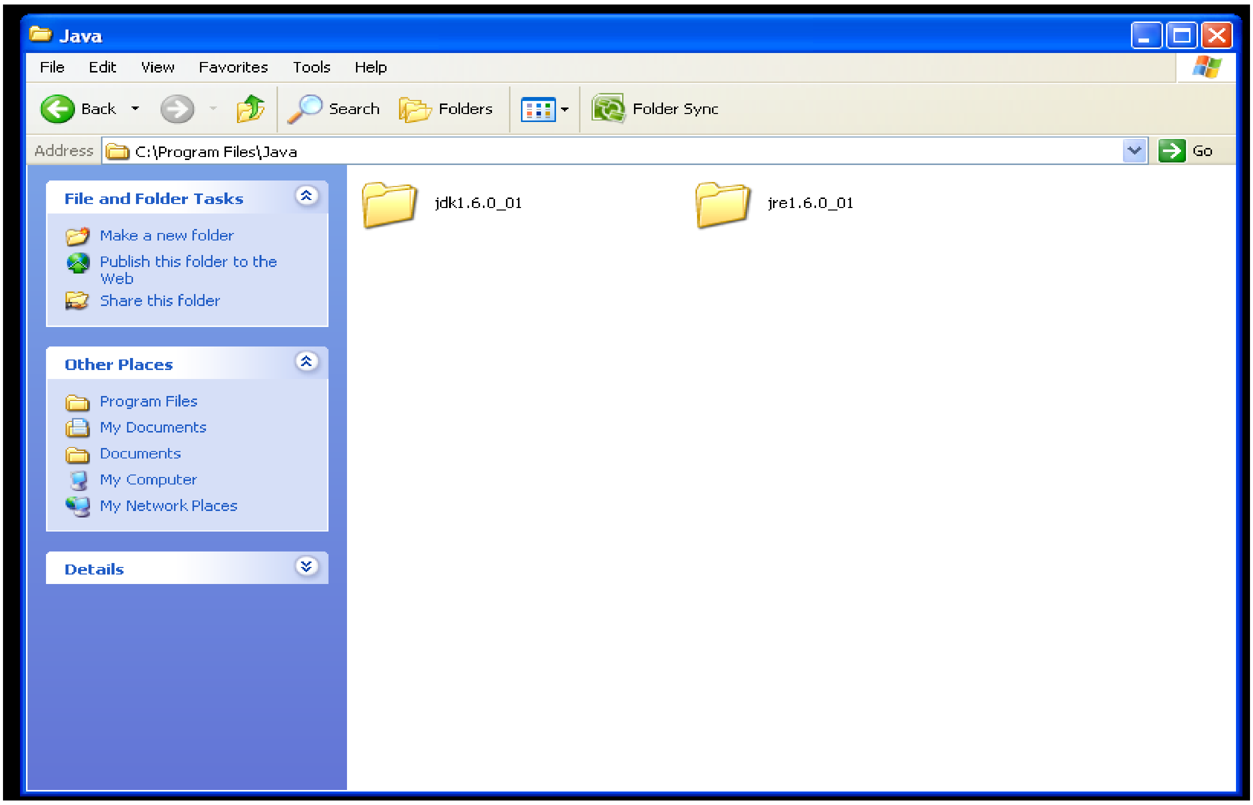
- Then go to the drive where u has installed java…

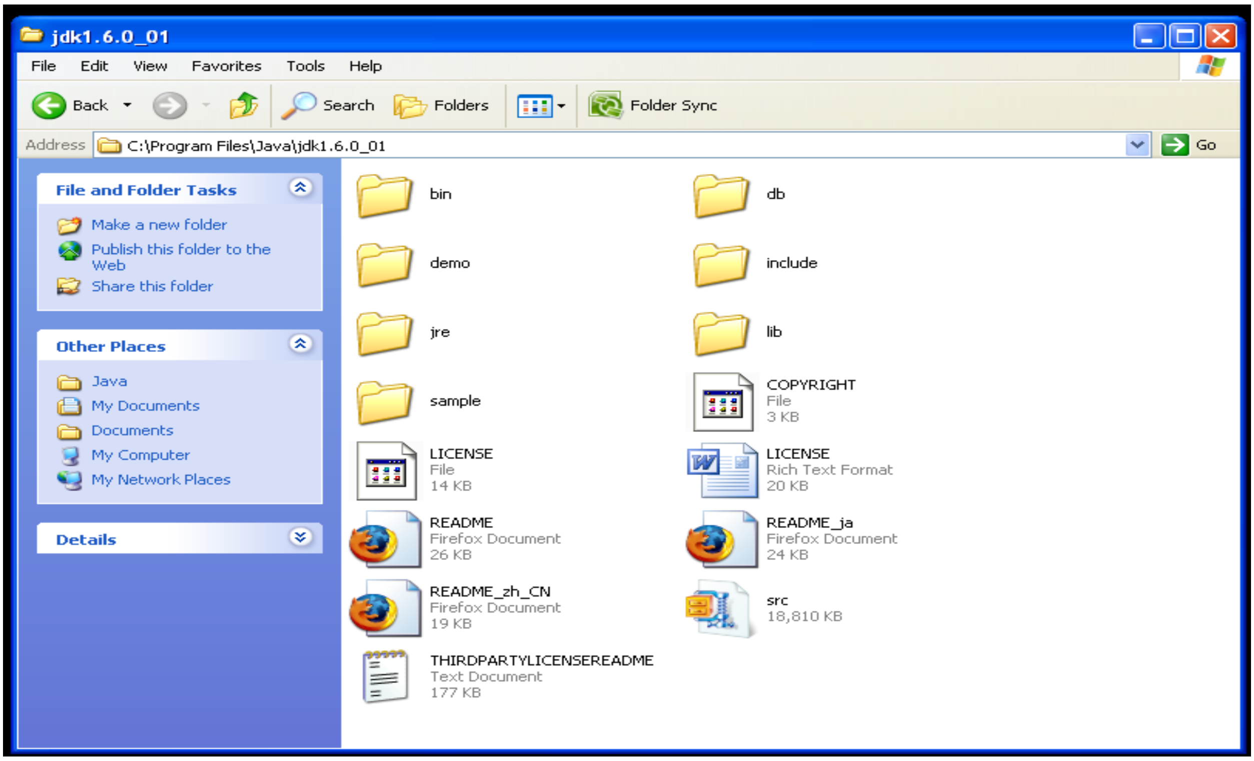
  For e.g.:

  1. Go to Drive.
  2. In Program Files
  3. Go to JAVA



4. Go to JDK(1.7) _01

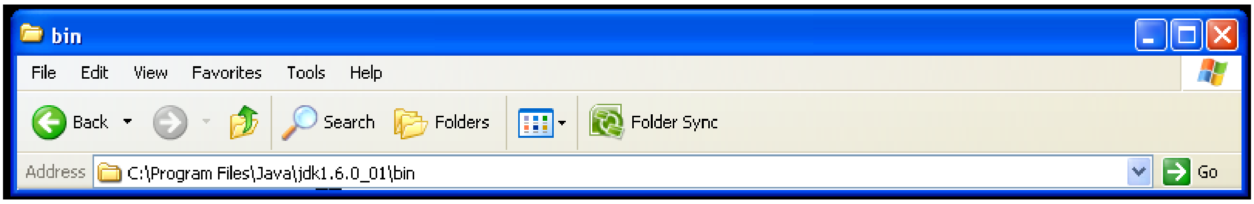




Address: C:\Program Files\Java\jdk1.6.0_01

jdk1.6.0_01
File Edit View Favorites Tools Help
Back Search Folders Folder Sync
Go
File and Folder Tasks
Make a new folder
Publish this folder to the Web
Share this folder
Other Places
Java
My Documents
Documents
My Computer
My Network Places
Details
bin
db
demo
include
jre
lib
sample
COPYRIGHT File 3 KB
LICENSE File 14 KB
LICENSE Rich Text Format 20 KB
README Firefox Document 26 KB
README_ja Firefox Document 24 KB
README_zh_CN Firefox Document 19 KB
src 18,810 KB
THIRDPARTYLICENSEREADME Text Document 177 KB

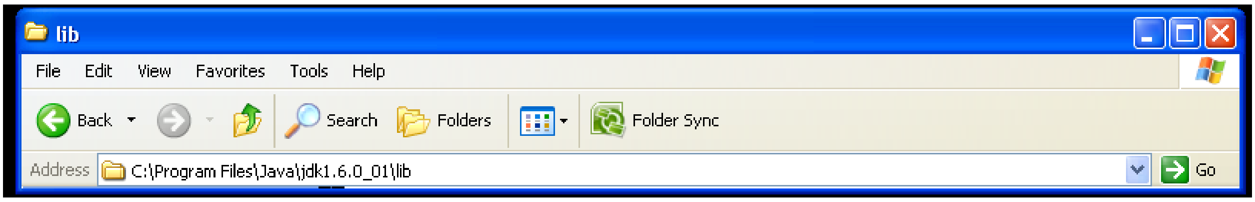
5.Go to bin



Address: C:\Program Files\Java\jdk1.6.0_01\bin

bin
File Edit View Favorites Tools Help
Back Search Folders Folder Sync
Go

6.Go to lib

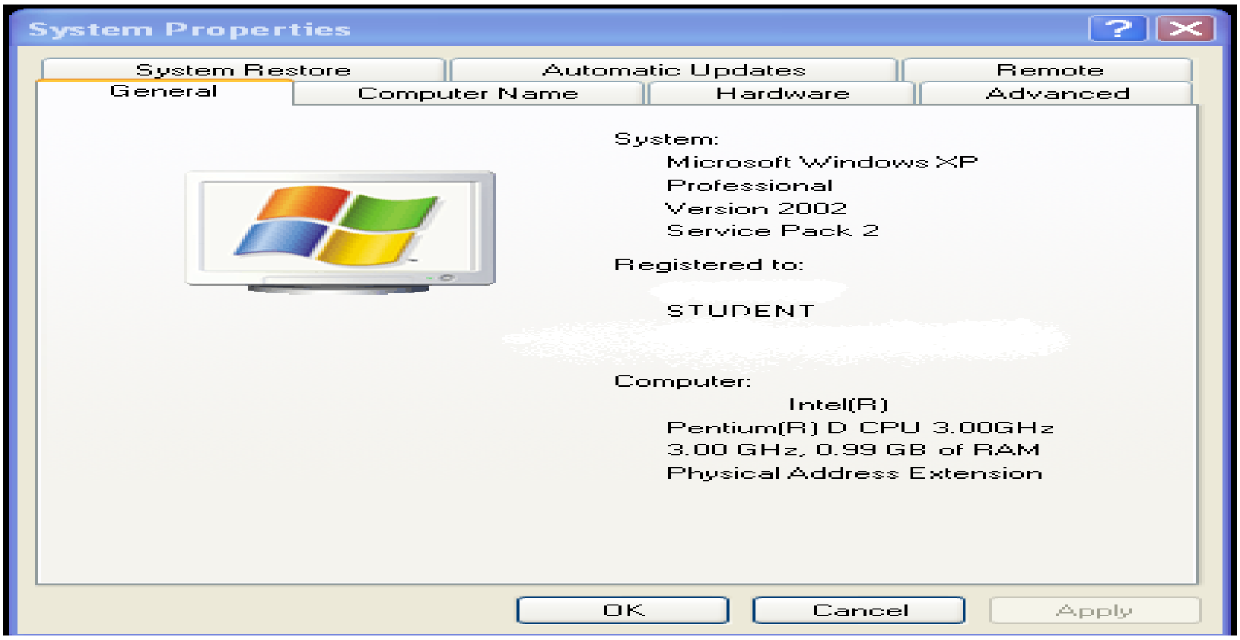


Address: C:\Program Files\Java\jdk1.6.0_01\lib

lib
File Edit View Favorites Tools Help
Back Search Folders Folder Sync
Go

STEP 3:

- My computer---→Properties



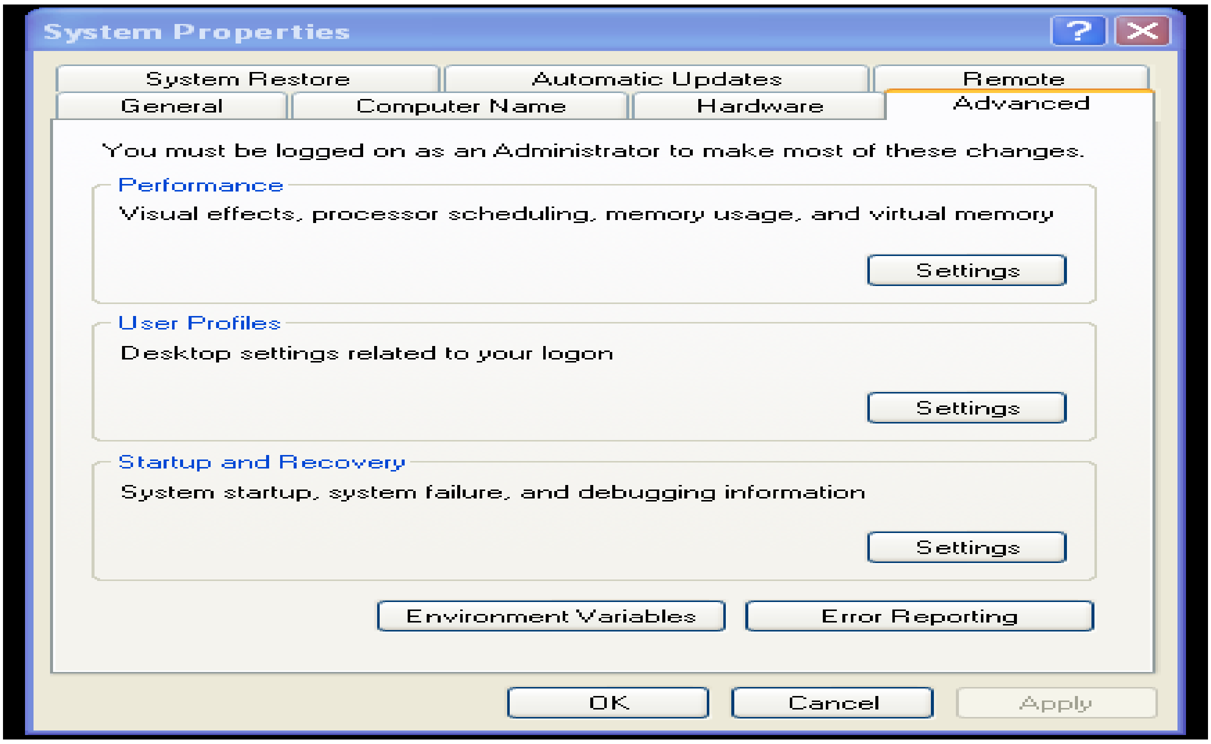
System:
Microsoft Windows XP
Professional
Version 2002
Service Pack 2

Registered to:

STUDENT

Computer:
Intel(R)
Pentium(R) D CPU 3.00GHz
3.00 GHz, 0.99 GB of RAM
Physical Address Extension

- Properties---→Advanced



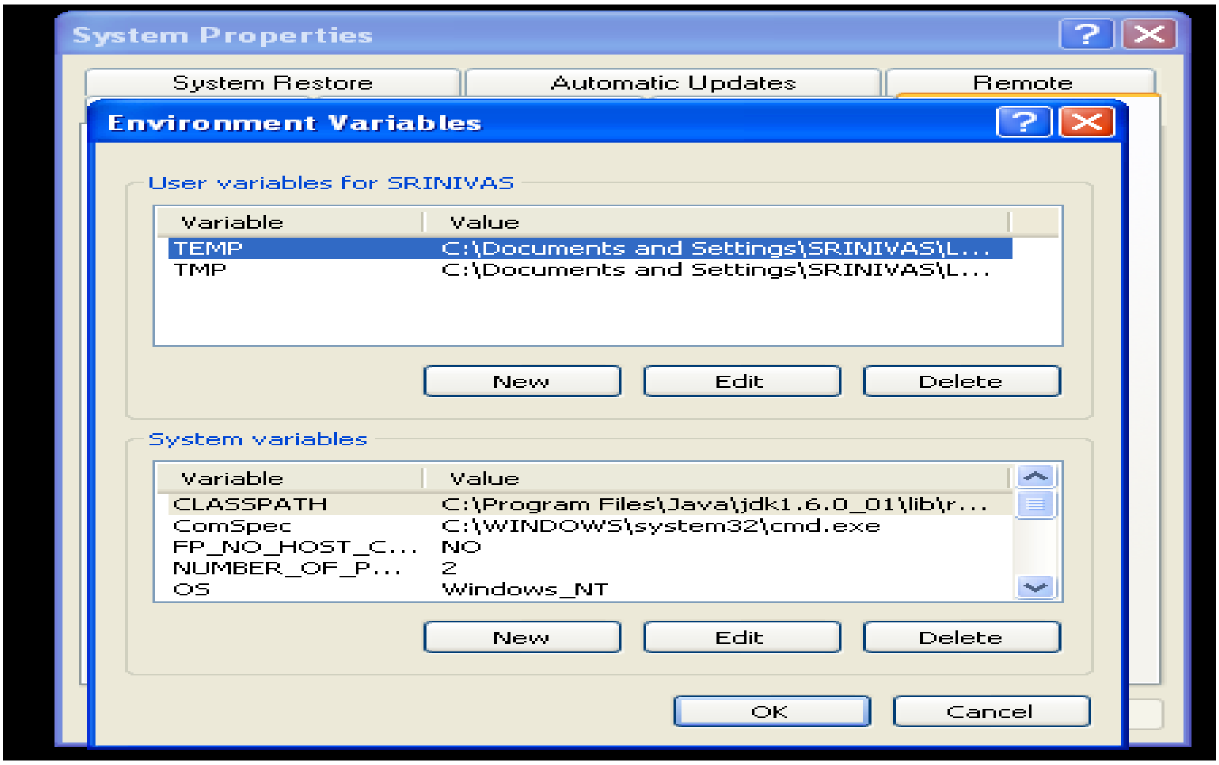
You must be logged on as an Administrator to make most of these changes.

Performance
Visual effects, processor scheduling, memory usage, and virtual memory

User Profiles
Desktop settings related to your logon

Startup and Recovery
System startup, system failure, and debugging information

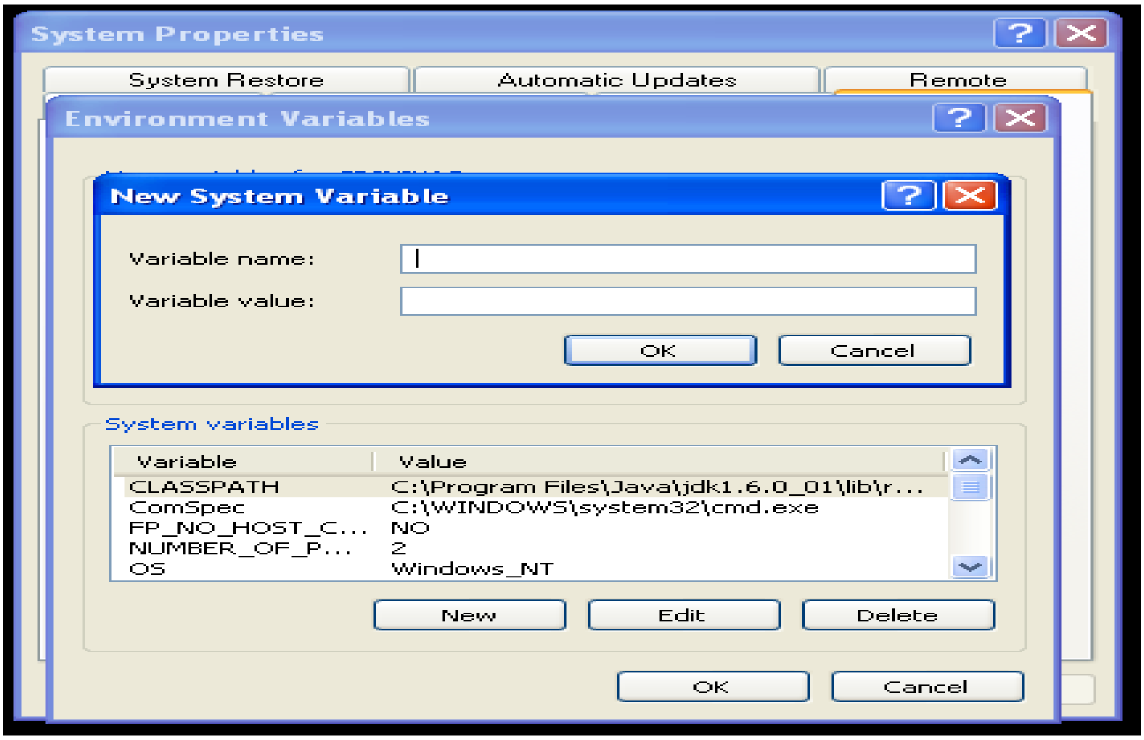
- Advanced---→Environmental variables



Variable	Value
TEMP	C:\Documents and Settings\SRINIVAS\L...
TMP	C:\Documents and Settings\SRINIVAS\L...

Variable	Value
CLASSPATH	C:\Program Files\Java\jdk1.6.0_01\lib\r...
ComSpec	C:\WINDOWS\system32\cmd.exe
FP_NO_HOST_C...	NO
NUMBER_OF_P...	2
OS	Windows_NT

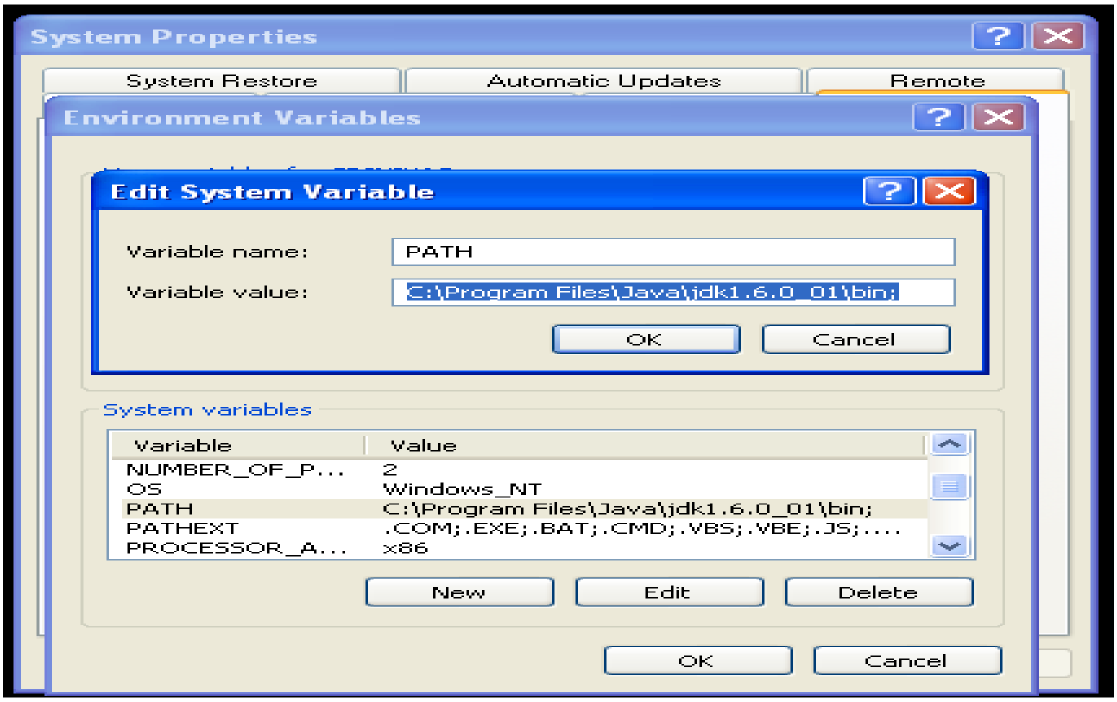
- ➢ Environmental variables---→System variables---→NEW



Variable	Value
CLASSPATH	C:\Program Files\Java\jdk1.6.0_01\lib\r...
ComSpec	C:\WINDOWS\system32\cmd.exe
FP_NO_HOST_C...	NO
NUMBER_OF_P...	2
OS	Windows_NT

- ➢ Variable name: PATH

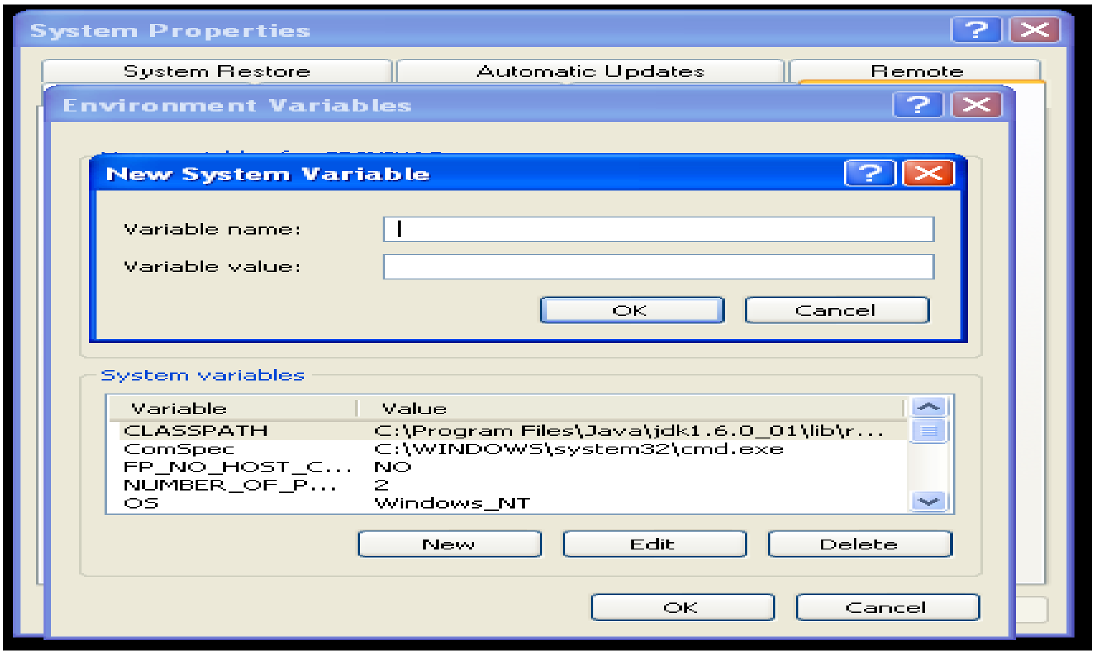
Variable value: C:\Program Files\Java\jdk1.6.0_01\bin;
**NOTE: For this go to step 2:-- 5th step!



Variable name: PATH
Variable value: C:\Program Files\Java\jdk1.6.0_01\bin;

Variable	Value
NUMBER_OF_P...	2
OS	Windows_NT
PATH	C:\Program Files\Java\jdk1.6.0_01\bin;
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;....
PROCESSOR_A...	x86

- Environmental variables---→System variables---→NEW



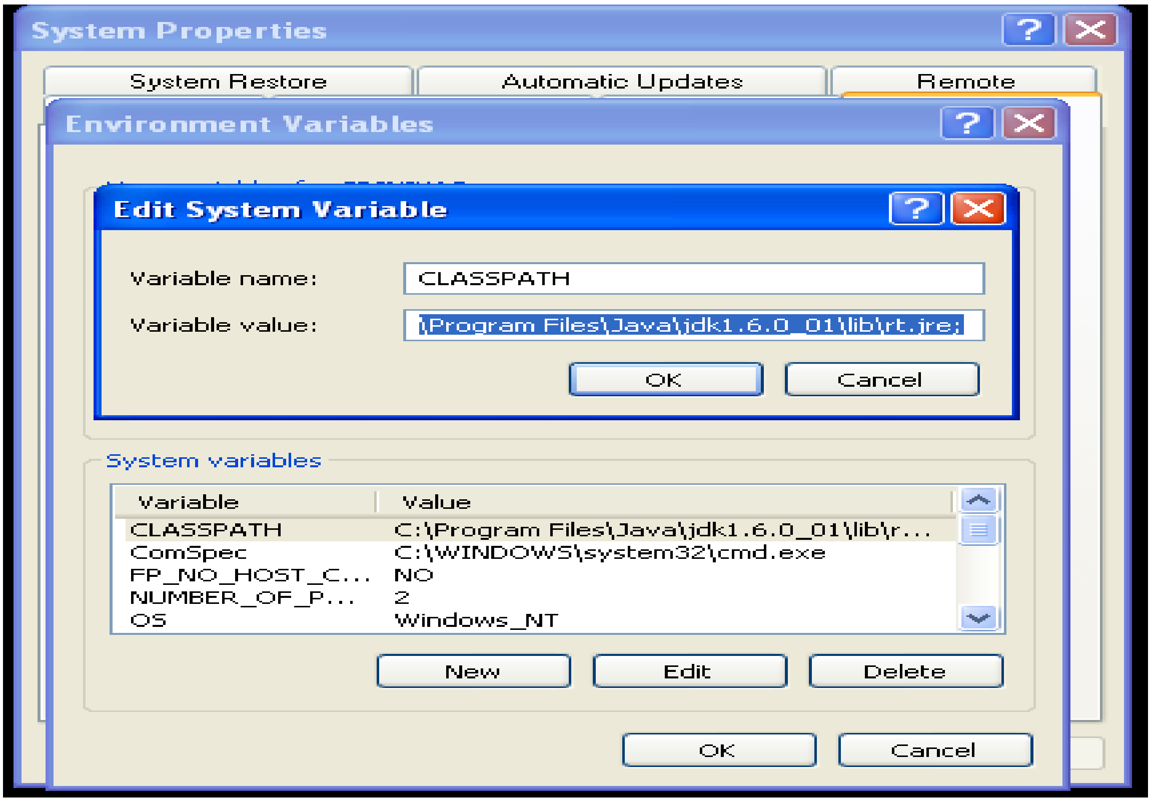
Variable name:
Variable value:

Variable	Value
CLASSPATH	C:\Program Files\Java\jdk1.6.0_01\lib\r...
ComSpec	C:\WINDOWS\system32\cmd.exe
FP_NO_HOST_C...	NO
NUMBER_OF_P...	2
OS	Windows_NT

Variable name: CLASSPATH
Variable value:
C:\Program Files\Java\jdk1.6.0_01\lib\rt.jre;

**NOTE: For this go to step 2:-- 6th step!

****//THE RED COLOURED PATH SHOULD BE ADDED….***//



Variable name: CLASSPATH

Variable value: \Program Files\Java\jdk1.6.0_01\lib\rt.jre;

Variable	Value
CLASSPATH	C:\Program Files\Java\jdk1.6.0_01\lib\r...
ComSpec	C:\WINDOWS\system32\cmd.exe
FP_NO_HOST_C...	NO
NUMBER_OF_P...	2
OS	Windows_NT

- Then click OK..!

JAVA INSTALLATION IS FINISHED...!

5. CODE OF CONDUCT

- Students should report to the concerned lab as per the time table.
- Students who turn up late to the labs will not be permitted to do the program schedule for the day.
- After completion of the program, certification of the concerned staff in-charge in the observation book is necessary.
- Student should bring a notebook of 100 pages and should enter the readings/observations into the notebook while performing the experiment.
- The record of observations along with the detailed experimental procedure of the experiment in the immediate last session should be submitted and certified staff member in-charge.
- Not more than 3-students in a group are permitted to perform the experiment on the set.
- The group-wise division made in the beginning should be adhered to and no mix up of students among different groups will be permitted.
- The components required pertaining to the experiment should be collected from stores in-charge after duly filling in the requisition form.
- When the experiment is completed, should disconnect the setup made by them, and should return all the components/instruments taken for the purpose.
- Any damage of the equipment or burn-out components will be viewed seriously either by putting penalty or by dismissing the total group of students from the lab for the semester/year.
- Students should be present in the labs for total scheduled duration
- Students are required to prepare thoroughly to perform the experiment before coming to laboratory

6. SYLLABUS

Course Objectives:

- To write programs using abstract classes.
- To write programs for solving real world problems using java collection frame work.
- To write multithreaded programs.
- To write GUI programs using swing controls in Java.
- To introduce java compiler and eclipse platform.
- To impart hands on experience with java programming.

Course Outcomes:

- Able to write programs for solving real world problems using java collection frame work.
- Able to write programs using abstract classes.
- Able to write multithreaded programs.
- Able to write GUI programs using swing controls in Java.

LIST OF PROGRAMS

1. Write a java Program to Demonstrate the concept of class Box with Constructors
2. Write a java Program to Demonstrate the concept of Method Overloading
3. Write a java Program to Demonstrate the concept of Inheritance
4. Write a Java programto Demonstrate the concept of Multiple Inheritance
5. A program to illustrate thread synchronization.
6. Write a Java program to illustrate Exception handling
7. A program using string tokenizer
8. Write a Java program to demonstrate LinkedList
9. A program using TreeSet class
10. Write a java program to demonstrate HashSet class
11. Write a java program to demonstrate map classes
12. A program to illustrate the usage of Byte and character I/O Streams.
13. A program to illustrate the usage of serialization
14. An application involving GUI with different controls, menus and event handling.
15. A program to implement an applet.

REFERENCE BOOKS

1. Java for Programmers, P. J. Deitel and H. M. Deitel, 10th Edition Pearson education.
2. Thinking in Java, Bruce Eckel, Pearson Education.
3. Java Programming, D. S. Malik and P. S. Nair, Cengage Learning.
4. Core Java, Volume 1, 9th edition, Cay S. Horstmann and G Cornell, Pearson.

7. SESSION PLAN

S. NO.	WEEK NO.	UNIT AS PER SYLLABUS	ACTIVITY	REMARKS
1	1	UNIT-I	Write a java Program to Demonstrate the concept of class Box with Constructors	OU
2	1	UNIT-I	Write a java Program to Demonstrate the concept of Method Overloading	OU
3	2	UNIT-I	Write a java Program to Demonstrate the concept of Inheritance	OU
4	3	UNIT-I	Write a Java programto Demonstrate the concept of Multiple Inheritance	OU
5	4	UNIT-II	A program to illustrate thread synchronization.	OU
6	4	UNIT-II	Write a Java program to illustrateException handling	OU
7	5	UNIT-III	A program using string tokenizer	OU
8	5	UNIT-III	Write a Java program to demonstrate LinkedList	OU
9	6	UNIT-III	A program using TreeSet class	OU
10	7	UNIT-III	Write a java program to demonstrate HashSet class	OU
11	8	UNIT-III	Write a java program to demonstrate map classes	OU
12	9	UNIT-IV	A program to illustrate the usage of Byte and character	OU

			I/O Streams.	
13	10	UNIT-IV	A program to illustrate the usage of serialization	OU
14	11	UNIT-V	An application involving GUI with different controls,menus and event handling.	OU
15	12	UNIT-V	A program to implement Applet	OU

EXPERIMENT-1

Preamble and Aim.

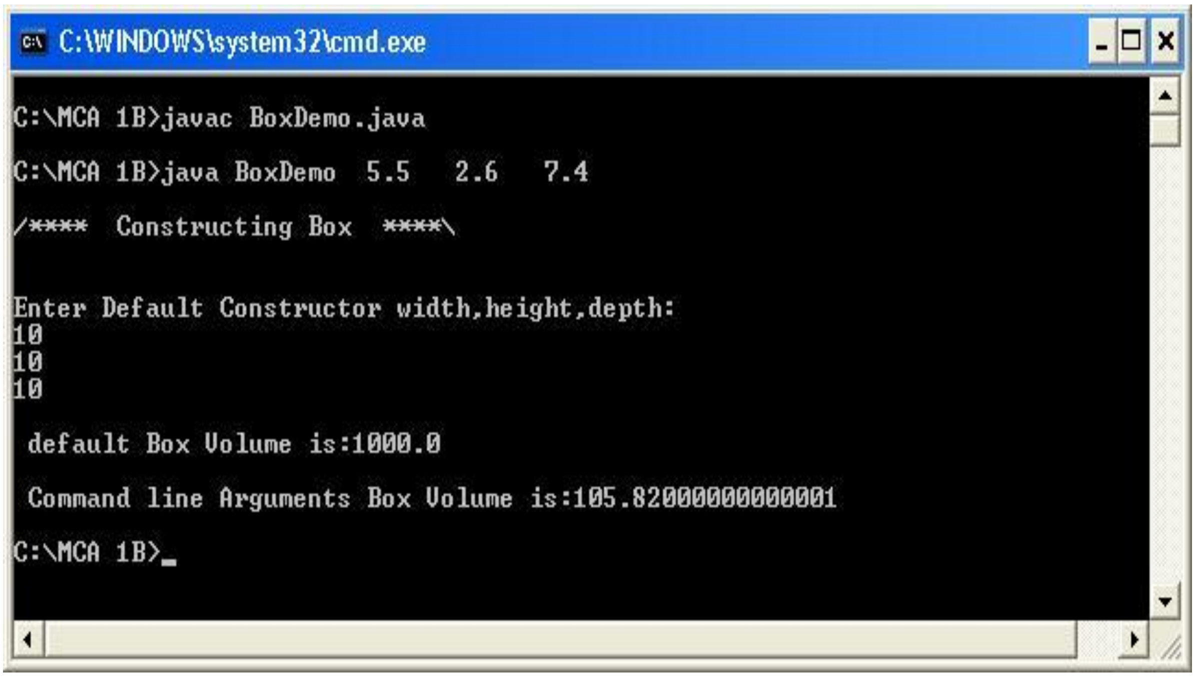
Use Eclipse or Netbean platform or Edit plus and acquaint with the various menus. Create a test project, add a Box class and run it. See how you can use auto suggestions, auto fill. Try code formatter and code refactoring like renaming variables, methods and classes. Try debug step by step with a small program of about Constructors.

Program:-

```
import java.io.*;
class  Box
{
      double width;
      double height;
      double depth;
Box()
      {
      System.out.println("\n/****  Constructing Box  ****\\\n");
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
      System.out.println("\nEnter Default Constructor width,height,depth:");
        try
        {
            width=Integer.parseInt(br.readLine());
            height=Integer.parseInt(br.readLine());
            depth=Integer.parseInt(br.readLine());
        }
        catch (IOException ioe)
        {
        }
}
Box(double w,double h,double d)
      {
        width=w;
        height=h;
        depth=d;
      }
double volume()
      {
       return (width*height*depth);
      }
}
class BoxDemo
{
                  public static void main(String[] args)
      {
```

```
                Box mybox1= new Box();
                Box mybox2= new Box(Double.parseDouble(args[0]),
                                    Double.parseDouble(args[1]),
                                    Double.parseDouble(args[2]));
                double vol;
                //get Volume of first box
                vol=mybox1.volume();
                System.out.println("\n default Box Volume is:"+vol);
                //get Volume of second box
                if(args.length==3)
                {
                vol=mybox2.volume();
                System.out.println("\n Command line Arguments Box Volume is:"+vol);
                }
                else

        System.out.println("!EnterCorrectCommandlineargumentstoBox(width,height,depth)"
    );
        }
    }
```

OUTPUT:-



```
C:\MCA 1B>javac BoxDemo.java

C:\MCA 1B>java BoxDemo  5.5   2.6   7.4

/****  Constructing Box  ****\

Enter Default Constructor width,height,depth:
10
10
10

 default Box Volume is:1000.0

 Command line Arguments Box Volume is:105.82000000000001

C:\MCA 1B>_
```

EXPERIMENT-2

Preamble and Aim.

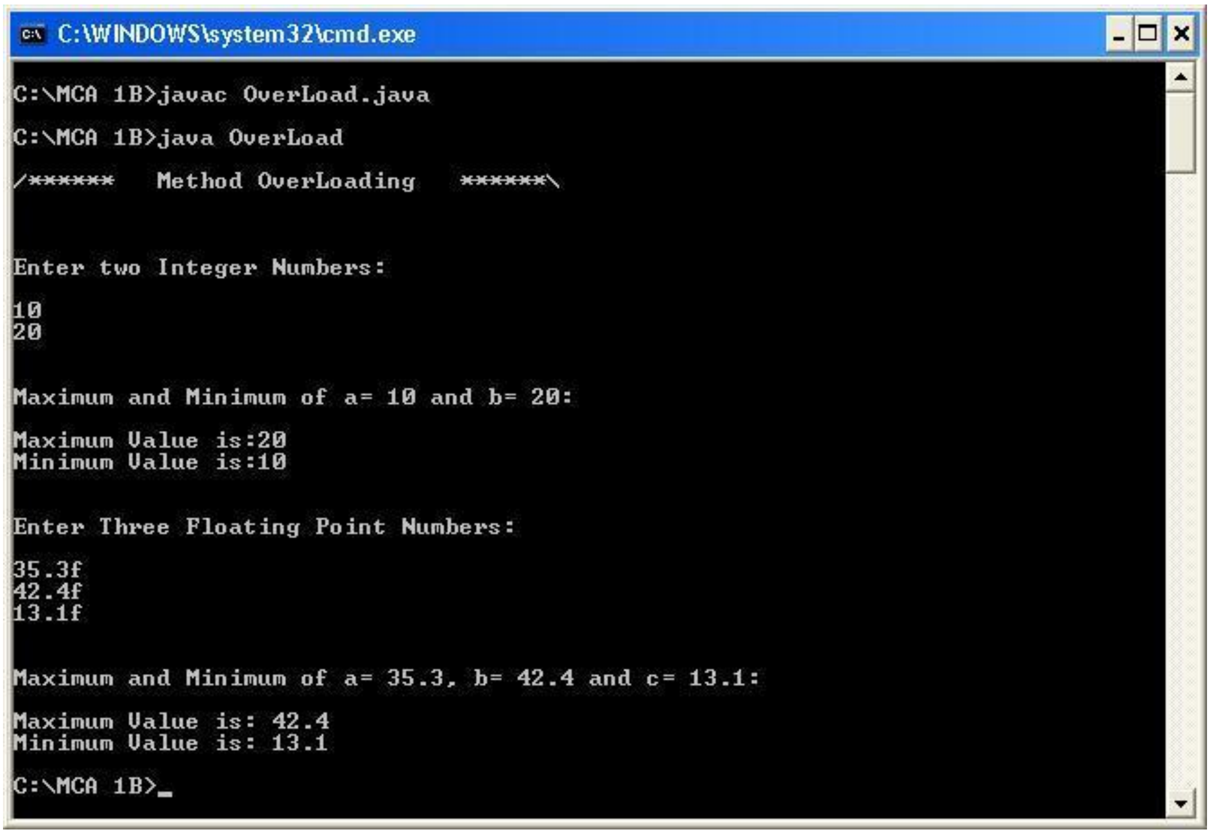
Write a java Program to Demonstrate the concept of Method Overloading

Program:-

```
import java.io.*;
class  OverLoadDemo
{
      void max_min(int a,int b)
      {
               System.out.println("\n\nMaximum  and  Minimum  of  a=  "+a+"  and  b=
"+b+":\n");
               if(a>b)
                 {
                      System.out.println("\nMaximum Value is:"+a);
                    System.out.println("Minimum Value is:"+b);
                 }
               else
                 {
                         System.out.println("Maximum Value is:"+b);
                      System.out.println("Minimum Value is:"+a);
                 }
      }
      void max_min(float a,float b,float c)
      {
              float max,min;
              max=(a>b)?((a>c)?a:c):((b>c)?b:c);
              min=(a<b)?((a<c)?a:c):((b<c)?b:c);
              System.out.println("\n\nMaximum and Minimum of a= "+a+" b= "+b+" and
c= "+c+":\n");
              System.out.println("Maximum Value is: "+max);
            System.out.println("Minimum Value is: "+min);
      }
}
class OverLoad
{
      public static void main(String[] args) throws IOException
      {
              BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
              OverLoadDemo ol=new OverLoadDemo();
              System.out.println("\n/***** Method Overloading   ******\\\n");
System.out.println("\n\nEnter two Integer Numbers:\n");
ol.max_min(Integer.parseInt(br.readLine()),Integer.parseInt(br.readLine()));
System.out.println("\n\nEnter Three Floating Point Numbers:\n");
```

```
    ol.max_min(Float.parseFloat(br.readLine()),Float.parseFloat(br.readLine()),Float.parseFl
    oat(br.readLine()));
    }
}
```

OUTPUT:-



```
C:\MCA 1B>javac OverLoad.java

C:\MCA 1B>java OverLoad

/******   Method OverLoading   ******\


Enter two Integer Numbers:

10
20


Maximum and Minimum of a= 10 and b= 20:

Maximum Value is:20
Minimum Value is:10


Enter Three Floating Point Numbers:

35.3f
42.4f
13.1f


Maximum and Minimum of a= 35.3, b= 42.4 and c= 13.1:

Maximum Value is: 42.4
Minimum Value is: 13.1

C:\MCA 1B>_
```

EXPERIMENT-3

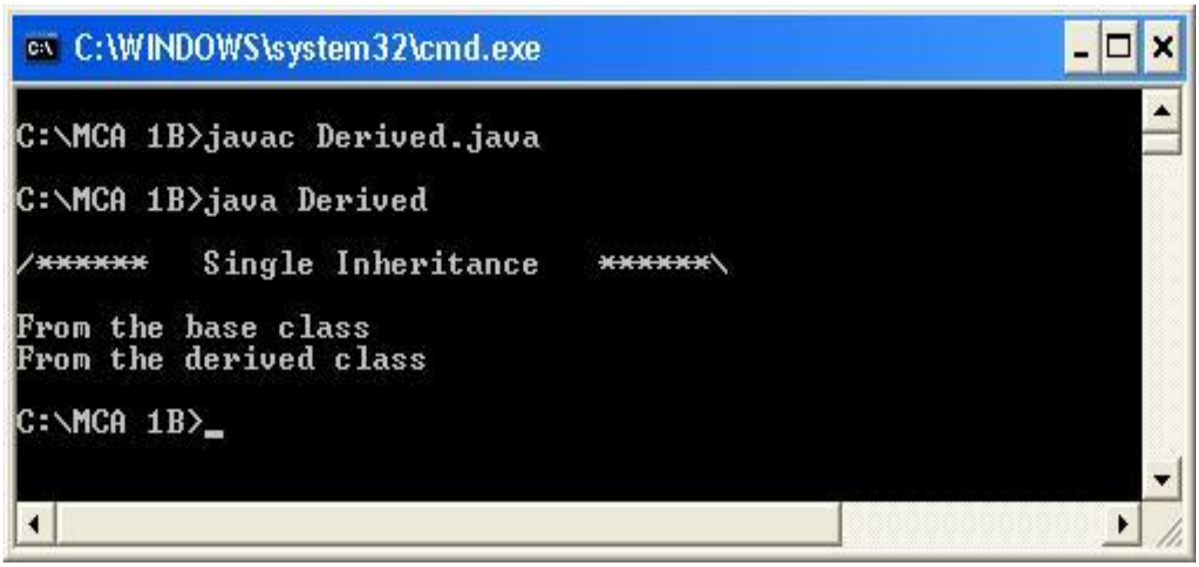
Preamble and Aim.

Write a java Program to Demonstrate the concept of Inheritance

Program:-

```
class  Base
{
    void fun1()

    {
            System.out.println ("From the base class");
    }
}
class Derived extends Base
{
    void fun2()
    {
    fun1();
    System.out.println("From the derived class");
    }
    public static void main(String[] args)
    {
            Derived d=new Derived ();
            System.out.println ("\n/******   Single Inheritance   ******\\\n");
            d.fun2 ();
    }
}
```

OUTPUT:-



```
C:\MCA 1B>javac Derived.java

C:\MCA 1B>java Derived

/******   Single Inheritance   ******\

From the base class
From the derived class

C:\MCA 1B>_
```

EXPERIMENT-4

Preamble and Aim.

Write a Java program to Demonstrate the concept of Multiple Inheritance

Program:-

```
interface A
{
    Public void display1();
}
interface B
{
    Public void display2();
}

class Cimplements A,B
{
    Public void display1()
    {
            System.out.println("implementation method of interface A);
    }
  Public void display2()
    {
            System.out.println("implementation method of interface B);
    }

    public static void main(String[] args)
    {
            C d=new C();
            System.out.println("\n/******   Multiple Inheritance   ******\\\n");
            d.display1();
            d.display2();

    }
}
```

OUTPUT:-

```
******   Multiple Inheritance   ******
    implementation method of interface A
    implementation method of interface B
```

EXPERIMENT-5

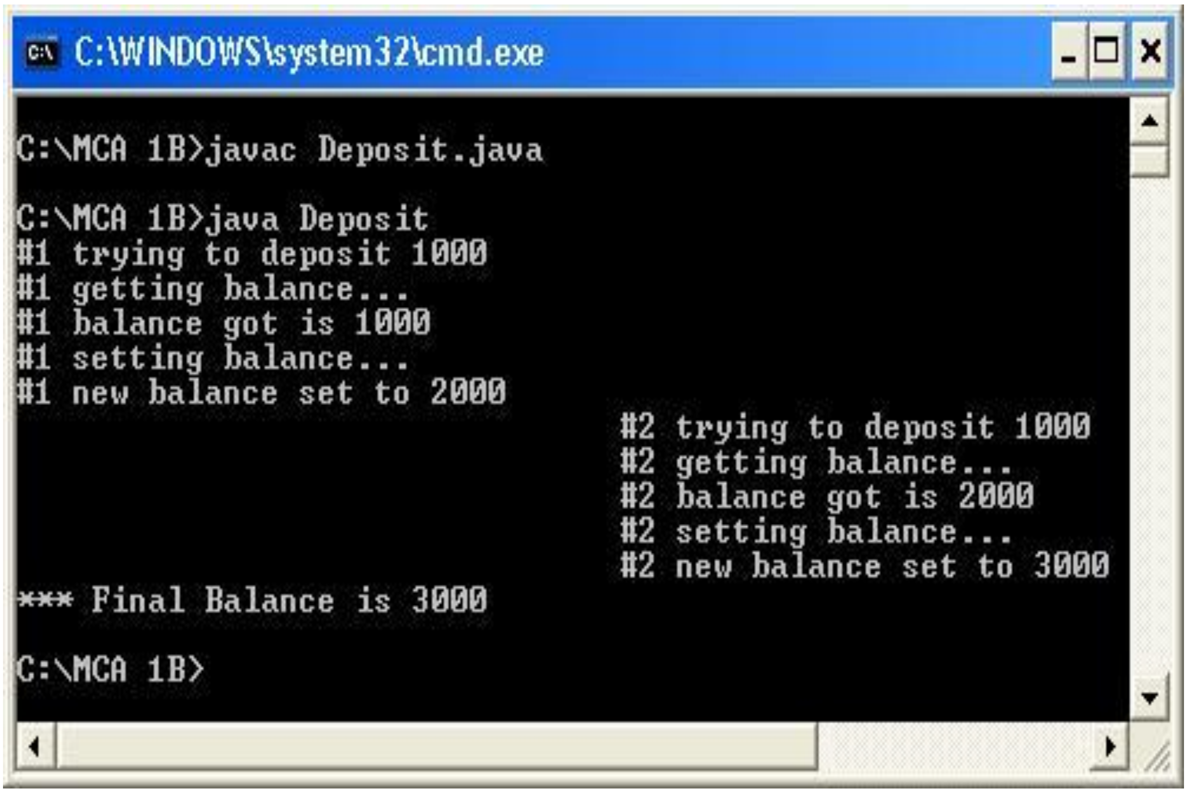
Preamble and Aim.

A program to illustrate thread synchronization

Program:-

```
import java.io.*;
public class Deposit
{
   static int balance = 1000;
   public static void main(String args[])
    {
       PrintWriter out = new PrintWriter(System.out, true);
       Account account = new Account(out);
       DepositThread first, second;
       first = new DepositThread(account,1000, "#1");
       second = new DepositThread(account, 1000, "\t\t\t\t#2");
       first.start();
       second.start();
       try
        {
           first.join();
           second.join();
         }
        catch(InterruptedException e)
         {

         }
        out.println("*** Final Balance is " + balance);
      }
}
class Account
{
  PrintWriter out;
  Account(PrintWriter out)
     {
         this.out = out;
     }
  synchronized void deposit(int amount, String name)
     {
        int balance;
        out.println(name + " trying to deposit " + amount);
        out.println(name + " getting balance...");
        balance = getBalance();
        out.println(name + " balance got is " + balance);
        balance += amount;
        out.println(name + " setting balance...");
        setBalance(balance);
        out.println(name + " new balance set to " + Deposit.balance);
     }
int getBalance()
     {
      try
       {
        Thread.sleep(5000);
       }
      catch(InterruptedException e)
       {
        }
     return Deposit.balance;
    }
 void setBalance(int balance)
    {
     try
      {
        Thread.sleep(5000);
     }
     catch(InterruptedException e)
       {
        }
     Deposit.balance = balance;
    }
}
class DepositThread extends Thread
{
  Account account;
  int depositAmount;
  String message;


  DepositThread(Account account, int amount, String message)
    {
            this.message = message;
            this.account = account;
            this.depositAmount = amount;
         }
        public void run()
         {
            account.deposit(depositAmount, message);
         }
    }
```

OUTPUT:



```
C:\MCA 1B>javac Deposit.java

C:\MCA 1B>java Deposit
#1 trying to deposit 1000
#1 getting balance...
#1 balance got is 1000
#1 setting balance...
#1 new balance set to 2000
                                #2 trying to deposit 1000
                                #2 getting balance...
                                #2 balance got is 2000
                                #2 setting balance...
                                #2 new balance set to 3000
*** Final Balance is 3000

C:\MCA 1B>
```

EXPERIMENT-6

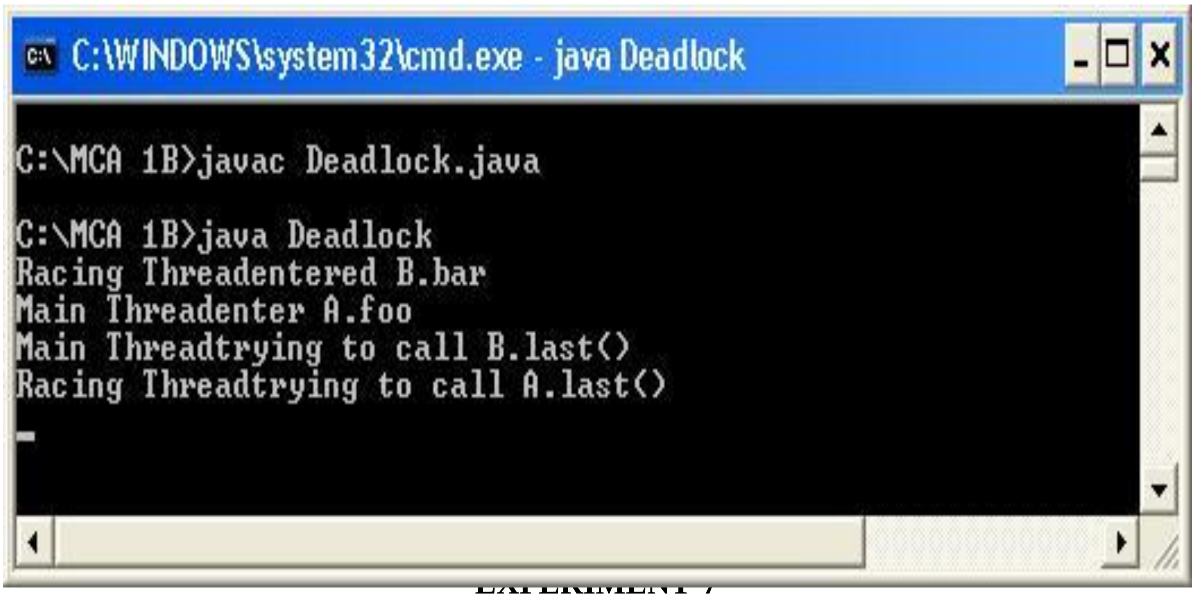
Preamble and Aim.

Write a Java program to illustrate Exception handling

Program:-

```
class A
{
    synchronized void foo(B b)
    {
            String name=Thread.currentThread().getName();
        System.out.println(name+"enter A.foo");
            try
            {
                    Thread.sleep(1000);
            }
            catch (Exception e)
            {
            }
            System.out.println(name+"trying to call B.last()");
            b.last();
    }
    synchronized void last()
    {
            System.out.println("Inside A.last");
    }
}
class B
{
    synchronized void bar(A a)
    {
        String name=Thread.currentThread().getName();
        System.out.println(name+"entered B.bar");
            try
            {
                    Thread.sleep(1000);
            }
            catch (Exception e)
            {
            }
            System.out.println(name+"trying to call A.last()");
            a.last();
    }


    synchronized void last()
    {
            System.out.println("Inside A.last");
        }
    }
    class Deadlock implements Runnable
    {
        A a=new A();
        B b=new B();
        Deadlock()
        {
             Thread.currentThread().setName("Main Thread");
             Thread t=new Thread(this,"Racing Thread");
             t.start();
             a.foo(b);
             System.out.println("Back in main Thread");
        }
        public void run()
        {
            b.bar(a);
            System.out.println("Back in other Thread");
        }
        public static void main(String args[])
        {
            new Deadlock();
        }
    }
```

OUTPUT:



```
C:\MCA 1B>javac Deadlock.java

C:\MCA 1B>java Deadlock
Racing Threadentered B.bar
Main Threadenter A.foo
Main Threadtrying to call B.last()
Racing Threadtrying to call A.last()
_
```

EXPERIMENT 7

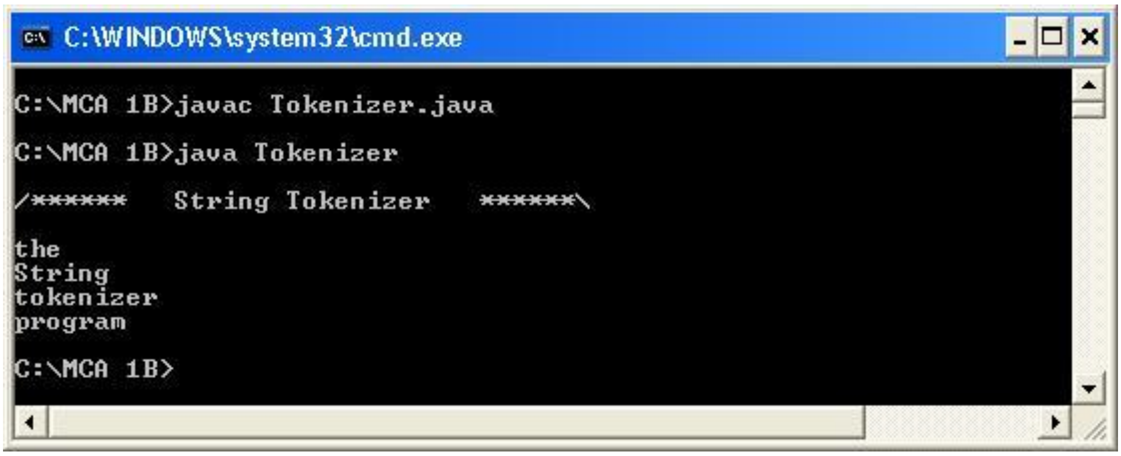
Preamble and Aim.

A program using string tokenizer

Program:-

```
 import java.util.*;
class Tokenizer
{
      public static void main(String[] args)
      {
                  System.out.println("\n/******   String Tokenizer   ******\\\n");
                  String s1="the+String-tokenizer+-program";
                  StringTokenizer st =new StringTokenizer(s1,"+-");
                  while(st.hasMoreTokens())
                              System.out.println(st.nextToken());
      }
}
```

OUTPUT:



```
C:\MCA 1B>javac Tokenizer.java

C:\MCA 1B>java Tokenizer

/******   String Tokenizer   ******\

the
String
tokenizer
program

C:\MCA 1B>
```

EXPERIMENT-8

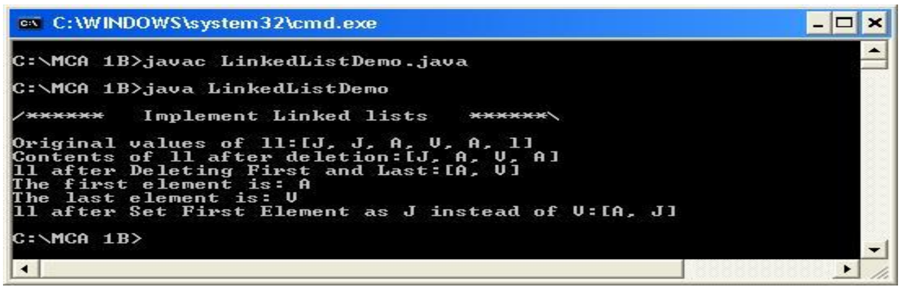
Preamble and Aim.

Write a Java program to demonstrate LinkedList

Program:-

```
import java.util.*;
class LinkedListDemo
{
      public static void main(String[] args)
      {
          System.out.println("\n/******   Implement Linked lists   ******\\\n");
                  LinkedList l=new LinkedList();
                  l.add("J");
                  l.add("A");
                  l.add("V");
                  l.add("A");
                  l.addFirst("J");
                  l.addLast("l");
                  System.out.println("Original values of ll:"+l);
                  l.remove("J");
                  l.remove("l");
                  System.out.println("Contents of ll after deletion:"+l);
                  l.removeFirst();
                  l.removeLast();
                  System.out.println("ll after Deleting First and Last:"+l);
                  System.out.println("The first element is: "+l.getFirst());
                  System.out.println("The last element is: "+l.getLast());
                  l.set(1,"J");
                  System.out.println("ll after Set First Element as J instead of V:"+l);
      }
}
```

OUTPUT:



```
C:\MCA 1B>javac LinkedListDemo.java

C:\MCA 1B>java LinkedListDemo

/******   Implement Linked lists   ******\

Original values of ll:[J, J, A, V, A, l]
Contents of ll after deletion:[J, A, V, A]
ll after Deleting First and Last:[A, V]
The first element is: A
The last element is: V
ll after Set First Element as J instead of V:[A, J]

C:\MCA 1B>
```

EXPERIMENT-9

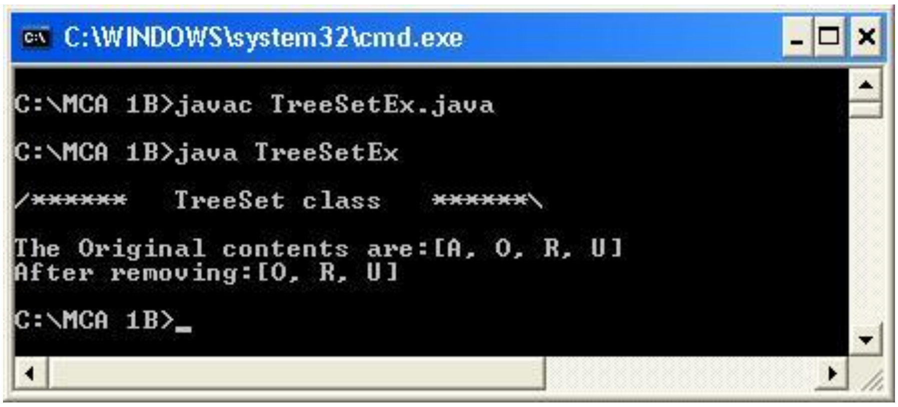
Preamble and Aim.

A program using TreeSet class

Program:-

```
import java.util.*;
class TreeSetEx
{
      public static void main(String[] args)
      {
         System.out.println("\n/******   TreeSet class   ******\\\n");
                TreeSet ts=new TreeSet();
                ts.add("A");
                ts.add("U");
                ts.add("R");
                ts.add("O");
                ts.add("R");
                ts.add("A");
                System.out.println("The Original contents are:"+ts);
                ts.remove("A");
                System.out.println("After removing:"+ts);
      }
}
```

OUTPUT:



```
C:\MCA 1B>javac TreeSetEx.java

C:\MCA 1B>java TreeSetEx

/******   TreeSet class   ******\

The Original contents are:[A, O, R, U]
After removing:[O, R, U]

C:\MCA 1B>_
```

EXPERIMENT-10

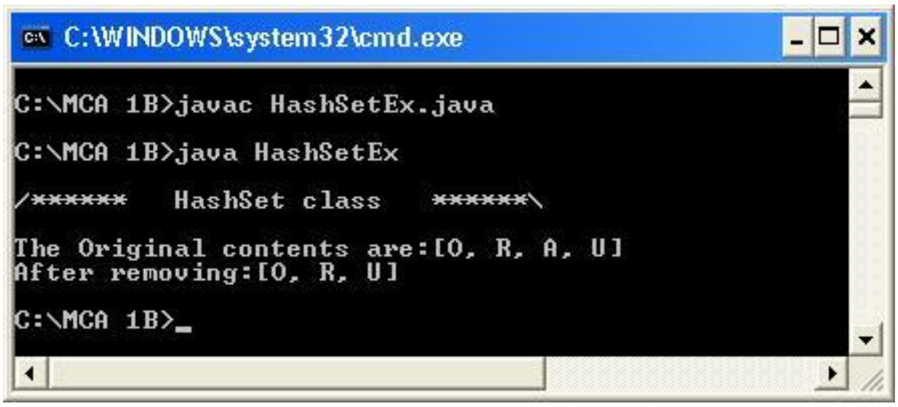
Preamble and Aim.

Write a java program to demonstrate HashSet class

Program:-

```
 import java.util.*;
class HashSetEx
{
      public static void main(String[] args)
      {
                  System.out.println("\n/******   HashSet class   ******\\\n");
                  HashSet hs=new HashSet();
                 hs.add("A");
                  hs.add("U");
                  hs.add("R");
                  hs.add("O");
                  hs.add("R");
                  hs.add("A");
                  System.out.println("The Original contents are:"+hs);
                  hs.remove("A");
                  System.out.println("After removing:"+hs);
      }
}
```

OUT PUT:



```
C:\MCA 1B>javac HashSetEx.java

C:\MCA 1B>java HashSetEx

/******   HashSet class   ******\

The Original contents are:[O, R, A, U]
After removing:[O, R, U]

C:\MCA 1B>_
```

EXPERIMENT-11

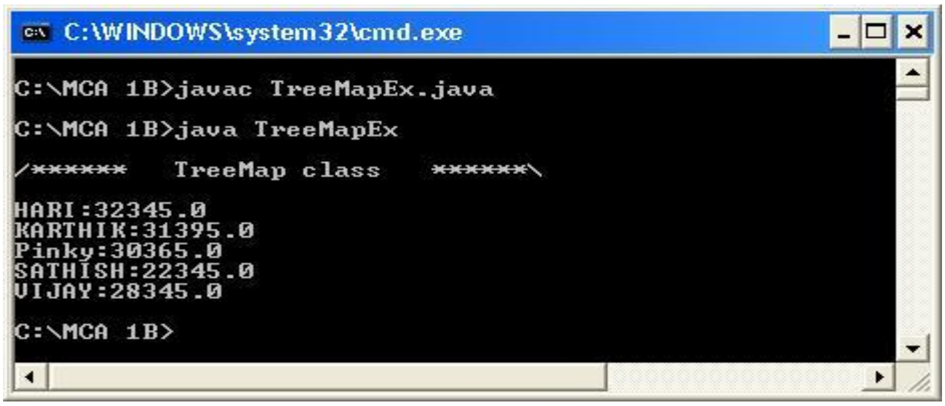
Preamble and Aim.

Write a java program to demonstrate map classes

Program:-

```
import java.util.*;
class TreeMapEx
{
      public static void main(String[] args)
      {
                  System.out.println("\n/******   TreeMap class   ******\\\n");
                  TreeMap tm=new TreeMap();
                 tm.put("HARI",new Double(32345.00));
                  tm.put("KARTHIK",new Double(31395.00));
                  tm.put("Pinky",new Double(30365.00));
                  tm.put("SATHISH",new Double(22345.00));
                  tm.put("VIJAY",new Double(28345.00));
                  Set s=tm.entrySet();
                  Iterator i=s.iterator();
                  while(i.hasNext())
                  {
                              Map.Entry m=(Map.Entry)i.next();
                             System.out.print(m.getKey()+":");
                             System.out.print(m.getValue());
                              System.out.println("");
      }
      }
}
```

OUTPUT:



```
C:\MCA 1B>javac TreeMapEx.java

C:\MCA 1B>java TreeMapEx

/******   TreeMap class   ******\

HARI:32345.0
KARTHIK:31395.0
Pinky:30365.0
SATHISH:22345.0
VIJAY:28345.0

C:\MCA 1B>
```

EXPERIMENT-12

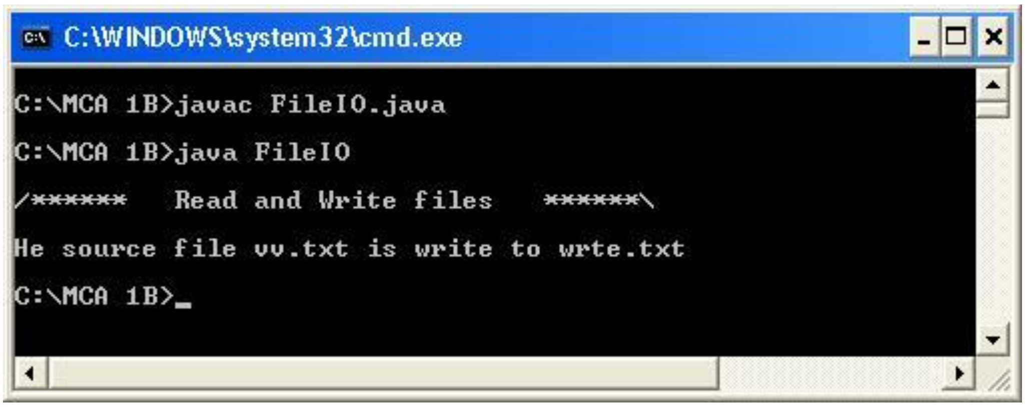
Preamble and Aim.

A program to illustrate the usage of Byte and character I/O Streams.

Program:-

```
import java.io.*;
class  FileIO
{
    public static void main(String[] args)
    {
            System.out.println("\n/******   Read and Write files   ******\\\n");
            try
            {
                    FileInputStream f1=new FileInputStream("vv.txt");
                    FileOutputStream f2=new FileOutputStream("wrte.txt");
                    int k;
                    while((k=f1.read())!=-1)
                    {
                            f2.write(k);
                    }
                    f2.close();
                    f1.close();
            }
            catch (IOException e)
            {
             System.out.println("The source file doesnot exist");
            }
            System.out.println("He source file vv.txt is write to wrte.txt");
    }
}
```

OUTPUT:



```
C:\MCA 1B>javac FileIO.java

C:\MCA 1B>java FileIO

/******   Read and Write files   ******\

He source file vv.txt is write to wrte.txt

C:\MCA 1B>_
```

EXPERIMENT-13

Preamble and Aim.

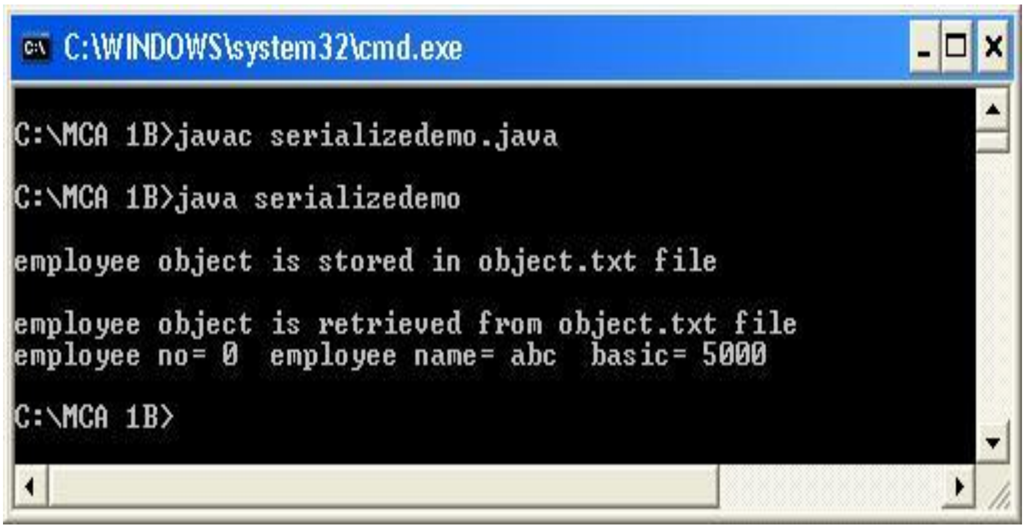
A program to illustrate the usage of serialization.

Program:-

```
import java.io.*;
class emp implements Serializable
{
        transient int empno;
        int basic;
        String ename;

        emp(int x,int y,String z)
        {
                empno=x;
                basic=y;
                ename=z;
        }
        public String toString()
        {
                return "employee no= "+empno+"  employee name= "+ename+"
basic= "+basic;
        }
}

class serializedemo
{
        public static void main (String []a) throws Exception
        {

ObjectOutputStreamos=newObjectOutputStream(newFileOutputStream("object.txt"))
;
          emp x=new emp(10,5000,"abc");
          os.writeObject(x);
          System.out.println("\nemployee object is stored in object.txt file");
          os.close();
  ObjectInputStream ois=new ObjectInputStream(new FileInputStream("object.txt"));
          emp e1=(emp)ois.readObject();
        System.out.println("\nemployee object is retrieved from object.txt file");
```

```
                System.out.println(e1);
                ois.close();
                }
        }
```

OUTPUT:



```
C:\MCA 1B>javac serializedemo.java

C:\MCA 1B>java serializedemo

employee object is stored in object.txt file

employee object is retrieved from object.txt file
employee no= 0  employee name= abc  basic= 5000

C:\MCA 1B>
```

EXPERIMENT-14

Preamble and Aim.

An application involving GUI with different controls,menus and event handling.

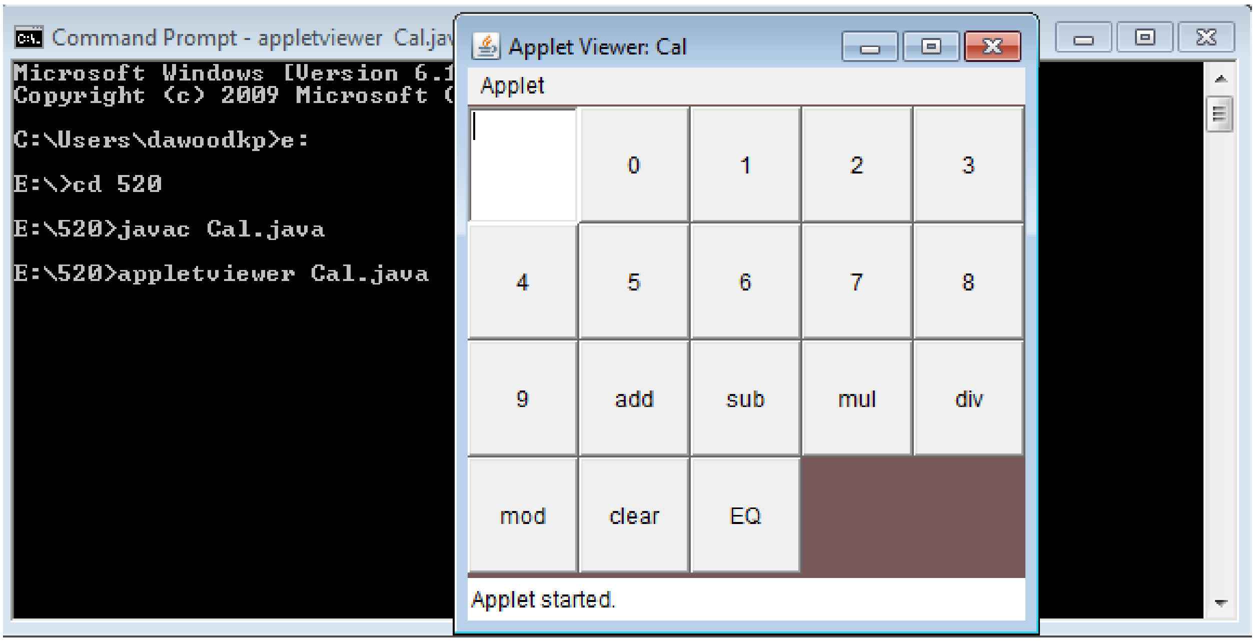
Program:-

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
/*
<applet code="Cal" width=300 height=300>
</applet>
*/
public class Cal extends Applet
implements ActionListener
{
        String msg=" ";
        int v1,v2,result;
        TextField t1;
        Button b[]=new Button[10];
        Button add,sub,mul,div,clear,mod,EQ;
        char OP;
        public void init()
        {
                Color k=new Color(120,89,90);
                setBackground(k);
                t1=new TextField(10);
                GridLayout gl=new GridLayout(4,5);
                setLayout(gl);
                for(int i=0;i<10;i++)
                {
                        b[i]=new Button(""+i);
                }
```

```
		add=new Button("add");
		sub=new Button("sub");
		mul=new Button("mul");
		div=new Button("div");
		mod=new Button("mod");
		clear=new Button("clear");
		EQ=new Button("EQ");
		t1.addActionListener(this);
		add(t1);
		for(int i=0;i<10;i++)
		{
			add(b[i]);
		}
		add(add);
		add(sub);
		add(mul);
		add(div);
		add(mod);
		add(clear);
		add(EQ);
		for(int i=0;i<10;i++)
		{
			b[i].addActionListener(this);
		}
		add.addActionListener(this);
		sub.addActionListener(this);
		mul.addActionListener(this);
		div.addActionListener(this);
		mod.addActionListener(this);
		clear.addActionListener(this);
		EQ.addActionListener(this);
	}
```

```
public void actionPerformed(ActionEvent ae)
{
        String str=ae.getActionCommand();
        char ch=str.charAt(0);
        if ( Character.isDigit(ch))
        t1.setText(t1.getText()+str);
        else
        if(str.equals("add"))
        {
                v1=Integer.parseInt(t1.getText());
                OP='+';
                t1.setText("");
        }
        else if(str.equals("sub"))
        {
                v1=Integer.parseInt(t1.getText());
                OP='-';
                t1.setText("");
        }
        else if(str.equals("mul"))
        {
                v1=Integer.parseInt(t1.getText());
                OP='*';
                t1.setText("");
        }
        else if(str.equals("div"))
        {
                v1=Integer.parseInt(t1.getText());
                OP='/';
                t1.setText("");
        }
        else if(str.equals("mod"))
        {
            v1=Integer.parseInt(t1.getText());
            OP='%';
            t1.setText("");
        }
        if(str.equals("EQ"))
        {
            v2=Integer.parseInt(t1.getText());
            if(OP=='+')
                result=v1+v2;
            else if(OP=='-')
                result=v1-v2;
            else if(OP=='*')
                result=v1*v2;
            else if(OP=='/')
                result=v1/v2;
            else if(OP=='%')
                result=v1%v2;
            t1.setText(""+result);
        }
        if(str.equals("clear"))
        {
            t1.setText("");
        }
    }
}
```

OUTPUT:



```
Microsoft Windows [Version 6.1
Copyright (c) 2009 Microsoft C

C:\Users\dawoodkp>e:

E:\>cd 520

E:\520>javac Cal.java

E:\520>appletviewer Cal.java
```

EXPERIMENT-15

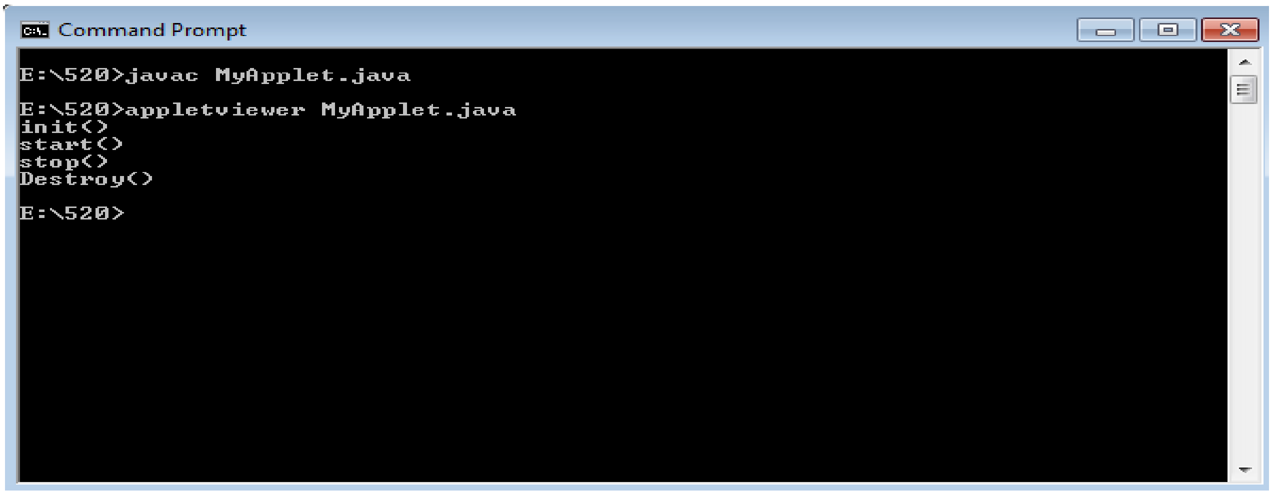
Preamble and Aim.

A program to implement an applet

Program:-

```
import java.applet.*;
/*
<applet code="MyApplet" width=300 height=500>
</applet>
*/
public class MyApplet extends Applet
{
 public void init(){
 System.out.println("init()");}
 public void start(){
 System.out.println("start()");}
 public void stop(){
 System.out.println("stop()");}
 public void destroy(){
 System.out.println("Destroy()");}
}
```

OUTPUT:-



```
E:\520>javac MyApplet.java

E:\520>appletviewer MyApplet.java
init()
start()
stop()
Destroy()

E:\520>
```

Lab Manual

Soft Skills

COURSE OBJECTIVES

1. Learn conversational skills
2. Learn reading strategies
3. Learn time management
4. Learn stress management
5. Learn career planning

COURSE OUTCOMES

1. Express conversational skills
2. Specify reading strategies
3. Perform time management
4. Perform stress management
5. Explore career planning

ACTIVITIES

1. Conversational skills (formal and informal)
2. Group discussion and interview skills
3. Making presentations
4. Listening to lectures, discussions, talk shows, news programmers, dialoguesfrom TV/radio/Ted talk/Podcast
5. Watching videos on interesting events on YouTube.
6. Reading different genres of tests ranging from newspapers to philosophicaltreatises
7. Reading strategies such as graphic organizers, summarizing and interpretation.
8. Writing job applications – cover letter – resume – emails – letters – memos –reports – blogs – writing for publications.
9. Civil Service (Language related) – Verbal ability.
10. Motivation – self image
11. Goal setting – managing changes
12. Time management – Stress management
13. Leadership traits
14. Team work
15. Career and life planning.
16. Multiple intelligences
17. Emotional intelligence
18. Spiritual quotient (ethics)
19. Intercultural communication
20. Creative and critical thinking
21. Learning styles and strategies.

TEACHING METHODS:

- To be totally learner-centric with minimum teacher intervention as the courserevolves around practice.
- Suitable audio/video samples from Podcast/YouTube to be used for illustrativepurposes.
- Portfolio approach for writing to be followed. Learners are to be encouraged to blog,tweet, text and email employing appropriate language.
- GD/Interview/Role Play/Debate could be conducted off the laboratory (in a regular classroom) but learners are to be exposed to telephonic interview and video conferencing.
- Learners are to be assigned to read/write/listen/view materials outside theclassroom as well for graining proficiency and better participation in the class.
- **REFERENCES:**

1. Business English Certificate Materials, Cambridge University Press.
2. Graded Examinations in Spoken English and Spoken English for Work downloadable

Presentation Skill

Presentations skills and public speaking skills have become the essential components for professional living. Effective presentation and public speaking skills are more very importantin the areas of business, training, teaching etc.

The helpful competencies for self-development and for social situations are developing theconfidence and capability for better presentations.

Presentation skills and public speaking abilities are not limited to quite a few only.Meticulous preparation and good practice would help any individual to develop the skill.

The formats and purposes of presentations can be very different, for example: oral (spoken), multimedia (using various media - visuals, audio, etc.), PowerPoint presentations, short impromptu presentations, long planned presentations, educational or training sessions, lectures, and delivering a talk on a subject to a group for pleasure. Even speeches at weddingsand eulogies at funerals are types of presentations.

Public speaking poses a threat to certain speakers. According to a survey taken, Presenting or speaking to an audience tops the Presentation or speaking to an audience regularly tops the list people's top fears, more than heights, flying or dying.

The key to managing and controlling anything is first to understand it, especially its causes. The cause of fear is a feeling of insecurity or uncontrollable threat. In the context ofpresentations and public speaking this is usually due to:

- lack of confidence
- lack of control (or a feeling of not having control) - over the situation, otherpeople (the audience) and our own reactions and feelings
- a bad memory or experience from our past

The effects of these are heightened according to the size of the audience, and potentially alsothe nature of the audience/situation - which combine to represent a perceived uncontrollable threat to us at a very basic and instinctive level (which we imagine in the form or critical judgment, embarrassment, humiliation, etc.).

The two big causal factors, low confidence and control, stem typically from inadequate preparation, rehearsal, and/or poor experience. Preparation and rehearsal are usually very manageable elements. It's a matter of making the effort to prepare and rehearse before the task is upon us. Presentations which do not work well usually do so because they have not been properly prepared and rehearsed. Preparation and knowledge, of subject and the presentation itself, are the pre-requisites for a successful presentation, which importantly produce confidence and control.

The basic sequence of actions for creating and preparing a presentation up to
the point ofactually delivering the presentation to an audience:

1. Define purpose
2. Gather content and presentation ideas
3. Structure the subject matter (sections, headings, order)
4. Develop how to present it (style, elements, props, equipment)
5. Prepare presentation (wording, design, materials and equipment)
6. Practice and rehearse (get feedback, refinement)
7. Plan venue, control the environment
8. Relax and prepare yourself - confidence and control

.

Topics for Presentation

The glut of paper products. TV violence.

Is the Fast-Food Industry Accountable Legally for poor health? Intelligence depends more on the environment than genetic factors. Should there be stronger limits on immigration?

Technological shift in phone design Environment vs. technology

Impact of technology on learning Learning does not eradicate ignorance How WIFI improved your life?

SEZ (Special Economic Zone)

Group Discussion

Group Discussion (GD) is a judging platform which examines the speaking ability of an individual in connection to a given topic. It is not a platform for you to fight your way through and dominate. Flexibility and getting along with the group is very important. It is expected to contribute meaningfully content that would help us to arrive at a conclusion

Important points to get through with flying colors:

1. Impressive look and Confidence are very important. It is very important to maintainpunctuality.

2. Doubts or clarifications regarding the subject/topic of the discussion must beclarified before the discussion begins.

4. It is very essential to retain originality and not to be arrogant. Gestures and bodylanguageare very important.as it decides the attitude of an individual.

5. Initiating the discussion is an advantage when the participant feels very confident of thetopic content.

6. Maintaining eye contact with the team members ensures the active participation in thegroup discussion.

7. A successful impact in the group discussion can be created by handling conflicts andarguments with a convincing solution.

8. it is very important to understand that the aim is not to speak often or for long periods butbe precise and clear with the points.

9.Topics need to be related to latest happenings.

Topics for Group Discussion

- NGOs - Do they serve peoples' interests or are they pressure groups?
- Role of women in development.
- Kids today are not what they used to be.
- Repeated elections - Should taxpayers pay for it?
- In India, the whole is less than the parts - Do we lack in team spirit?
- "Dot.com" companies - Is there room for everyone?
- Artificial Intelligence - Will man be ever replaced by machines?
- . If I were to choose my person of the millennium...
- . All world is a stage....
- Materialism - Have we sold our souls to the Devil?
- The rise and rise of feminist power.
- Are we unfit for Democracy?
- Survival tools for the new millennium.
- Examinations - has it killed education?
- Is E-Commerce the best thing for India?

Interview skills

A job interview is a conversation which occurs between a potential employer and a job applicant. During the job interview, the employer has the opportunity to appraise the applicant's qualifications, It is an important part of the process of applying for a job, and it may range in formality from a casual conversation to a series of serious discussions with an assortment of people working within the company.

Types of Interviews

Telephone -An employer initially calls a candidate over phone for an interview. Based on the essential criteria successful applicants are usually invited to the next level of one-to-one interview.

Video - this type of interview is increasingly popular for graduate roles in sales, media and marketing. They are usually held during the initial screening process.

One-to-one - Face-to-face encounter with one interviewer, after the organization decides that you've got what it's looking for. They're usually formal, but can also take place over lunch. You could also be interviewed by different people at different times.

Panel - Similar to one-to-one interviews, except two or more people – often from different parts of the organization – will be assessing you at the same time.

Group - Multiple candidates are interviewed together. They're asked questions in turn, or discuss certain topics.

Assessment centers - These involve tasks including presentations, written tests, role-playsand in-tray exercises. They're used to assess a candidate's performance in a range of situationsand last from one to three days.

Interviews require much research and planning. Generally, the following must be looked into when preparing for interview.

- anticipate potential questions and prepare answers accordingly.
- consider how to explain problematic aspects of your CV, such as leaving an employer.
- contact your references, alerting them that you'll be interviewed and that they mayreceive a call.
- understand the role that you are applying for by revisiting the job description, identifyingwhat skills, interests and experiences the employer is looking for.
- prepare questions for clarifications with the interviewer.
- read the organization’s website, social media profiles and key literature (e.g. business plan, financial reports and corporate social responsibility strategy), ensuring that you are prepared to share your views and ideas.
- research the news, trends, competitors, history and opportunities of the organization

How to make a good impression

- Answer questions clearly and concisely.
- Ask relevant, thought-provoking questions at appropriate moments, as this can show thatyou're genuinely interested in the role and really listening to the interviewer.
- Avoid talking about any personal problems.
- Be as enthusiastic as possible.
- Be well-mannered with any staff that you meet before the interview.
- Display positive body language, speaking clearly, smiling frequently and retaining eye-contact.
- Don’t badmouth any previous employers.
- Give a firm handshake to your interviewer(s) before and after.
- Inform your interviewer(s) that you're available to answer any follow-up questions.
- Let your personality shine.
- Relax and sit naturally, but without slouching in your chair or leaning on the desk.
- Show your hands, as this is a sign of honesty.
- Wear smart business attire with comfortable, polished shoes.

Tips to overcome fear

- pause before answering a difficult question to give yourself thinking time, or ask for clarification if, at first, you are unsure what the question means.

- put everything into perspective; remind yourself that the worst thing that can happen isnot getting the job.
- Take deep breaths and do not speak too quickly.
- take notes with you, write down cues to highlight examples that you want to draw upon.
- think about positive and happy experiences before the interview starts, and visualizeyourself in complete control during the interview.

Basic Interview Questions

Tell me about yourself. What are your strengths? What are your weaknesses? Why do you want this job?

Where would you like to be in your career five years from now?What's your ideal company?

What attracted you to this company?Why should we hire you?

Do you have any to me?

Behavioral Interview Questions

Give me an example of a time that you felt you went above and beyond the call of duty atwork.

Can you describe a time when your work was criticized?What is your greatest failure, and what did you learn from it?

What irritates you about other people, and how do you deal with it?

What was the most difficult period in your life, and how did you deal with it? Give me an example of a time you did something wrong. How

did you handle it? What irritates you about other people, and how do
you deal with it?

What's the most difficult decision you've made in the last two years and how did you come tothat decision?

Describe how you would handle a situation if you were required to finish multiple tasks bythe end of the day, and there was no conceivable way that you could finish them.

Writing Blogs

Keep your paragraphs short and sweet. Long paragraphs would create monotony.

Do not make spelling mistakes. To avoid making mistakes it's best to nip it in the bud – during the writing phase. Be sensitive to spelling mistakes like there and there, it's and its, form and from.

Do not use txt talk. Txt talk is the language used by people on mobile phones, which will save character space and make writing easier and quicker. It has its place and used on mobilephones, but not on a blog. It makes your blog harder to read.

Proof read your blog post after a day or two. When you have just written a blog post you cannot look upon it with a cold eye. The text is still fresh in your mind and you will have more trouble reading your text without skim reading. If you look at it a few days later you will have to read every line again, and you will pick up more of your errors that you missedthe first time.

Check for contradictions. Do not forget to check the message/content of your blog post. Somepeople will contradict themselves in various minor or glaring ways and be completely unaware of it because they proofread their texts without noticing the theme or content.

Write about things you're passionate about. Good blogger starts with the content. What does that mean? You need to have something to say before any other part of the writing can begin.Write about something you are passionate about. If you like sport – write about sport. If you are passionate about tech and apps then write about them.

Break your text apart. When you proofread it Cut your text into relevant sections, and then proofread each section individually. Proofreading a whole document can be daunting, and bythe middle or end you will become less diligent and more forgiving.

You can avoid this by proofreading in sections. Make sure one section is perfect beforestarting with a new one.

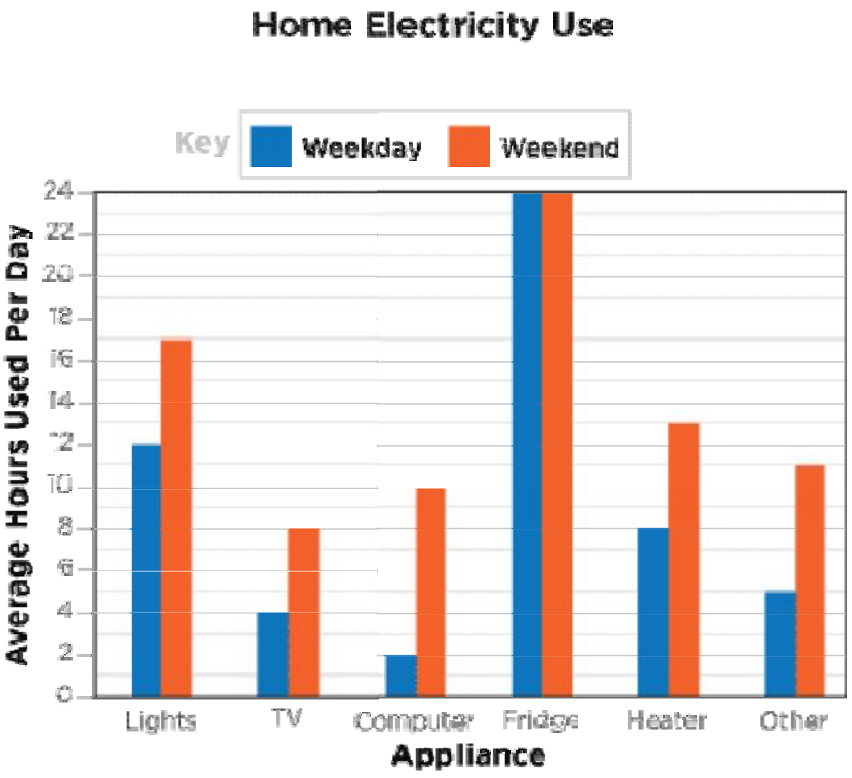
Writings a Tweet

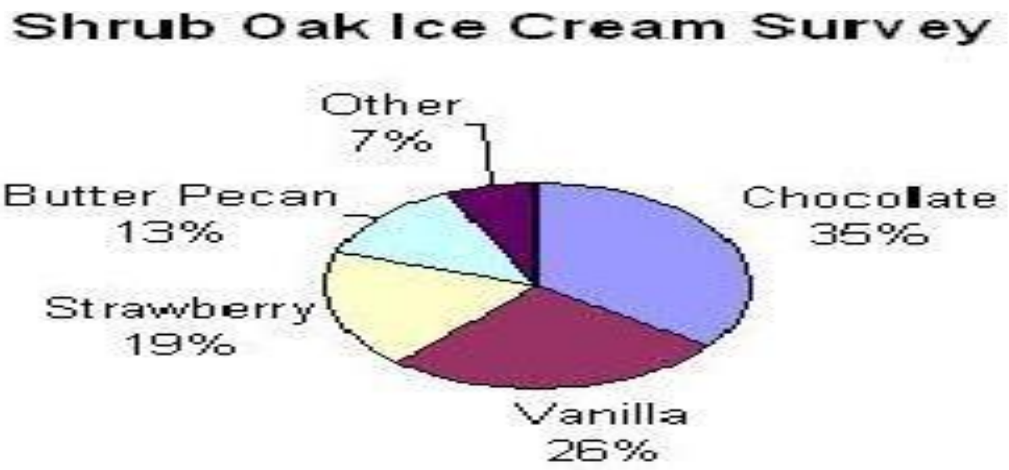
Twitter was never designed to be a social network but now it has become one.
In fact, it was designed initially in 2006 for individuals as an SMS type service to communicate with a small group at the podcasting company Odeo. In these many years, over140 million people are active users and generate over 340 million tweets daily.

12 Keys to Success on Twitter

- Success on Twitter takes time. It requires patience and persistence. True success takeslonger and it takes discipline and commitment.
- Quality and size of the content is very important for a tweet.
- It is very essential to build a targeted tribe on Twitter.
- This is where it starts to scale when you not only have a big following but a tribe thatwants to hear what you have to say and links that take them to the valuable content that they want to read and view, whether it is yours or another trusted source.
- An enduring presence on Twitter can be sought by sharing content that is useful and helpful regularly. We need to Aim to educate, entertain and inspire and not just inform.
- Tweeting great content is essential.

Interpretation of pictorial representations





Writing an SMS

A common way of modern communication is text messaging ie, sending an SMS (Short Message Service) message, with 160 characters or less. Texting in today's world (2002) has become one of the most popular ways of communication. The rapid growth of SMS text messaging was due to the cheap access to the networks, the decreasing prices of mobile phones and the popularity it had with the youth market as the latest 'must-have' accessory. With the maximum number of characters in a message being 160, people have learned ways of shortening words so that they can say more in one message.

Things to be remembered while sending an official message

- Don’t use texting abbreviations – most mobile users don’t understand them.
- Keep it short – A single text is 160 characters (although you can send multi-part textmessages).
- Tell them who you are – Consumers don’t know where the message is coming fromunless you mention it.
- Tell them what to do – A strong call to action is the key to increasing theeffectiveness of your messaging.

Worksheet 1

Idioms and Phrases

For each definition below, two of the idioms (a–c) have been invented, but one is correct.Can you identify the correct idiom in each case?

1. If someone looks very angry, we can say he/she has

   a) a face like thunder.

   b) A face like the moon.

   c) A face like a camel.

2. Work that we do because we enjoy it or because we think it is worthwhile, notbecause it gives us a financial reward, can be described as

   a) A job from the stars.

   b) Work of the heart.

   c) A labour of love.

3. If we think something will certainly happen but we don't know when, we can say it is

   a) A decision by time.

   b) A question of time.

   c) A present from time.

4. Another way of saying that something happens very quickly is that it happens

   a) as fast as a mouse.

   b) as rapid as a rabbit.

   c) as quick as a flash.

5. A person who can be trusted to do something in a way that won't cause any harm ortrouble can be described as

   a) a good pair of feet.

   b) a safe pair of hands.

   c) a warm pair of gloves.

Worksheet

Idioms

For each idiom below, identify the correct definition in each case?

1. What does it mean if you 'lose face'?

   a) You are no longer respected by people.

   b) You start showing physical signs of getting older.

   c) Your relationship with a very close friend starts to become more distant.

2. What does it mean if you 'lose your head'?

   a) You start becoming less anxious.

   b) You start becoming less powerful.

   c) You becomesoupsetthatyoustopthinkingclearlyorbehavinginasensibleway.

3. After someone has been successful in something, what does it mean if the success 'goesto their head'?

   a) They start behaving more intelligently.

   b) They start thinking they are better or more important than they really are.

   c) They become determined to continue their success.

4. What does 'get cracking' mean?

   a) To start doing something or going somewhere immediately.

   b) To panic.

   c) To start behaving aggressively towards other people.

5. What is a 'yes man'?

   a) Someone who will usually lend money to people if they ask for it.

   b) Someone with a positive attitude who generally believes they can achieve their goals.

   c) Someone who always agrees with people inauthority, usually because they want those people to like them.

Worksheet 3

Subject and Verb Agreement

Choose the correct form of the verb that agrees with the subject.

1. Annie and her brothers (is, are) at school.

2. Either my mother or my father (is, are) coming to the meeting.

3. The dog or the cats (is, are) outside.

4. Either my shoes or your coat (is, are) always on the floor.

5. George and Tamara (doesn't, don't) want to see that movie.

6. Benito (doesn't, don't) know the answer.

7. One of my sisters (is, are) going on a trip to France.

8. The man with all the birds (live, lives) on my street.

9. The movie, including all the previews, (take, takes) about two hours to watch.

10. The players, as well as the captain, (want, wants) to win.

11. Either answer (is, are) acceptable.

12. Every one of those books (is, are) fiction.

13. Nobody (know, knows) the trouble I've seen.

14. (Is, Are) the news on at five or six?

15. Mathematics (is, are) John's favorite subject, while Civics (is, are) Andrea's favorite subject.

16. Eight dollars (is, are) the price of a movie these days.

17. (Is, Are) the tweezers in this drawer?

18. Your pants (is, are) at the cleaner's.

19. There (was, were) fifteen candies in that bag. Now there (is, are) only one left!

20. The committee (debates, debate) these questions carefully.

21. The committee (leads, lead) very different lives in private.

22. The Prime Minister, together with his wife, (greets, greet) the press cordially.

23. All of the CDs, even the scratched one, (is, are) in this case.

Worksheet 4

Proverbs

Read the following proverbs and relate their application to the exercise given below.

A bird in the hand is worth two in

the bushA friend in need is a

friend indeed.

A stitch in time saves nine.

Absence makes the heart grow

fonder.As you sow, so you

shall reap.

Beauty is only skin

deep. Better late, than

never.

Better safe than sorry

Don't count your chickens before they're

hatched.Don't cry over spilt milk.

Don't judge a book by its cover.

Don't throw the baby out with the bath water.

Early to bed and early to rise makes a man healthy, wealthy

and wise.Every cloud has a silver lining.

God helps those who help

themselves.Half a loaf is better

than no bread.

He who laughs last laughs

longest. Make hay while the

sun shines Necessity is the
mother of invention.
Never put off till tomorrow what you can
do today.One man's meat is another man's
poison. Out of sight, out of mind.Rome
wasn't built in a day.
The best way to a man's heart is through his
stomach.The end justifies the means.
The grass is always greener on the other side of
the fence.Too many cooks spoil the broth.
Two heads are better
than one.Waste not,
want not.
When in Rome, do as the Romans do.
You can lead a horse to water, but you cannot make
him drink.You can't make a silk purse out of a sow's
ear.
Let sleeping
dogs
lie.Hunger is
the best sauce.

EXERCISE: A

Which of the proverbs mean the same as these sentences?

1. It is not a good idea to decide if a thing is good or bad just by how it appears at first or by its outward appearance.

2. When a bad thing happens there is always a positive aspect to it.

3. If you don't see someone for a long time you like them better.

4. When you are in a strange place (temporarily) it is better to follow the local customs.

5. You should be happy with what you have got even if it is less than what you want.

6. In many areas in life if you do not confront a problem when it starts it can deteriorate rapidly therefore it is much better to act soon.

7. Don't become very upset by

trivialities.8.Don't take risks.

EXERCISE: 2

Decide which proverb could help you express yourself in the following situations.

1. You make an appointment with your chiropodist for 6pm. You arrive at 6.15. He complainsthat he has been waiting for 15 minutes. What would you say?

2. Your boss calls you into his office to ask you why you haven't finished the task that he askedyou to do earlier and he complains that you are slow. What would you say to him?

3. Your friend smokes 60 cigarettes a day. He has a bad cough and he is always whining (complaining) that he would like to give up smoking. You offer to pay for expensive residential anti-smoking therapy. He says that he doesn't want to do it because he will miss his favorite television programme. What do you think to yourself?

Worksheet 5

Reading Comprehension

In cobra country a mongoose was born one day who didn't want to fight cobras or anything else. The word spread from mongoose to mongoose that there was a mongoose whodidn't want to fight cobras. If he didn't want to fight anything else, it was his own business, but it was the duty of every mongoose to kill cobras or be killed by cobras.

"Why? "Asked the peace like mongoose, and the word went round that the strange new mongoose was not only pro-cobra and anti-mongoose but intellectually curious and against the ideals and traditions of mongoosism. "He is crazy," cried the young's mongoose's father. "He's sick, "said his mother. "He is a coward," shouted his brothers. "He's a mongo sexual," whispered his sisters.

Strangers who had never laid eyes on the peace like mongoose remembered that they hadseen him crawling on his stomach, or trying cobra hoods, or plotting the violent overthrow of Mongoodia. "I am trying to use reason and intelligence," said the strange new mongoose. "Reason is six-sevenths of treason," said one of his neighbors. "Intelligence is what the enemy uses," said another.

Finally, the rumor spread that the mongoose had venom in his sting, like a cobra, and he was tried, convicted by a show of paws, and condemned to banishment.

Moral:
Ashes to ashes, and clay to clay,
if the enemy doesn't get you your own folks may.

1. Cobras and mongooses are animals who

   a. get along well with each other

   b. live peacefully side by side

   c. wage war against

each other 2.The word

"spread "suggests that:

a. no mongoose was interested in the event.

b. the news was kept secret

c. they all told each other the news.

a. A coward is a person

i. who can easily control his fear?

ii. who likes to fight?

iii. who is not courageous?

b. "Had never laid eyes "means

i. hadn't seen him before.

ii. had already seen him.

iii. had been watching him for a long time.

c. Strangers remembered him although:

i. they had never laid eyes on him

ii. they had actually seen him

iii. they thought they'd never met him.

d. If this particular mongoose had been seen crawling on his stomach it might mean that:

i. he was like a snake and had no legs.

ii. it was his only way of moving.

iii. he was trying to imitate cobras.

e. "Plotting the overthrow" suggests:

i. conspiring against the existing government.

ii. agreeing with the present leaders' policy.

iii. accepting the present state of affairs

f. Mongoodia is meant to be

i. A politicalparty

ii. a country

iii. a synonym of mongoosism

g. Reason is six-sevenths of treason implies that:

i. by using reason, the peace like mongoose is very close to being a traitor tohis ownfolks.

ii. The neighbor was only interested in how the words sounded.

iii. You can always create new words by adding a letter.

h. Convict" and "condemn" belong to the vocabulary of

i. religion

ii. politics

iii. justice

i. To declare the accused mongoose guilty, the jury voted by

i. putting up their hands.

ii. telling the judge.

iii. clapping their hands.

j. In this text, "own folks “contrasts with

i. ashes

ii. enemy

iii. clay

k. What does NOT belong to the ideals and traditions of mongoosism

i. killing cobras.

ii. using reason and intelligence.

iii. being intellectually curious.

l. The story and its moral:

   i. are purely fictional and have no symbolical meaning.

   ii. are meant to reflect the relationships between human beings.

   iii. can only apply to the world of cobras and mongooses.

m. In this text, reason and intelligence are related to:

   i. family relationships.

   ii. the idea of justice.

   iii. revolution

Cloze Test

I _______the first flat we lived in_________our marriage very well. I was not

________much, and Radha was_________a student, so our weekly income was very small,

and we simply________to find a cheap flat about. Looking over lots of places, we _________

found one that we could_________and it was not too uncomfortable. There was a big sitting

room and a tiny kitchen, just big_________for one___________to get__________.

1. (a) Recollect (b)Remember (c) Forget (d)Decorate (e) Furnish

2. (a) Before (b)During (c) In (d)After (e) For

3. (a) Obtaining (b)Earning (c) Paying (d)Gaining (e) Collecting

4. (a) Just (b)Hardly (c) Firmly (d)Fast (e) Chiefly

5. (a) Have (b)Had (c) Will have (d)Have had (e) Would have

6. (a) Would (b)Have (c) Will (d)Had (e) Could

7. (a) Sell (b)Rent out (c) Give away (d)Afford (e) Use

8. (a) About (b)Adequate (c) Efficient (d)Afford (e) Enough

9. (a) Group (b) Family (c) Personal (d) Get-together (e) Party

10. (a) About (b)By (c) Along (d) Down (e) In

Vocabulary

BuildingI Attitudes of a modern

secretary

Match each statement with the proper term:

1. knowing how to keep a secret a. sense of humor

2. just being there when it's necessary b. confidentiality

3. adapting oneself to the situation at hand c. computer skills

4. having a strong belief in one's own capacities d. organizational
skills

5. ability to work together effectively with many

different sorts of people e. availability

6. efficient in setting up events, arranging meetings

and solving practical problems f. self-confidence

7. on the look-out for new developments, keen to

master new skills g. flexibility

8. appreciates a good joke, shares a laugh with

colleagues h. teamwork

9. masters at least one word processing

program;spreadsheets and databases

have little secrets

for him/her i. loyalty

10. will defend his/her boss and/or company, no

matter what happens j. eagerness to learn

II Choose the correct word to fill in the blank

1. Do you have this shirt in a bigger_________?

a. Area

b. Size

2. Why is this item so much more_________than the other one?

a. expensive

b. cost

3. Can you_________that I will receive the item by next Friday?

a. warrantee

b. guarantee

4. Is the item_________in any way?

a. damaging

b. damaged

5. I'll have to__________(= check) that.

a. verify

b. prove

6. I'll_________(= contact you) by the end of the day.

a. get you back

b. get back to you

7. Unfortunately, that item is ________.

a. sold-out

b. outsold

8. Do you only accept payment_________Paypal?

a. through

b. in

9. I feel a little uneasy about_________my credit card information.

a. donating

b. sharing

10. I'm sorry for not_________. I was very busy.

a. talking

b. responding

Common Errors in English

1. There is a rise of temperature today.

2. **I** am afraid about dogs.

3. The car crashed with a bus.

4. Fill the glass by water.

5. The car collided into a bus.

6. I am confident in my success.

7. I am complaining about a headache.

8. I cannot agree with your idea.

9. He died from cancer.

10. I am looking forward to get a good job.

Worksheet 6

Reading Comprehension

Horse owners who plan to breed one or more mares should have a working knowledge of heredity and know how to care for breeding animals and foals. The number of mares bred that actually conceive varies from about 40 to 85 percent, with the average running less than 50 percent. Some mares that do conceive fail to produce living foals. This means that, on average, two mares arekept a whole year to produce one foal, and even then, some foals are disappointments from the standpoint of quality.

By careful selection, breeders throughout history have developed various kinds of horseswith a wide variety of characteristics to suit many different needs. The Great Horseof the Middle Ages, for example, was bred for size and strength to carry a heavily armored knight. The massive horses of such breeds are often called "cold blooded." The Arabs bred lithe desert horses that weresmall and swift. These animals are often referred to as "hot blooded. " Cross-breeding of hot- blooded and cold-blooded horses for certain characteristics produced breeds ranging fromriding horses-to-draft horses. The Thoroughbred is considered by many to be the highpoint of elegance and fine selective breeding. Many persons mistakenly apply the name Thoroughbred to any purebred horse. But a Thoroughbred is a distinct breed of running horses that traces its ancestry through the male line directly back to three Eastern stallions: the Byerly Turk, the Darley Arabian, and the Godolphin Barb. For convenience the breeds of horses are often divided into three major groups: (1) ponies, (2)
heavy, or draft horses, and (3) light horses.

1. Which of the following is not an example of an Eastern stallion?
   (A) Byerly Turk
   (B) Darley Arabian
   (C) Thoroughbred
   (D) Godolphin Barb

2. Which of the following was NOT a characteristic of the Great Horse of the Middle Ages?

(A) Large size

(B) Swiftness

(C) Strength

(D) "Cold-bloodedness"

3. It can be inferred from the passage that cold-blooded and hot-blooded horseswere cross-bred for what reason?

(A) Such cross-breeding was a safer means of reproduction.

(B) Cross-bred horses were preferred by Arabs.

(C) By cross-breeding, horses with desirable mixed characteristics could be produced.

(D) Cross-breeding produced Thoroughbred horses.

4. In line 11, "lithe" most nearly means

(A) graceful.

(B) clumsy.

(C) massive.

(D) bulky.

5.Which of the following is NOT one of the major divisions of horse breeds?

(A) Draft horses

(B) Ponies

(C) Foals

(D) Light horses

6. According to the passage, which of the following horses is considered to be the finest purebred?

(A) Darley Arabian

(B) Thoroughbred

(C) Godolphin Barb

(D) Byerly Turk

7. To conceive is to

(A) become sick.

(B) become pregnant.

(C) die.

(D) be born.

8. A foal is a

(A) male horse.

(B) female horse.

(C) old horse.

(D) baby horse.

9. The average amount of mares bred which actually conceive is less than whatpercent?

(A) 40

(B) 85

(C) 50

(D) 75

10. A mare is a

(A) male horse.

(B) baby horse.

(C) female horse.

(D) old horse.

Cloze Test

Bill is a____(1)____young man. Three years ago, when he finished middle school, he found

(2) work in a shop. Usually, he works until nine o'clock in the evening. When he (3) home, he always feels very__(4)___. After a quick supper he goes to bed and falls asleep. So he never makes any noise. His grandfather is very satisfied with him. One day, he went back from work a little earlier.

____(5)____he met Mary, one of his classmates. They____(6)_____very happy. Bill asked Maryto his house, the girl agreed. He brought her some fruit and drinks. They talked about their school, their teachers, their classmates and their future. They talked for a long time. "Have a look at your watch, please," said the girl. "what's the time now?" "Sorry, there is (7)_my watch," said Bill.

"where is__(8)___?" "I___(9)___it at home." Bill thought for a moment and found a way.He began to stamp his foot on the floor. The sound woke his grandfather up. The oldman shouted

___(10)__."It's twelve o'clock at night, Bill, what are you still jumping upstairs?"

1.twenty-year-old   twenty-years-old   twenty-years old   twenty years old

2.a   an   the   _

3. gets   arrives at   reaches to   arrives in

4. happy   tired   thirsty   hungry

5. On his way to home   in his way home   On his way home   By his wayhome

6. 6.were both   both were   all were   were all

7. wrong something with something wrong with wrong something to anything wrong with

8. you   your   yours   yoursel f f

9. forgot   brought   putted   left

10.upstairs   in upstairs   downstairs   in downstairs

Vocabulary Building

1. A ridge on the surface of the sea. Also, what you might do when you see somebody youknow orsay goodbye.

   A. wave B. sediment C. Ocean D. coral

2. Water sport that involves riding waves with a board.

   A. surfing B. flotsam C. dolphin D. sediment

3. Hard covering of marine snails and other molluscs.

   A. seashell B. marine C. sediment D. surfing

4. Describes anything to do with the sea.

   A. buoy B. wave C. marine D. flotsam

5. Rock-like substance produced by tiny animals. Often forms reefs.

   A. flotsam B. coral C. sediment D. seashell

6. Fine material at the bottom of a liquid, for example mud on the sea floor.

   A. surfing B. coral C. ocean D. sediment

7. vast body of water.

   A. coral B. ocean C. flotsam D. sediment

8. Floating debris.

   A. flotsam B. wave C. buoy D. dolphin

9. An intelligent aquatic mammal that is often friendly to humans.

   A. sediment B. dolphin C. surfing D. flotsam

10. float anchored in water, often as a warning to boats

    A. ocean B. flotsam C. buoy D. Sediment

11. The pipe in my kitchen is______.

    a. running b. leaking

12. I got in an accident and_______my bumper.

    a. bruised b. dented

13. My phone battery____________.

    a. died b. ended

14. I fell and___________________my arm.

    a. dented b. bruised

15. I spilled some wine and_______the tablecloth.

    a. broke b. stained

16. Our washing machines________.

    a. broke down b. came off

17. I_________________________a tooth while I was eating dinner.

    a. chipped b. banged

18. We________gas on the way to work.

    a. ran out of b. ended up

19. I forgot my keys so now I'm______of my apartment.

    a. stuck out b. locked out

20. I________my head on the kitchen cupboard.

    a. bumped b. collided

Common errors in English

1. He entered into my room.
2. We discussed about the problem.
3. India and Pakistan entered an agreement.
4. Despite of the rain, we went out.
5. Inspite him being a gentleman, he behaves rudely sometimes.
6. Write with ink
7. Sit under the shade of a tree
8. She is good in mathematics
9. Send on my address
10. I like to go in train

Worksheet 7

Reading Comprehension

Frank Tower, have you ever heard of him? He is the questionable figure who supposedly survived three doomed ships in the 1900's. Some consider him one of the luckiest men alive. He was touted to be a middle-aged fireman in the engine room. Some considered him an ordinary, hardworking person, but he had the ability to avoid dying in some of the most horrendous ocean liner accidents ever recorded.

He was said to have once been a crew member on the *Titanic* at the time that the ship hit the iceberg. Two years later, he was working on the *Empress of Ireland* when she collided with the *Storstad*. Over one thousand people died in that disaster. He was then employed in May of 1915 on the *Lusitania* when it was hit by a U-20 torpedo. He apparently lived through that without a scratch as well. If you are beginning to doubt this man's existence, you are probably not to far from the truth. No records have been found ever listing a man by Frank Towers working on any of the three ships.

The legend of Frank Towers seems to be another case of an urban folk tale, humanity's desire to see triumph over a tragic situation. Fact or fiction, Frank Towers is one of the multiple characters that help color the history books.

1. According to the article, the *Lusitania* is a?

A. Airplane

B. Torpedo Boat

C. Ship

D. Train

2. In this statement from the article, "He was touted to be a middle-aged fireman in theengine room." Touted means?

A. Publicized

B. Demeaned

C. Set-up

D. Sighted

3. This passage sheds doubt on

A. The fact that three ships sank.

B. The thought that America was at war with other countries.

C. Frank Towers causing the ships to sink.

D. Urban folk tales are not always based on real people.

4. The title of this article could be

A. “Frank Towers, a Man of Mystery”

B. “Surviving the Impossible”

C. “The Legend of Frank Towers Debunked”

D. “How to Survive Doomed Ships”

5. According to this passage, which of the following phrases best captures the intent of thearticle?

A. A stitch in time saves nine.

B. Don’t believe everything you read.

C. Seeing is believing.

D. There are many layers of an onion.

6. The article states that urban folk tales are created because

A. They are fun to hear.

B. Humanity likes to see someone triumph over tragedy

C. They trick people into believing lies.

D. People do not remember details clearly.

Cloze Test

___(1)_was Jimmy's birthday, and he was five years old. He got a lot of birthday presents__(2)___his family. One of them was a drum. Of course, Jimmy liked thedrum very much. He always made a terrible (3)_with it, but his mother didn't

___(4)___. His father didn't,___(5)___. Because he was working in his office during the day, and when he got home in the evening, Jimmy was already in bed. But Mrs. Johnson, their_(6)_, didn't like to hear the sound of the drum all day long. One morning a few days___(7)___, she went to Jimmy's house____(8)____a knife. Jimmy was hitting the drum. "Hello, Jimmy," she said to the boy with a smile. "There is

___(9)___very interesting inside your drum. Do you want to see it? Here's a knife. Cut the drum____(10)____and let's find it." 1 - It That This Today 2 - of for from with 3 - voice shout answer noise 4 - like notice talk understand 5 - too either also still 6 - grandmother aunt neighbor teacher 7 - -before ago after soon 8 - by with in for 9 - bywith in for 10 - anything something nothing everything

1 - It	That	This	Today
2 –of	for	from	with
3 - voice	shout	answer	noise
4 - like	notice	talk	understand
5 - too	either	also	still
6 - grandmother	aunt	neighbor	teacher
7 - -before	ago	after	soon

8 - by with in for

9 - by with in for

10 - anything something nothing everything

Vocabulary Building

1. After you had your interview, it's a good idea to_____________(= to get in touch by

   saying thank you, asking about their decision, etc.) with the employer.

   a. follow up

   b. follow through

2. A good way to follow up is by sending a_____________note.

   a. thanks

   b. thank you

3. The interviewers were impressed with Jim's positive________ .

   a. attitude

   b. aptitude

4. P1: How's the dress code at your company? Is it formal? P2: No, it's________ .

   a. caustic

   b. casual

5. When the interviewer concludes the interview, offer a firm handshake and make eye

   ____________.

   a. contact

   b. connection

6. I looked at your resume and noticed that you have some_____________in youremployment hist

   a. missing time

   b. gaps

7. P1: I got a_____________from Google. P2: Really? Did you accept it?

   a. job proposal

   b. job offers

8. What's another word for a "job search"?

a. job mission

b. job hunt

9. Don't forget to look through the_____________in your newspaper.

a. job listings

b. job articles

10. P1: What are some of your companies provides? P2: Let's see... Afree dental plan,three weeks paid vacation, etc.

a. benefit

b. benefits

11. Unemployment_____________= money that is paid by the government to peoplewho do not have a job.

a. benefits

b. money

12. Did you quit or were you_____________?

a. hired

b. fired

13. My friend tried to commit unemployment_____________, but he was caught.

a. scam

b. fraud

14. I_____________for unemployment benefits last week.

a. filed

b. tried

15. My benefits are automatically_____________into my account every week.

a. deposited

b. given

16. People who receive benefits have to fill out a weekly_____________.

a. request

b. claim

17. I didn't receive my check this week. It must have gotten_____________.

a. lost in the mail

b. lost in mail

18. An_____________occurs when you receive (or pay) more money than you arelegally entitled to.

a. underpayment

b. overpayment

19. You have to report all your_____________on your weekly claim.

a. wage

b. wages

20. In legal terms, the "person making a claim" is the_____________.

a. claimant

b. claimer

Common Errors in English

21. The birds build nest on the trees

22. Congratulate him for his success

23. She is fond to talk

24. I complained to the doctor about my stomachache

25. Marry was married John

26. I have visited Niagara Falls last weekend.

27. The woman which works here is from Japan.

28. She’s married with a dentist.

29. She was boring in the class.

30. I must to call him immediately

Lab Manual
Operating Systems

1. OBJECTIVE AND RELEVANCE

OPERATING SYSTEMS:

To understand how a set of computing resources can be shared safely, efficiently, and effectively amongmany users, including the major concepts of modern operating systems and the relationship between the OS and the architecture of computer systems

- Introduction to Operating Systems
- Operating system structures
- Concurrent processes and programming
- CPU scheduling
- Deadlock
- Memory management and virtual memory
- File systems and storage management

Students will be able to understand the data structures and algorithms for the main components in amodern operating system through lecture sessions, and through lab sessions and assignments, to implement some of these components, and to reinforce between theory and practice.

2. OUTCOMES

OPERATING

SYSTEMS:

This Course provides the complete understanding of OS, functions and services in its full view. Along

with operating concepts this course gives knowledge about different existing operating systems such as UNIX, Windows, and LINUX. Operating systems are an essential part of any computer system. There are tremendous range and variety of computer systems for which operating systems are designed. These include single user workstations, personal computers, large mainframes and specialized machines like real systems. In spite of this variety and pace of change, certain fundamental concepts apply consistently throughout. To be sure, the application of these concepts depends on current state of technology and the particular application requirements. Perhaps one of the important reasons for studying operating systems is to learn how to extract the best performance from them. Understanding the reasons behind the developments of Operating systems gives us an appreciation for what an Operating system does and howit does it.

3. EQUIPMENT REQUIRED

Hardware

No. of System	:	60(IBM)
Processor	:	PIV™ 1.67 GH$_z$
RAM	:	512 MB
Hard Disk	:	40 GB
Mouse	:	Optical Mouse
Network Interface card	:	Present

Software

Operating System	:	Window XP
Application Software	:	Turboc C++
Office	:	Ms-Office-2007

4. CODE OF CONDUCT

- Students should report to the concerned lab as per the time table.
- Students who turn up late to the labs will in no case be permitted to do the program schedule for the day.
- After completion of the program, certification of the concerned staff in-charge in the observation book is necessary.
- Student should bring a notebook of 100 pages and should enter the readings/observations into the notebook while performing the experiment.
- The record of observations along with the detailed experimental procedure of the experiment in the immediate last session should be submitted and certified staff member in-charge.
- Not more than 3-students in a group are permitted to perform the experiment on the set.
- The group-wise division made in the beginning should be adhered to and no mix up of students among different groups will be permitted.
- The components required pertaining to the experiment should be collected from stores in-charge after duly filling in the requisition form.
- When the experiment is completed, should disconnect the setup made by them, and should return all the components/instruments taken for the purpose.
- Any damage of the equipment or burn-out components will be viewed seriously either by putting penalty or by dismissing the total group of students from the lab for the semester/year.
- Students should be present in the labs for total scheduled duration
- Students are required to prepare thoroughly to perform the experiment before coming to laboratory

5. SYLLABUS ANALYSIS

S.No.	Week No.	Name of the Experiment	Unit No.	Text / Reference Book	Page Nos
Operating System					
1	1	Simulate the following cpu scheduling algorithms: a) Round Robin b) SJF c) FCFS d) priority	Unit-2	T1,T2,R1,R2,R3,R4,R5	
2	2	Simulate the file allocation strategies: a) Sequential b) Indexed c) Linked	Unit-4	T1,T2,R1,R2,R3,R4,R5	
3	3	Simulate MVT and MFT	Unit-3	T1,T2,R1,R2,R3,R4,R5	
4	4	Simulate all File organization techniques a) Single level Directory b) Two Level c)Hierarchical d) DAG	Unit-4	T1,T2,R1,R2,R3,R4,R5	
5	5	Simulate Bankers Algorithm for Deadlock Avoidance	Unit-5	T1,T2,R1,R2,R3,R4,R5	
6	6	Simulate Bankers algorithm for Deadlock Prevention	Unit-5	T1,T2,R1,R2,R3,R4,R5	
7	7	Simulate all page replacement Algorithms a)FIFO b) LRU c) LFU	Unit-3	T1,R1,R2,R3	
8	8	Simulate Paging Technique of memory management.	Unit-3	T1,T2,R1,R2,R3,R4	

6. List of text book (OU Prescribed Text Books)

A) COMPUTER NETWORKS

TEXT BOOKS

T1: Operating System Principles, Abraham Silberchatz, Peter B. Galvin, Greg Gagne 8th Edition, Wiley Student Edition

T2: Operating System – Internals and Design Principles, W.Stallings 6Th Edition,Pearson

REFERENCE BOOKS

R1 : Operating Systems, Internal and Design Principles Stallings, 5th Edition 2005, PHI.

R2 : Operating System A Design Approach, Crowley, TMH.

R1 : Modern Operating Systems, Andrew S Tanenbaum, 2nd Edition, PHI.

R2 : Operating System, Milan Milenkovie, Tata McGraw Hill Edition.

R3 : Operating System A Modern Perspective, Gary Nutt, 2nd Edition, Low Price Edition

.

7. SESSION PLAN

S. No.	Week No.	Unit as per Syllabus	Activit y	Remarks
OPERATING SYSTEMS				
1	1	-	Introduction Basic c programs	PREREQUISITE
2	2	Unit-2	Simulate the following cpuscheduling algorithms: a) Round Robin b) SJF c)FCFS d) priority	OU
3	3	Unit-4	Simulate Paging Technique ofmemory management.	OU
4	4	Unit-3	Simulate Bankers Algorithmfor Deadlock Avoidance	OU
5	5	Unit-4	Simulate Bankers algorithmfor Deadlock Prevention	OU
6	6	Unit-5	Simulate the file allocationstrategies: a) Sequential b) Indexed c)Linked	OU
7	7	Unit-5	Simulate MVT and MFT	OU
8	8	Unit-3	Simulate all File organizationtechniques a) Single level Directory b) Two Level c)Hierarchical d)DAG	OU
9	9	Unit-3	Simulate all page replacementAlgorithms a)FIFO b) LRU c) LFU	OU

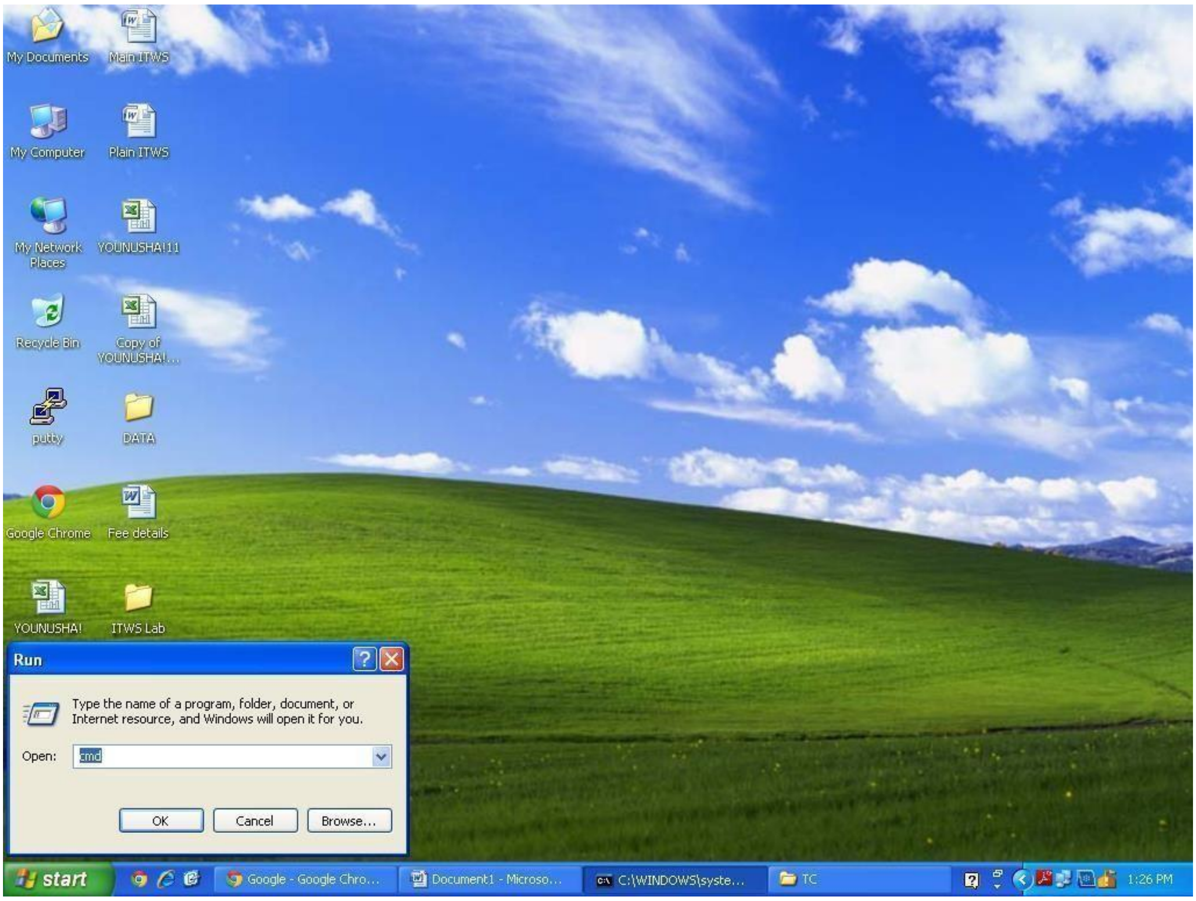
8. Each Experiment Write Up

8. 1 Introduction to Lab

Get familiar with the C Programming Language: If you don't already know it, in this lab you will get started on learning a new programming language. Although this is not meant to be a tutorial, you will get exposed to the most basic features of the language by reading and analyzing some basic programs.

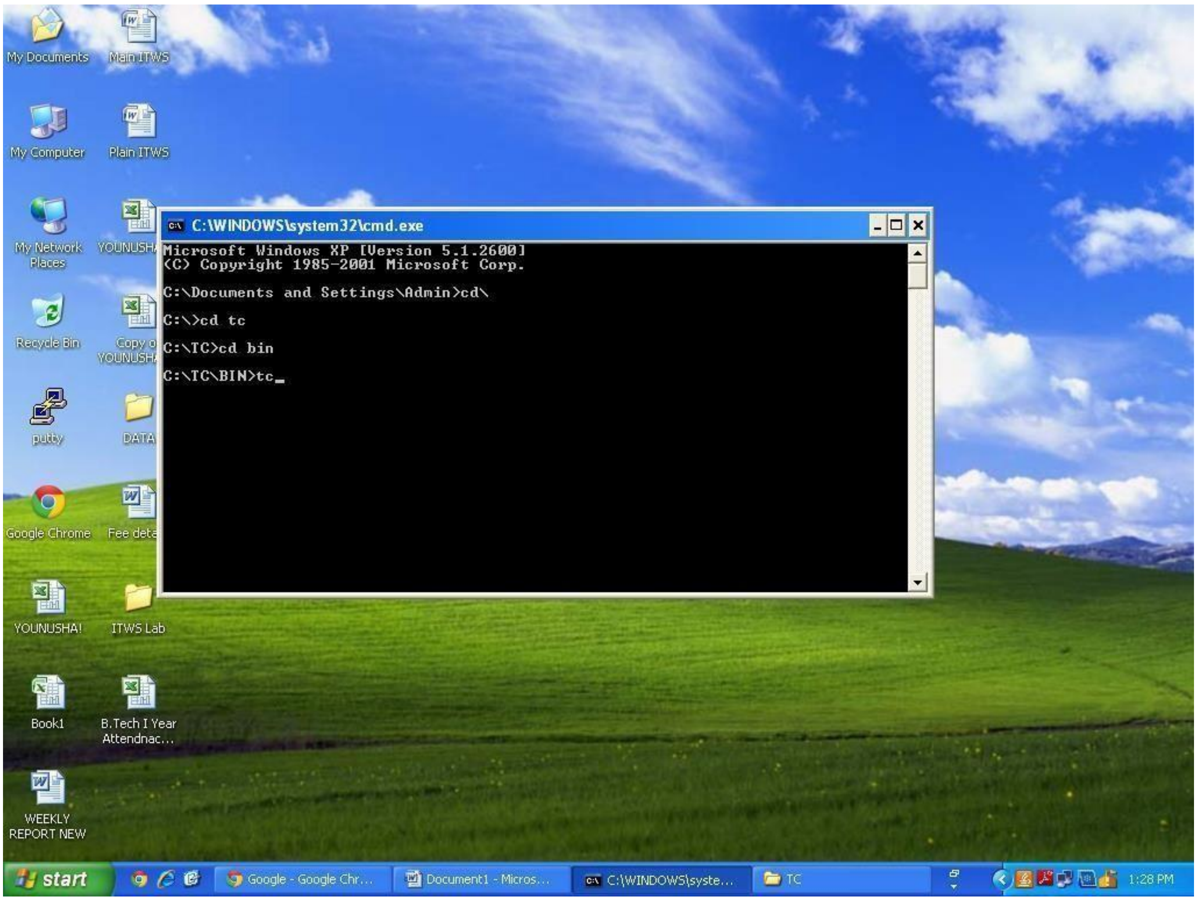
How to Open Software Screen Shots

How to Run C programs on Turbo C editor

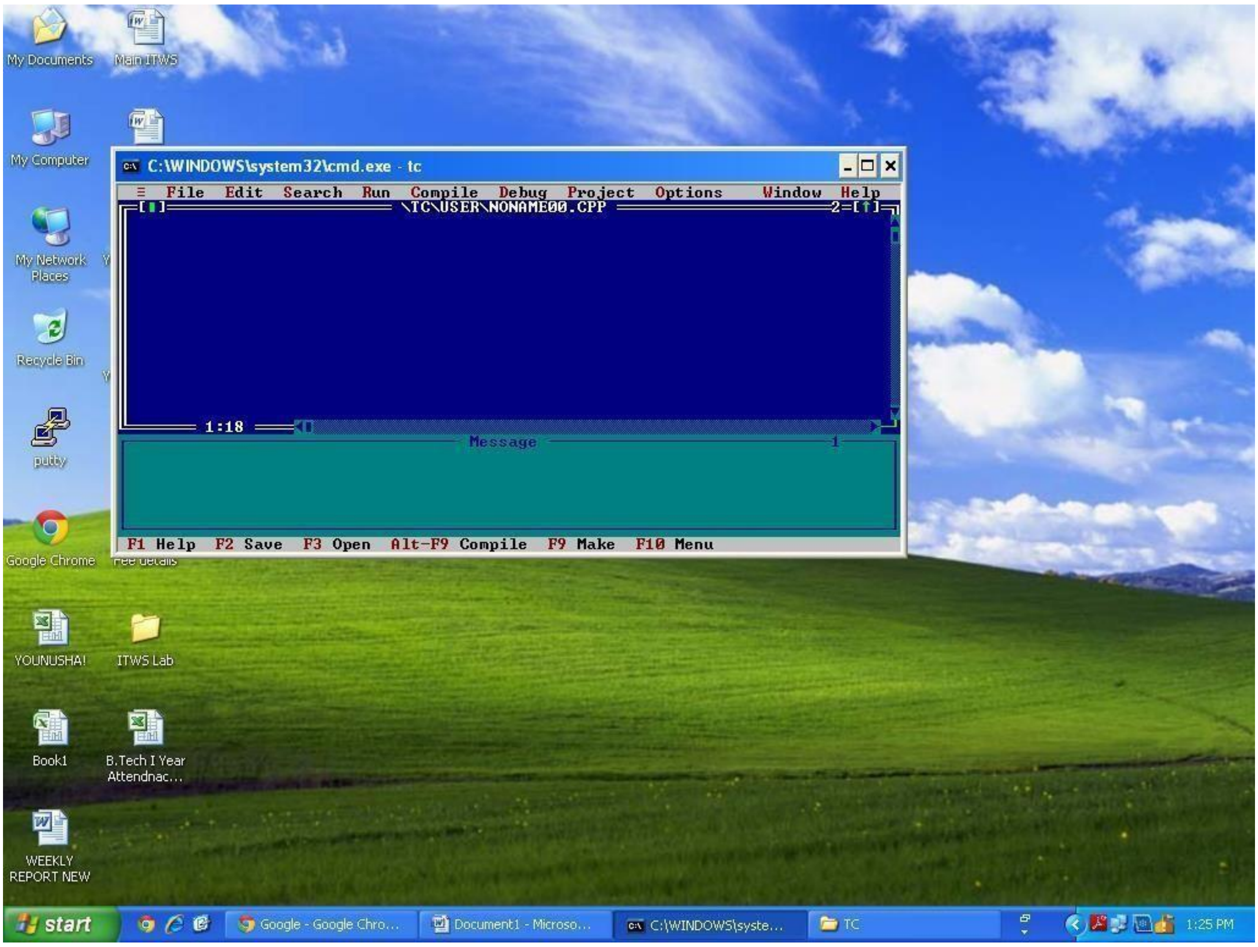


Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

Open: cmd

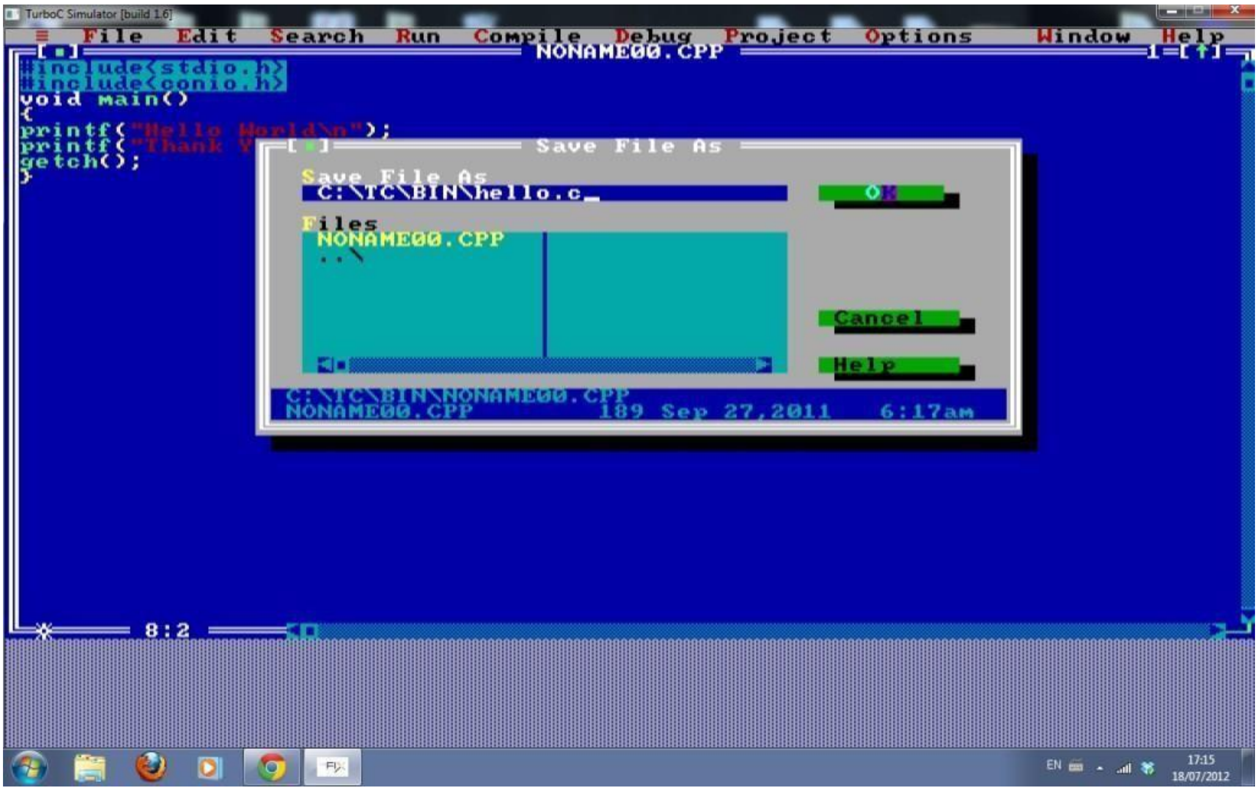


```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Admin>cd\

C:\>cd tc

C:\TC>cd bin

C:\TC\BIN>tc_
```



```
#include
void main()
{
printf("Hello World\n");
printf("Thank You");
getch();
}
```

I hope that's enough for a basic explanation of the program. If you still have doubts please ask through comments. Now let's **RUN this program using Turbo C**. Before going into the steps, you may SAVE your C program. Select **"File"** from menu -> click-> **Save**. Name the files as ->**hello.c** or some other name with a .c extension. See the screen shot below.



TurboC Simulator [build 1.6]
File Edit Search Run Compile Debug Project Options Window Help
NONAME00.CPP
Save File As
OK
Cancel
Help
8:2
EN
17:15
18/07/2012

```
#include<stdio.h>
#include<conio.h>
void main()
{
printf("Hello World\n");
printf("Thank Y
getch();
}
```

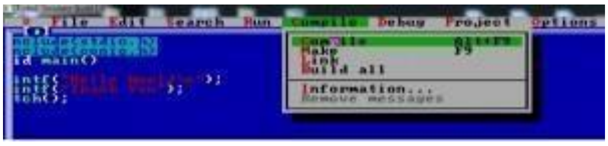
Save File As
C:\TC\BIN\hello.c

Files
NONAME00.CPP
..\

C:\TC\BIN\NONAME00.CPP
NONAME00.CPP 189 Sep 27,2011 6:17am

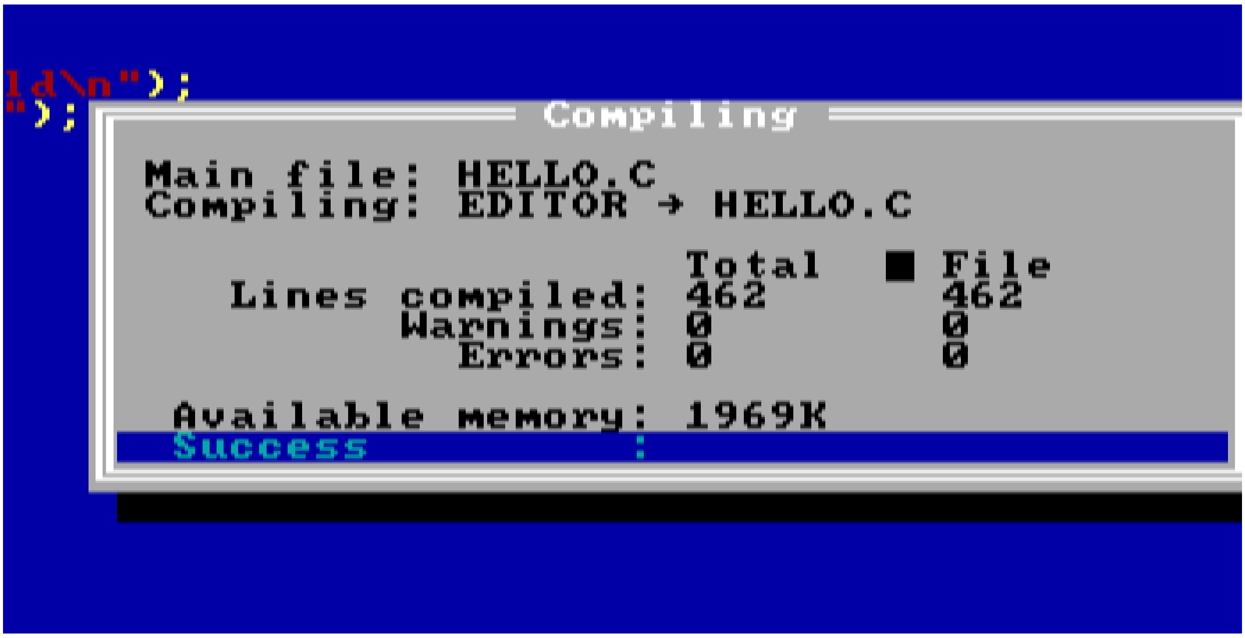
How to Compile a C program in Turbo C

The first step is compiling. Compiling makes sure your program is **free of syntax errors**. How ever compiler won't check for any logical/algorithmic errors. There is a lot of process that happens while the compiler compiles a program – which we will discuss later in coming articles. To do compiling **- Select -> Compile** from menu and **click-> compile**. See the image below.



After compiling, you will see a dialog box as shown below.If the compilation is success – you will see a**"success"** message. Else you will see the number of errors. Both are shown using screen shots.

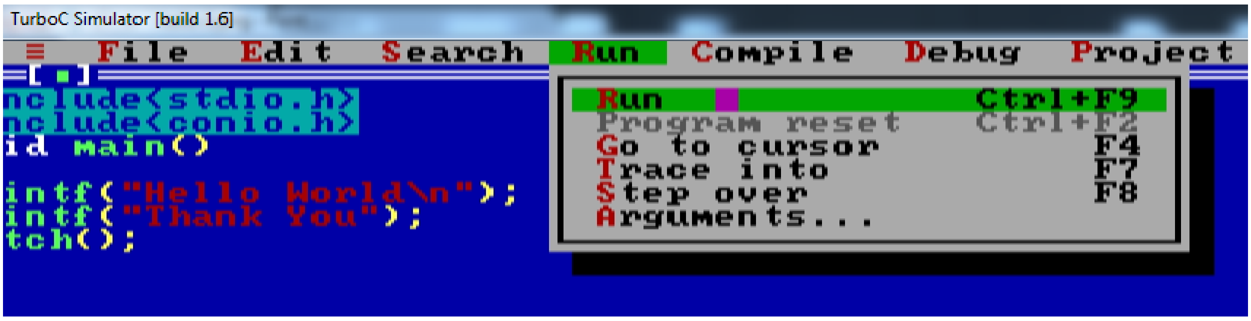
he screen shot of a “success” compilation



```
ld\n");
");
                    Compiling
  Main file:  HELLO.C
  Compiling:  EDITOR → HELLO.C

                          Total    File
     Lines compiled:      462      462
              Warnings:   0        0
                Errors:   0        0

   Available memory:  1969K
   Success          :
```

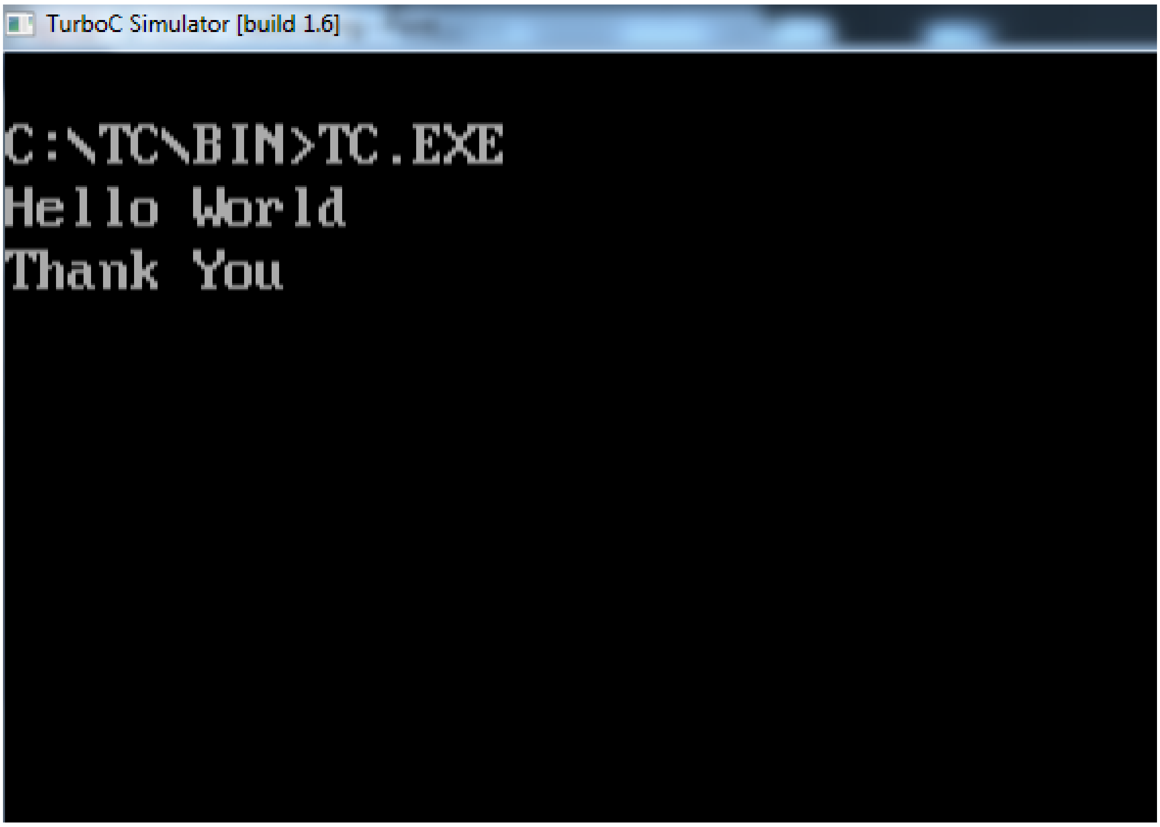
How to RUN a C Program in Turbo C compiler?

To RUN the program – you may **select ->Run** from menu and **click -> Run** (as shown in the image below).



```
nclude<stdio.h>
nclude<conio.h>
id main()

intf("Hello World\n");
intf("Thank You");
tch();
```

Now you will see the output screen as shown in the screen shot below.



```
C:\TC\BIN>TC.EXE
Hello World
Thank You
```

PATR-A

OPERATING SYSTEMS

1 Round Robin Scheduling

Preamble:

A Round Robin Scheduler algorithm is designed especially for time sharing systems. It is similar to FCFS scheduling but preemption is added to switch between processes Preemption: The act of interrupting a currently running task in order to give time to another task.

Aim:

Write a C program to implement the various process scheduling mechanisms such as Round Robin Scheduling.

Theory:

To implement Round robin scheduling we keep the ready queue as a FIFO queue of processes. New process is added to the tail of the ready queue. The cpu scheduler picks the first process from the ready queue, sets a **timer to interrupt** after one time quantum and dispatches the process. A small unit of time called a time quantum or time slice is defined. A time quantum is generally from 10 to 100 milli seconds.

The process may have CPU burst of less than one time quantum .In this case the process itself will release the CPU voluntarily .The scheduler will then proceed to the next process in the ready queue. Otherwise if the CPU burst of the currently running process is longer then 1 time quantum, the timer will go off and will cause an interrupt to the operating system. The average waiting time under Round Robin policy is how ever quite long.

Algorithm for RR

Step 1: Start the process

Step 2: Accept the number of processes in the ready Queue and time quantum (or) time slice

Step 3: For each process in the ready Q, assign the process id and accept the CPU burst time

Step 4: Calculate the no. of time slices for each process where

No. of time slice for process(n) = burst time process(n)/time slice

Step 5: If the burst time is less than the time slice then the no. of time slices =1.

Step 6: Consider the ready queue is a circular Q, calculate

(a) Waiting time for process(n) = waiting time of process(n-1)+ burst time of process(n-1) + the time difference in getting the CPU from process(n-1)

(b) Turn around time for process(n) = waiting time of process(n) + burst time of process(n)+ the time difference in getting CPU from process(n).

Step 7: Calculate

(a) Average waiting time = Total waiting Time / Number of process

(b) Average Turnaround time = Total Turnaround Time / Number of process

Step 8: Stop the process

/* Round Robin Scheduling Algorithm*/

Program:

```
#include<stdio.h>
#include<conio.h>
void main()
{
 int ts,pid[10],need[10],wt[10],tat[10],i,j,n,n1;
 int bt[10],flag[10],ttat=0,twt=0;
 float awt,atat;
 clrscr();

 printf("\t\t ROUND ROBIN SCHEDULING \n");
 printf("Enter the number of Processors \n");
 scanf("%d",&n);

 n1=n;
 printf("\n Enter the Timeslice \n");
 scanf("%d",&ts);
 for(i=1;i<=n;i++)
 {
   printf("\n Enter the process ID %d",i);
   scanf("%d",&pid[i]);
   printf("\n Enter the Burst Time for the process");
   scanf("%d",&bt[i]);
   need[i]=bt[i];
 }
 for(i=1;i<=n;i++)
 {
  flag[i]=1;
  wt[i]=0;
 }
 while(n!=0)
 {
   for(i=1;i<=n;i++)
   {
     if(need[i]>=ts)
     {
       for(j=1;j<=n;j++)
       {
               if((i!=j)&&(flag[i]==1)&&(need[j]!=0))
```

```
            wt[j]+=ts;
        }

        need[i]-=ts;
        if(need[i]==0)
        {
              flag[i]=0;
              n--;
        }
      }
      else
      {
        for(j=1;j<=n;j++)
        {
              if((i!=j)&&(flag[i]==1)&&(need[j]!=0))
              wt[j]+=need[i];
        }
        need[i]=0;
        n--;
        flag[i]=0;
    }
  }
}
for(i=1;i<=n1;i++)
{
  tat[i]=wt[i]+bt[i];
  twt=twt+wt[i];
  ttat=ttat+tat[i];
}
awt=(float)twt/n1;
atat=(float)ttat/n1;

printf("\n\n ROUND ROBIN SCHEDULING ALGORITHM \n\n");
printf("\n\n Process \t Process ID \t BurstTime \t Waiting Time \t TurnaroundTime \n ");
for(i=1;i<=n1;i++)
{
  printf("\n %5d \t %5d \t\t %5d \t\t %5d \t\t %5d \n", i,pid[i],bt[i],wt[i],tat[i]);
}

printf("\n The average Waiting Time=4.2f",awt);
printf("\n The average Turn around Time=4.2f",atat);
getch();
}
```

Output:

Round Robin Scheduling

```
Enter the number of Processors
4

 Enter the Timeslice
5
```

```
Enter the process ID 1 5

Enter the Burst Time for the process 10

Enter the process ID 2 6

Enter the Burst Time for the process 15

Enter the process ID 3 7

Enter the Burst Time for the process 20

Enter the process ID 4 8

Enter the Burst Time for the process 25
```

Round Robin Scheduling Algorithm:

```
Process      Process ID     BurstTime      Waiting Time    TurnaroundTime


   1             5              10             15              25

   2             6             15               25             40

   3             7             20              25              45

   4              8             25             20               45

The average Waiting  Time=4.2f
The average Turn around Time=4.2f
```

Applications:

1. Used in Resource sharing.

Viva Questions:

1. What is an interrupt?
2. Define FIFO?
3. Define Queue?
4. Define stack?
5. What is a process?

Sample Questions:

1. What is turnaround time?
2. Define race condition?
3. Define entry section and exit section?
4. What is preemptive and non preemptive scheduling?
5. Explain the process of Round Robin Algorithm?

2 SJF Scheduling

Preamble:

Shortest Process Next (SPN), is a scheduling policy that selects the waiting process with the smallest execution time to execute next. SJN is a non-preemptive algorithm.Shortest remaining time is a preemptive variant of SJN.

Aim:

Write a C program to implement the various process scheduling mechanisms such as SJF Scheduling .

Theory:

The SJF algorithm also can be implemented as FCFS, but on this the jobs are sorted by their burst time. A job with shortest burst time will be scheduled first i.e. sorting should be done from short to large time of burst time. Take arrival time also in consideration. This is SJF Non pre-emptive algorithm. While implementing Priority scheduling we have read the priority also and they should be sorted by highest priority to lowest priority.

It maintains the Ready queue in order of increasing job lengths. When a job comes in, insert it in the ready queue based on its length. When current process is done, pick the one at the head of the queue and run it.

Data Structures Required:

For all CPU scheduling algorithm implementations, we need to have the following arrays.

A two-dimensional array "process [20][1 0]" of data type float.

This array stores the job numbers, their arrival times, and their burst times, which are read from the user.

And this array also stores the additional data after some calculations. They are...

1. Start time of the job,
2. Finish time of the job,
3. Waiting time of job,
4. Turn around time of job.

A row of the process array contains the-following:

Process[1,0] = job number,

Process[1,1] = arrival time,

Process[1,2] = burst time, "

And after calculations...

Process[1,3] = start time,

Process[1,4] = finish time,

Process[1,5] = waiting time,

Process[1 ,6] = turn around time,

Remaining columns may be used while implementing the other algorithms.

Variables avgwt, avgtat, of float indicating average waiting time and average turn around time respectively, and variable TOT_JOBS for keeping track of number of jobs.

Algorithm for SJF

Step 1: Start the process

Step 2: Accept the number of processes in the ready Queue

Step 3: For each process in the ready Q, assign the process id and accept the CPU burst time

Step 4: Start the Ready Q according the shortest Burst time by sorting according to lowest to highest burst time.

Step 5: Set the waiting time of the first process as ‘0’ and its turnaround time as its burst time.

Step 6: For each process in the ready queue, calculate

(a) Waiting time for process(n)= waiting time of process (n-1) + Burst time of process(n-1)

(b) Turn around time for Process(n)= waiting time of Process(n)+ Burst time for process(n)

Step 6: Calculate

(c) Average waiting time = Total waiting Time / Number of process

(d) Average Turnaround time = Total Turnaround Time / Number of process

Step 7: Stop the process.

/* SJF Scheduling Algorithm */

Program:

```
#include<stdio.h>
void main()
{
 int i,j,k,n,sum,wt[10],tt[10],twt,ttat;
 int t[10],p[10];
 float awt,atat;
 clrscr();

 printf("Enter number of process\n");
 scanf("%d",&n);
```

```
for(i=0;i<n;i++)
{
   printf("\n Enter the Burst Time of Process %d",i);
   scanf("\n %d",&t[i]);
}

for(i=0;i<n;i++)
  p[i]=i;
  for(i=0;i<n;i++)
  {
    for(k=i+1;k<n;k++)
    {
      if(t[i]>t[k])
      {
              int temp;
              temp=t[i];
              t[i]=t[k];
              t[k]=temp;

              temp=p[i];
              p[i]=p[k];
              p[k]=temp;
      }
    }
    printf("\n\n SHORTEST JOB FIRST SCHEDULING ALGORITHM");
    printf("\n PROCESS ID \t BURST TIME \t WAITING TIME \t TURNAROUND TIME \n\n");
    wt[0]=0;
    for(i=0;i<n;i++)
    {
      sum=0;
      for(k=0;k<i;k++)
      {
            wt[i]=sum+t[k];
            sum=wt[i];
      }
    }
    for(i=0;i<n;i++)
    {
      tt[i]=t[i]+wt[i];
    }
    for(i=0;i<n;i++)
    {
      printf("%5d \t\T2%d \t\t %5d \t\t %5d \n\n",p[i],t[i],wt[i],tt[i]);
    }
    twt=0;
    ttat=t[0];
    for(i=1;i<n;i++)
    {
            twt=twt+wt[i];
            ttat=ttat+tt[i];
    }
    awt=(float)twt/n;
```

```
     atat=(float)ttat/n;

     printf("\n AVERAGE WAITING TIME %4.2f",awt);
     printf("\n AVERAGE TURN AROUND TIME %4.2f",atat);
     getch();
    }
}
```

Output:

```
Enter number of process
3

 Enter the Burst Time of Process 04

 Enter the Burst Time of Process 13

 Enter the Burst Time of Process 25
```

Shortest Job First Scheduling Algorithm:

```
 PROCESS ID      BURST TIME      WAITING TIME    TURNAROUND TIME

    1           3                0             3

    0           4                3             7

    2           5                7            12


 AVERAGE WAITING TIME 3.33
 AVERAGE TURN AROUND TIME 7.33
```

Applications:

1. Used in scheduling algorithm.
2. Used in Duration of time.
3. Best possible AWT (Average waiting Time)

Viva Questions:

1. Difference between SJF and FCFS?

2. What is waiting time?

3. What is brust time?

4 .How jobs are schedule in SJF?

5. The most optimal scheduling algorithm is?

Sample Questions:

Define CPU scheduling?

1. What is a Dispatcher?

2. What is turnaround time?
3. What is job scheduling?
4. The real difficulty with SJF in short term scheduling is?

3 FCFS Scheduling

Preamble:

First come, first served (FCFS) is an operating system process scheduling algorithm and a network routing management mechanism that automatically executes queued requests and processes by the order of their arrival. With first come, first served, what comes first is handled first; the next request in line will be executed once the one before it is complete.

Aim:

Write a C program to implement the various process scheduling mechanisms.

Theory:

The implementation of FCFS policy is easily managed with a FIFO queue. When a process enters the ready queue, its PCB is linked on to the tail of the queue. When the CPU is free it is allocated to the process at that end of the queue. The running process is the removed from the queue.

Data Structures Required:

For all CPU scheduling algorithm implementations, we need to have the following arrays.

A two-dimensional array "process [20][1 0]" of data type float.

This array stores the job numbers, their arrival times, and their burst times, which are read from the user.

And this array also stores the additional data after some calculations. They are...

1. Start time of the job,
2. Finish time of the job,
3. Waiting time of job,
4. Turn around time of job. .

A row of the process array contains the-following:

Process[1,0] = job number,

Process[1,1] = arrival time,

Process[1,2] = burst time, "

And after calculations...

Process[1,3] = start time,

Process[1,4] = finish time,

Process[1,5] = waiting time,

Process[1 ,6] = turn around time,

Remaining columns may be used while implementing the other algorithms.

Variables avgwt, avgtat, of float indicating average waiting time and average turn around time respectively, and variable TOT_JOBS for keeping track of number of jobs.

Functions Needed:

While implementing algorithms one can implement his/her own logic and can write own functions that is up to the programmers. As far as possible avoid using global variables. Though it is so, the suggested functions for implementing these algorithms are

1. for reading the job entries (read _job_entry())
2. FCFS() for fcfs algorithm
3. SJF_non_preempt ( ) for SJF algorithm,
4. Calculate () for doing various calculations like start times, finish times, waiting times, turn around times of jobs.
5. Sorting () for sorting the job entries by their arrival time (for FCFS), or by their burst time (for SJF), or by their priority (for Priority algo).
6. Print_sorted () for printing inputted data after sorting.
7. Printing (): printing the outputs after all the calculations are over.

Procedure for FCFS:

Read the number of jobs into variable tot _obs.

Then read the data for jobs as.....

Job number, into process [I,O],

Job arrival time into process [I,I],

Job burst into process [1,2], through read_job_entry().

Then sort the job entries by their arrival times because FCFS algorithm works by arrival times.

Print the sorted jobs on the output.

Then go for calculations.

Then Print the results on the output.

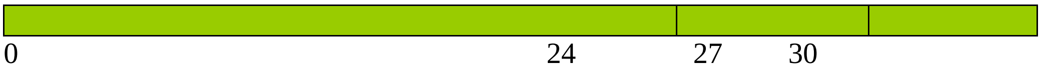
The read function can be called from the main function. Then FCFS function can be as

```
FC FS( ) {
sort(...):
printf("The Scheduling
according to FCFS:");
print_sorted ( . . .);
calculated(... );
getch( );
printing( };
getch( };
 }
```

Process	Burst Time
P1	24
P2	3
P3	3

Suppose that the processes arrive in the order: P1 , P2 , P3

The Gantt Chart for the schedule is:



Waiting time for P1 = 0; P2 = 24; P3 = 27

Average waiting time: (0 + 24 + 27)/3 = 17

Algorithm for FCFS scheduling:

Step 1: Start the process

Step 2: Accept the number of processes in the ready Queue

Step 3: For each process in the ready Q, assign the process id and accept the CPU burst time

Step 4: Set the waiting of the first process as ‘0’ and its burst time as its turn around time

Step 5: for each process in the Ready Q calculate

(c) Waiting time for process(n)= waiting time of process (n-1) + Burst time of process(n-1)

(d) Turn around time for Process(n)= waiting time of Process(n)+ Burst time for process(n)

Step 6: Calculate

(e) Average waiting time = Total waiting Time / Number of process

(f) Average Turnaround time = Total Turnaround Time / Number of process

Step 7: Stop the proces

/* FCFS scheduling Algorithm */

Program:

```
#include<stdio.h>
void main()
{
 int i,n,sum,wt,tat,twt,ttat;
 int t[10];
 float awt,atat;
 clrscr();

 printf("Enter number of processors:\n");
 scanf("%d",&n);
 for(i=0;i<n;i++)
 {
   printf("\n Enter the Burst Time of the process %d",i+1);
   scanf("\n %d",&t[i]);
 }
 printf("\n\n FIRST COME FIRST SERVE SCHEDULING ALGORITHM \n");
 printf("\n Process ID \t Waiting Time \t Turn Around Time \n");
 printf("1 \t\t 0 \t\t %d \n",t[0]);
 sum=0;
 twt=0;
 ttat=t[0];
 for(i=1;i<n;i++)
 {
  sum+=t[i-1];
  wt=sum;
  tat=sum+t[i];
  twt=twt+wt;
  ttat=ttat+tat;
  printf("\n %d \t\t %d \t\t %d",i+1,wt,tat);
  printf("\n\n");
 }
  awt=(float)twt/n;
  atat=(float)ttat/n;
  printf("\n Average Waiting Time %4.2f",awt);
  printf("\n Average Turnaround Time %4.2f",atat);
  getch();
}
```

Output:

```
 Enter number of processors: 3

 Enter the Burst Time of the process 1: 2

 Enter the Burst Time of the process 2:    5
```

```
Enter the Burst Time of the process 3:    4


First Come First Serve Scheduling Algorithm:

Process ID     Waiting Time    Turn Around Time
      1             0                 2

      2            2                 7


      3            7                11


Average Waiting Time 3.00
Average Turnaround Time 6.67
```

Applications:

1. Buying tickets in a movie.

Viva Questions:

1. What is FCFS?
2. What is SJF?
3. Explain Round Robin?
4. Explain SJF?
5. Expand FCFS?

Sample Questions:

1. Define non-preemptive scheduling?
2. What is mean by FCFS scheduling?
3. Define dispatch latency?
4. Define dispatcher and its functions?
5. What are file-server systems?

4 Priority Scheduling

Preamble:

Priority scheduling - Processes scheduling in which the **scheduler** selects tasks to run based on their priority

Aim:

Write a C program to implement the various process scheduling mechanisms such as Priority Scheduling.

Theory:

Run highest-priority processes first, use round-robin among processes of equal priority. Re-insert process in run queue behind all processes of greater or equal priority.

- Allows CPU to be given preferentially to important processes.
- Scheduler adjusts dispatcher priorities to achieve the desired overall priorities for the processes, e.g. one process gets 90% of the CPU.

Comments: In priority scheduling, processes are allocated to the CPU on the basis of an externally assigned priority. The key to the performance of priority scheduling is in choosing priorities for the processes.

Problem: Priority scheduling may cause low-priority processes to starve

Solution: (AGING) this starvation can be compensated for if the priorities are internally computed. Suppose one parameter in the priority assignment function is the amount of time the process has been waiting. The longer a process waits, the higher its priority becomes. This strategy tends to eliminate the starvation problem.

Waiting Time: It is the sum of the periods spent waiting in the ready queue.

Turnaround Time: The interval from the time of submission of a process to the time of completion is the turnaround time.

Response Time: It is the amount of time takes to start responding, but not the time that it takes to output that response.

Throughput: The number of processes that are completed per unit called throughput.

CPU Utilization: CPU Utilization may range from 0 to 100 percent

Algorithm for Priority Scheduling:

Step 1: Start the process

Step 2: Accept the number of processes in the ready Queue

Step 3: For each process in the ready Q, assign the process id and accept the CPU burst time

Step 4: Sort the ready queue according to the priority number.

Step 5: Set the waiting of the first process as '0' and its burst time as its turn around time

Step 6: For each process in the Ready Q calculate

(e) Waiting time for process(n)= waiting time of process (n-1) + Burst time of process(n-1)

(f) Turn around time for Process(n)= waiting time of Process(n)+ Burst time for process(n)

Step 7: Calculate

(g) Average waiting time = Total waiting Time / Number of process

(h) Average Turnaround time = Total Turnaround Time / Number of process

Step 8: Stop the process

/* Priority Scheduling */

Program:

```
#include   <stdio.h>
#include  <conio.h>
void main()
{
        int i,j,n,tat[10],wt[10],bt[10],pid[10],pr[10],t,twt=0,ttat=0;
        float awt,atat;
        clrscr();
        printf("\n-----------PRIORITY SCHEDULING -------------- \n");
        printf("Enter the No of Process: ");
        scanf("%d", &n);
        for (i=0;i<n;i++)
                {
                        pid[i] = i;
                        printf("Enter the Burst time of Pid %d : ",i);
                        scanf("%d",&bt[i]);
                        printf("Enter the Priority  of Pid %d : ",i);
                        scanf ("%d",&pr[i]);
                }
// Sorting start
                for (i=0;i<n;i++)
                        for(j=i+1;j<n;j++)
                        {
                                if (pr[i] > pr[j] )
                                {
                                        t = pr[i];
                                        pr[i] = pr[j];
                                        pr[j] = t;

                                        t = bt[i];
                                        bt[i] = bt[j];
                                        bt[j] = t;

                                        t = pid[i];
                                        pid[i] = pid[j];
                                        pid[j] = t;
                                }
                        }

// Sorting finished

        tat[0] = bt[0];
        wt[0] = 0;

        for (i=1;i<n;i++)
                {
                wt[i] = wt[i-1] + bt[i-1];
                tat[i] = wt[i] + bt[i];
                }
        printf("\n_____________________________________-______\n");
        printf("Pid\t Priority\tBurst time\t WaitingTime\tTurnArroundTime\n");
        printf("\n_____________________________________-_____\n");
                for(i=0;i<n;i++)
                {
                        printf("\n%d\t\t%d\t%d\t\t%d\t\t%d",pid[i],pr[i],bt[i],wt[i],tat[i]);
                }
        for(i=0;i<n;i++)
                {
                ttat = ttat+tat[i];
                twt = twt + wt[i];
                }
        awt = (float)twt / n;
        atat = (float)ttat / n;
        printf("\n\nAvg.Waiting Time: %f\nAvg.Turn Around Time: %f\n",awt,atat);
        getch();
}
```

Output:

```
----------- Priority Scheduling --------------

Enter the No of Process : 4
Enter the Burst time of PID 0: 2
Enter the Priority   of PID 0  :   3
Enter the Burst time of PID1:    6
Enter the Priority   of PID 1:    2
Enter the Burst time of PID2: 4
Enter the Priority   of PID 2:    1
Enter the Burst time of PID 3: 5
Enter the Priority   of PID 3:    7

-----------------------------------------------------------------------------------------------
PID     Priority      Burst time       WaitingTime    TurnArroundTime

-----------------------------------------------------------------------------------------------

2           1             4                  0                  4
1           2             6                  4                 10
0           3             2                 10                 12
3           7             5                 12                  17


Avg.Waiting Time: 6.500000
Avg.Turn around Time: 10.750000
```

Applications:

1. Used in Ticket Reservesation.
2. Used in DVD movie player vs send email.

Viva Questions:

1. What is mean by priority of a process?
2. What is starvation?
3. What is cause low-priority processes?
4. What is the strategy tends to eliminate the starvation problem?
5. What is turn around time?

Sample Questions:

1. What are privileged instructions?
2. What are the use of job queues, ready queues & device queues?
3. What is process control block?
4. How can a user program disrupt the normal operations of a system?
5. How is the protection for memory provided?
6. Define throughput?
7. Given the list of processes, their CPU burst times and arrival times, display/print the Gantt Chart for Priority Scheduling?

9.1.5 Sequential File Allocation

Preamble:

With contiguous allocation, each file has to occupy contiguous blocks on the disk. The location of a file is defined by the disk address of the first block and its length. Both sequential access and direct access are supported by the contiguous allocation.

Aim:

Write a C Program to implement Sequential File Allocation method.

Theory:

In this allocation strategy, each file occupies a set of contiguously blocks on the disk. This strategy is best suited. For of sequential files. The file allocation table consists of a single entry for each file. It shows the filenames, staring block of the file and size of the file. The main problem of this strategy is, it is difficult to find the contiguous free blocks in the disk and some free blocks could happen between two files.

Algorithm:

Step 1: Start the program.

Step 2: Get the number of memory partition and their sizes.

Step 3: Get the number of processes and values of block size for each process.

Step 4: First fit algorithm searches all the entire memory block until a hole which is big enough is encountered. It allocates that memory block for the requesting process.

Step 5: Best-fit algorithm searches the memory blocks for the smallest hole which can be allocated to requesting process and allocates if.

Step 6: Worst fit algorithm searches the memory blocks for the largest hole and allocates it to the process.

Step 7: Analyses all the three memory management techniques and display the best algorithm which utilizes the memory resources effectively and efficiently.

Step 8: Stop the program.

Program:

```
#include<stdio.h>
#include<conio.h>
main()
{
 int n,i,j,b[20],sb[20],t[20],x,c[20][20];
 clrscr();
 printf("Enter no.of files:");
 scanf("%d",&n);
 for(i=0;i<n;i++)
 {
        printf("Enter no. of blocks occupied by file%d",i+1);
        scanf("%d",&b[i]);
        printf("Enter the starting block of file%d",i+1);
        scanf("%d",&sb[i]);
        t[i]=sb[i];
        for(j=0;j<b[i];j++)
               c[i][j]=sb[i]++;
        }
 printf("Filename\tStart block\tlength\n");
 for(i=0;i<n;i++)
        printf("%d\t %d \t%d\n",i+1,t[i],b[i]);
 printf("Enter file name:");
 scanf("%d",&x);
 printf("File name is:%d",x);
 printf("length is:%d",b[x-1]);
 printf("blocks occupied:");
 for(i=0;i<b[x-1];i++)
        printf("%4d",c[x-1][i]);
 getch();
}
```

Output:

```
Enter no.of files: 2
Enter no. of blocks occupied by file1 4
Enter the starting block of file1 2
Enter no. of blocks occupied by file2 10
Enter the starting block of file2 5
 Filename        Start block     length
  1                   2              4
 2                    5              10

Enter file name: rajesh
File name is: 12803 lengths are: 0blocks occupied
```

Applications:

1. In Movies.

Viva Questions:

1. Different types of File Allocations?
2. FAT Standards for?
3. What are the advantages and disadvantages of Sequential File Allocation?

4. By using FAT, random access time is Increased or Decreased?
5. How the memory is divided in Sequential File Allocation?

Sample Questions:

1. Write Syntax for Sequential File Allocation?
2. Sequential File Allocation is best for which type of devices?
3. What is the average taken for Sequential File Allocation?
4. What is metadata?
5. A device driver can be throught of as a translator.Its input consiste of________commands and output consists of_________intrusctions.

6 Indexed File Allocation

Preamble:

This type of allocation will have a pointer which has the address of all the blocks of a file. This method solves the problem of fragmentation as the blocks can be stored in any location.

Aim:

Write a C Program to implement Indexed File Allocation method.

Theory:

Indexed allocation supports both sequential and direct access files.The file indexes are not physically stored as a part of the file allocation table. Whenever the file size increases, we can easily add some more blocks to the index. In this strategy, the file allocation table contains a single entry for each file. The entry consisting of one index block, the Index blocks having the pointers to the other blocks. No external fragmentation.

Algorithm:

```
Step 1:  Start.
Step 2: Let n be the size of the buffer
Step 3:  check if there are any producer
Step 4:  if yes check whether the buffer is full
Step 5: If no the producer item is stored in the buffer
Step 6:  If the buffer is full the producer has to wait
Step 7: Check there is any cosumer.If yes check whether the buffer is empty
Step 8: If no the consumer consumes them from the buffer
Step 9:  If the buffer is empty, the consumer has to wait.
Step 10: Repeat checking for the producer and consumer till required
Step 11: Terminate the process.
```

Program:

```
#include<stdio.h>
#include<conio.h>
main()
{
 int n,m[20],i,j,sb[20],s[20],b[20][20],x;
 clrscr();
 printf("Enter no. of files:");
 scanf("%d",&n);
 for(i=0;i<n;i++)
 {          printf("Enter starting block and size of file%d:",i+1);
            scanf("%d%d",&sb[i],&s[i]);
            printf("Enter blocks occupied by file%d:",i+1);
            scanf("%d",&m[i]);
            printf("enter blocks of file%d:",i+1);
            for(j=0;j<m[i];j++)
                        scanf("%d",&b[i][j]);
 } printf("\nFile\t index\tlength\n");
 for(i=0;i<n;i++)
 {
            printf("%d\t%d\t%d\n",i+1,sb[i],m[i]);
 }printf("\nEnter file name:");
 scanf("%d",&x);
 printf("file name is:%d\n",x);
 i=x-1;
 printf("Index is:%d",sb[i]);
 printf("Block occupied are:");
 for(j=0;j<m[i];j++)
            printf("%3d",b[i][j]);
 getch();
}
```

Output:

```
Enter no. of files:2
 Enter starting block and size of file1: 2 5
Enter blocks occupied by file1:10
enter blocks of file1:3
2 5  4  6  7  2  6  4 7
Enter starting block and size of file2: 3 4
Enter blocks occupied by file2:5
enter blocks of file2: 2 3 4 5 6
File     index length
 1       2       10
```

```
2       3       5
Enter file name: venkat
file name is:12803
Index is: 0Block occupied are:
```

Applications:

1. E-Books

Viva Questions:

1. Define File?

2. Define INDEX FILE ALLOCATION?

3. Explain the process of the index allocation procces?

4. Define the BLOCK?

5. What are the advantages and disadvantages of Index File Allocation?

Sample Questions:

1. Which memory block store files Name and Location?
2. Write the Syntax of Index File Allocation?
3. Is Index File Allocation is suitable for search engines?
4. What is the use of for loop in the program?
5. Indexed allocation__________direct access.(Supports or Does not Support)
6. The three major methods of allocating disk space that are in wide use are?

7 Linked File Allocation

Preamble:

In linked allocation, each file is a linked list of disk blocks. The directory contains a pointer to the first and (optionally the last) block of the file

Aim:

Write a C Program to implement Linked File Allocation method.

Theory:

It is easy to allocate the files, because allocation is on an individual block basis. Each block contains a pointer to the next free block in the chain. Here also the file allocation table consisting of a single entry Deptfor each file. Using this strategy any free block can be added to a chain very easily. There is a link between one block to another block, that’s why it is said to be linked allocation. We can avoid the external fragmentation.

Algorithm:

Step 1: Create a queue to hold all pages in memory

Step 2: When the page is required replace the page at the head of the queue

Step 3: Now the new page is inserted at the tail of the queue

Step 4: Create a stack

Step 5: When the page fault occurs replace page present at the bottom of the stack

Step 6: Stop the allocation.

Program:

```
#include<stdio.h>
#include<conio.h>
struct file
{
 char fname[10];
 int start,size,block[10];
}f[10];
main()
{
 int i,j,n;
 clrscr();
 printf("Enter no. of files:");
 scanf("%d",&n);
 for(i=0;i<n;i++)
 {
 printf("Enter file name:");
 scanf("%s",&f[i].fname);
 printf("Enter starting block:");
 scanf("%d",&f[i].start);
 f[i].block[0]=f[i].start;
 printf("Enter no.of blocks:");
 scanf("%d",&f[i].size);
 printf("Enter block numbers:");
 for(j=1;j<=f[i].size;j++)
 {
         scanf("%d",&f[i].block[j]);
 }
 }
 printf("File\tstart\tsize\tblock\n");
 for(i=0;i<n;i++)
 {
         printf("%s\t%d\t%d\t",f[i].fname,f[i].start,f[i].size);
         for(j=1;j<=f[i].size-1;j++)
                 printf("%d--->",f[i].block[j]);
         printf("%d",f[i].block[j]);
         printf("\n");
 }
 getch();
}
```

output:

```
Enter no. of files:2
Enter file name:venkat
Enter starting block:20
Enter no.of blocks:6
Enter block numbers: 4
12
15
45
32
25
Enter file name:rajesh
Enter starting block:12
Enter no.of blocks:5
Enter block numbers:6
5
4
3
2
 File    start   size    block
 venkat 20      6       4--->12--->15--->45--->32--->25
 rajesh  12      5      6--->5--->4--->3--->2
```

Applications:

1. In Blogs.

Viva questions:

1. There is no___________with linked allocation.
2. What will happen if a pointer is lost or damaged in a linked allocation?
3. What are the advantages and disadvantages of linked file allocation?
4. Contiguous allocation has two problems__________and__________that linked allocation solves.
5. Define Linked File Allocation?

Sample Questions:

1. The pointer overhead of indexed allocation is generally__________the pointer overhead of linked allocation.
2. Explain node**?**
3. Write the syntax for Linked file allocation**?**
4. Is the space wastage is less or more in Linked file allocation**?**
5. What will be the address in the last node**?**

8 Multiprogram Variable Task

Preamble:

IBM in their Mainframe Operating System OS/MFT implements the MFT concept. *OS/MFT* uses Fixed partitioning concept to load programs into Main memory

Aim:

Write a program to implement Dynamic allocation of memories in MVT.

Theory:

MVT stands for multiprogramming with variable number of tasks. Multiprogramming is a technique to execute number of programs simultaneously by a single processor. This is one of the memory management techniques. To eliminate the same of the problems with fixed partitions, an approach known as dynamic partitioning developed. In this technique, partitions are created dynamically, so that each process is loaded into partition of exactly the same size at that process. This scheme suffering from external fragmentation.

Fixed Partitioning:

- In fixed partitioning concept, RAM is divided into set of fixed partition of equal Size
- Programs having the Size Less than the partition size are loaded into Memory
- Programs Having Size more then the size of Partitions Size is rejected
- The program having the size less than the partition size will lead to internal Fragmentation.
- If all partitions are allocated and a new program is to be loaded, the program that lead to Maximum Internal Fragmentation can be replaced

Specification

Input Step-1

- Memory Size (memsize)
- Partition Size (parsize)

Process Step-1

- Compute number of partitions (n partitions)
- = memsize/parsize
- Create a Partition Table having n partition entries’

Partition Number	Partition Size	Process Number scheduled	Program Size	Internal Fragment Size
1	2	3	4	5

The column 3,4 and 5 initially will be blank

Input Step-2

- Number of Process (n processes) and their Sizes (process Size)

Process Step-3

- Create Process table having nprocesses

Process Number	Process Size Pratition Number Allocated	Partition Size	Internal Fragment Size	Allocated Flag
1	2	3	4	5

Column 3, 4, 5 and 6 initially will be blank

Process Step-2

- Allocate processes to Pratitions
- If the process Size is>Partition size Allocated Flag = o (over Size)
- If no partition exists Allocated flag =A and calculate the internal fragment size. Update the process row and Pratition row

Process Step-3

- Print the Process table and Partition table

Process step-4

- This step is done a LOOP till the user wants to exist
- Input 1 =Continue,2=Stop
- If 2 Break
- Read the Size of the Program>partition size then Reject

- If there is a unallocated partition, allocate the a partition and update the partition and process table
- If no partition is available, select the partition with maximum fragmentation and update the process table and partition table
- Print process table and partition Table.

Algorithm:

Step1: start the process.

Step2: Declare variables.

Step3: Enter total memory size.

Step4: Allocate memory for os.

Step5: allocate total memory to the pages.

Step6: Display the wastage of memory.

Step7: Stop the process.

/* MVT */

Program:

```
#include<stdio.h>
#include<conio.h>
main()
{
        int i,m,n,tot,s[20];
        clrscr();
        printf("Enter total memory size:");
        scanf("%d",&tot);
        printf("Enter no. of pages:");
        scanf("%d",&n);
        printf("Enter memory for OS:");
        scanf("%d",&m);
        for(i=0;i<n;i++)
        {
                printf("Enter size of page%d:",i+1);
                scanf("%d",&s[i]);
        }
        tot=tot-m;
        for(i=0;i<n;i++)
        {
                if(tot>=s[i])
                {
                        printf("Allocate page %d\n",i+1);
                        tot=tot-s[i];
                }
                else
                        printf("process p%d is blocked\n",i+1);
        }
        printf("External Fragmentation is=%d",tot);
        getch();
}
```

Output:

```
Enter total memory size: 50
Enter no.of pages : 4
Enter memory for OS : 10
Enter size of page: 10
Enter size of page: 9
Enter size of page: 9
Enter size of page: 10
External Fragmentation is = 2
```

Applications:

1. Used in Memory devices.

Viva Questions:

1. When memory is divided into several fixed sized partitions, each partition may contain_________.
2. MVT standards for?
3. The first fit, best fit and worst fit are strategies to select a_______.
4. In contiguous memory allocation is done by?
5. Define fragmentation?

Sample Questions:

1. What do you mean by first fit?
2. What do you mean by best fit?
3. What do you mean by worst fit?
4. What are the common strategies to select a free hole from a set of available holes?
5. What is the drawback of MVT?

9 Multiprogram Fixed Task

Preamble:

IBM in their Mainframe Operating ‘System OS/MFT implements the MFT concept. OSIMFT uses Dynamic Partition concept to load programs into Main memory.

Aim:

Write a program to implement Dynamic allocation of memories in MVT.

Theory:

MFT is the one of the memory management technique. In this technique, main memory is divided into no of static partitions at the system generated time. A process may be loaded into a partition of equal or greater size. The partition sizes are depending on o.s. in this memory management scheme the o.s occupies the low memory, and the rest of the main memory is available for user space

Dynamic Partitioning:

o Initially RAM is portioned according to the of programs to be loaded into Memory till such time no other program can be loaded.

o The Left over Memory is called a hole which is too small too fit any process.

o When a new program is to be into Memory Look for the partition, Which Leads to least External fragmentation and load the Program.

o The space that is not used in a partition is called as External Fragmentation

Specification

Input Step-1

- Memory Size (memsize)
- Number of Processes (n processes)

Input Step-2

- Create Process table having n processes entries

Process Number	Process Size	Partition Number Allocation	Partition Size	External Fragmentt Size	Allocated Flag
1	2	3	4	5	6

Column 3, 4, 5 and 6 initially will be blank

Process Step-2

- For each of the process do the following steps
- Input the process Size.
- Update the process Table with Process number and process size, partition number and size=0,Allocated Flag=N, fragment size=0
-

Process Step-3

- Create a partitions Table having One Row to start with
-

Partition Number	Partition Size	Process Number Scheduled	Program Size	External Fragment Size
1	2	3	4	5

Process Step-4

- Available Memory = Memory Size
- Partitionnum=0
- For each of the process the following steps are undertaken
- If Available memory> process Size
- If no row Exists
    - Create a new row in Partition table
    - Partitionnum = Partitionnum+1
    - Update a partition equel to the size of the program number, partition Size, and external fragment=0
    - Update the process table with partition number and size and Allocated Flag = A
    - Available Memory = Available Memory – Process Size

If a row exists

- Partitionnum=partitionnum+1
- Create a partition equal to the size of the program
- Update the Partition Table row with partition number, partition size, process number, process number, process size, and external fragment = 0
- Update the process table with partition number and size and
- Allocated Flag =A
- Available Memory = Available Memory – Process Size

Process Step-5

o Print the process and table and the hole size

Process step-6

o This step is done in a LOOP till user wants to exist

o Input 1 +continue, 2= stop

If 2 Break

Read the size of New Program

o Processnum = processnum +1

o If available memory>Program Size

o Partitionnum = partitionnum +1

o Create a partition equal to the size of the program

o Update the table row with partition number, partition size, process number, process size, and external fragment = 0

o Update the process table with partition number and size and Allocated Flag = A

o Available a memory = Available Memory – Process size

If available memory< Program Size

o Find the process row entry, which leads to least External Fragmentation

o External Fragmentation = Partition size – current program size

o Update the partition Table row related to process selected, with process number, process size, and external table with partition number and Allocated Flag=A

o Print process table and partition Table

Algorithm:

Step1: start the process.

Step2: Declare variables.

Step3: Enter total memory size.

Step4: Allocate memory for os.

Step5: allocate total memory to the pages.

Step6: Display the wastage of memory.

Step7: Stop the process.

Program:

```
#include<stdio.h>
#include<conio.h>
```

```
main()
{
        int ms,i,ps[20],n,size,p[20],s,intr=0;
        clrscr();
        printf("Enter size of memory:");
        scanf("%d",&ms);
        printf("Enter memory for OS:");
        scanf("%d",&s);
        ms-=s;
        printf("Enter no.of partitions to be divided:");
        scanf("%d",&n);
        size=ms/n;
        for(i=0;i<n;i++)
        {
                printf("Enter process and process size");
                scanf("%d%d",&p[i],&ps[i]);
                if(ps[i]<=size)
                {
                        intr=intr+size-ps[i];
                        printf("process%d is allocated\n",p[i]);
                }
                else
                        printf("process%d is blocked",p[i]);
        }
        printf("total fragmentation is %d",intr);
        getch();
}
```

Output:

```
Enter total memory size: 50
Enter memory for OS :10
Enter no.of partitions to be divided: 4
Enter size of page: 10
Enter size of page: 9
Enter size of page: 9
Enter size of page: 8

Internal Fragmentation is = 4
```

Applications:

1. Used in Memory devices.

Viva questions:

1. The disadvantage of moving all process to one end of memory and all holes to the other direction, producing one large hole of available memory is?
2. MFT standards for?

3. __________is generally faster than__________and__________.
4. External fragmentation exists when?
5. External fragmentation will not occur when?

Sample Questions:

1. When the memory allocated to a process is slightly larger than the process then?
2. A solution to the problem of external fragmentation is?
3. What is Contiguous memory allocation?
4. With relocation and limit registers, each logical address must be________the limit register.
5. How the relocation register helps?

10Simulate all File Organization Techniques:

a) Single level Directory b) Two level Directory c) Hierarchical d) DAG

Preamble:

The directory contains information about the files, including attributes, location and ownership. Sometimes the directories consisting of subdirectories also. The directory is itself a file, owned by the o.s and accessible by various file management routines

Aim: Write a program to implement

a) Single level Directory b) Two level Directory c) Hierarchical d) DAG

Theory:

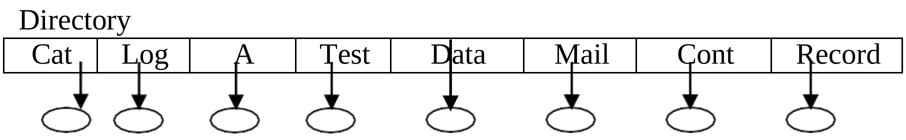
The file system of computer can be extensive some systems store thousands of files on hundreds of giga bytes of disk to manage all these data. We need to organize them. This organization is usually done in two ports. First the file system is broken into partition. Each disk on system contains at least one partition, which is a low level structure in which files and directories reside. Each partition is treated as a separate storage device; where as other systems allow partitions to be larger than a disk to group disks into one logical structure. In this way, the user needs to be concerned with only the logical directory and file structure.

Second, each partition contains information about files within it. This information is kept in entries in a device directory or volume of table contents. The device directory records information such as name location size and type for all files on that partition.

The most common schemes for defining the logical structure of a directory.

Single – level directory

The simplest directory structure is the single level directory. All files are contained in the same directory, which is easy to support and understand.

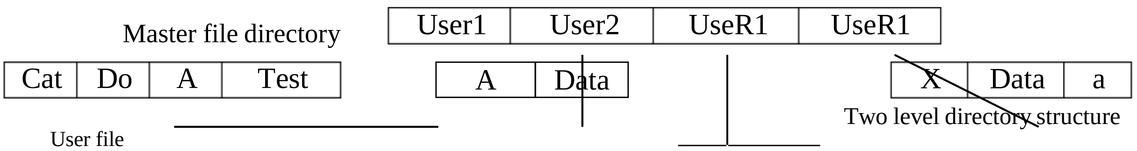


Single-level directory

Two – level directory

The major disadvantage to a single-level directory is the confusion of file names between different users. The standard solution is to create a separate directory for each user.

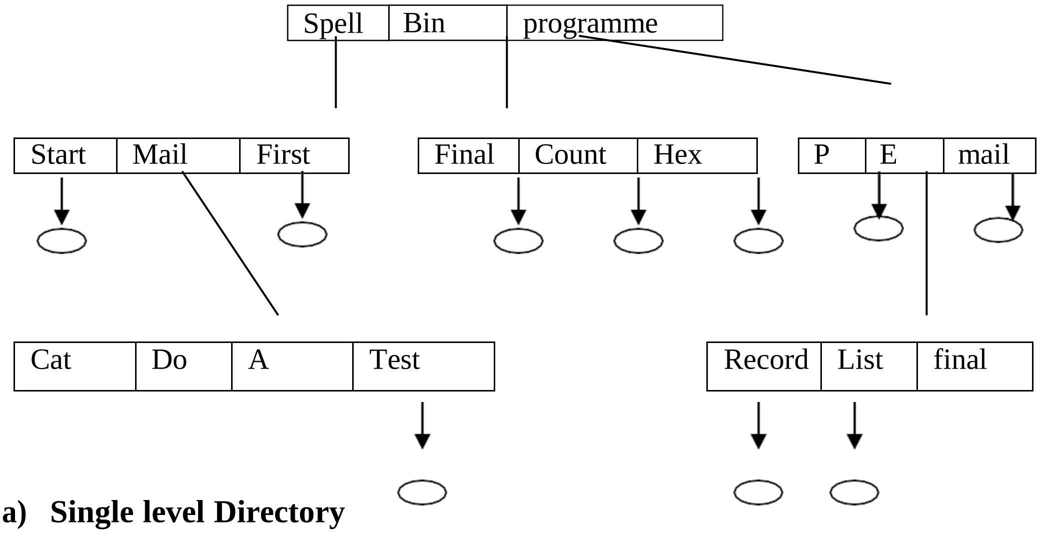
In the two-level directory structure, each user has his own user file directory (UFD). Each UFD has a similar structure but lists only the files of a single user.



Two level directory structure

Tree – Structured directories

A two – level directory is viewed as a two – level tree. The natural generalization is to extend the directory structure to a tree of arbitrary height. This generalization allows users to create their own sub directories and to organize their files accordingly a tree in the most common directory structure. The tree has root directory, every file on the system has a unique path name.



a) Single level Directory

Program:

```
 #include<stdio.h>
#include<conio.h>
#include<stdlib.h>
```

```
#include<graphics.h>
Dept
void main()
{

int gd=DETECT,gm,count,i,j,mid,cir_x;
char fname[10][20];

clrscr();
initgraph(&gd,&gm,"c:\tc\bgi");
cleardevice(); setbkcolor(GREEN);

puts("Enter no of files do u
have?"); scanf("%d",&count);
for(i=0;i<count;i++)
{
cleardevice(); setbkcolor(GREEN);
printf("Enter file %d name",i+1);
scanf("%s",fname[i]);
setfillstyle(1,MAGENTA);
mid=640/count;
cir_x=mid/3;
baR1d(270,100,370,150,0,0);
settextstyle(2,0,4);
settextjustify(1,1);
outtextxy(320,125,"Root
Directory"); setcolor(BLUE);
for(j=0;j<=i;j++,cir_x+=mid)
{
line(320,150,cir_x,250);
fillellipse(cir_x,250,30,30);
outtextxy(cir_x,250,fname[j]);
}
getch();
}
}
```

Output:

```
Enter no of files do u
 have? 3
 IT
Enter file 1 name: 1.c
Root 0 of 1.
 Enter File 2 name: 2.c
 Root
 1.
 2.
Enter File 3 name : 3.c
 Root
 1.
 2.
 3.
```

b) Two level Directory
Program:

```
 #include<stdio.h>
#include<graphics.h>
struct tree_element
{
char name[20];

int x,y,ftype,lx,rx,nc,level;
struct tree_element *link[5];
};
typedef struct tree_element
node; void main()
{

int gd=DETECT,gm;
node *root;
root=NULL;
clrscr();

create(&root,0,"null",0,639,320);
clrscr();
initgraph(&gd,&gm,"c:\tc\bgi");
display(root);

getch();
closegraph();
}
create(node **root,int lev,char *dname,int lx,int rx,int x)
{
```

```
int i,gap;
IT
if(*root==NULL)
{

(*root)=(node*)malloc(sizeof(node));
printf("enter name of dir/file(under
%s):",dname); fflush(stdin);

gets((*root)->name);
if(lev==0||lev==1)
of
(*root)->ftype=1; else
(*root)->ftype=2;
(*root)->level=lev;

(*root)->y=50+lev*50;
(*root)->x=x; (*root)-
>lx=lx; (*root)->rx=rx;
for(i=0;i<5;i++)
(*root)->link[i]=NULL;
JBIET
if((*root)->ftype==1)

{
if(lev==0||lev==1)
{

if((*root)->level==0)
printf("How many
users"); else

Dept
printf("hoe many files");
printf("(for%s):",(*root)->name);
scanf("%d",&(*root)->nc);
}

else (*root)-
>nc=0; if((*root)-
>nc==0) gap=rx-
lx;

else gap=(rx-lx)/(*root)-
>nc; for(i=0;i<(*root)-
>nc;i++)

create(&((*root)->link[i]),lev+1,(*root)-
>name,lx+gap*i,lx+gap*i+gap,lx+gap*i+gap/2);
}

else (*root)-
>nc=0;
}
}
display(node *root)
{
int i;
settextstyle(2,0,4);
settextjustify(1,1);
setfillstyle(1,BLUE);
setcolor(14);
if(root!=NULL)
{
for(i=0;i<root->nc;i++)
{
line(root->x,root->y,root->link[i]->x,root->link[i]->y);
}

if(root->ftype==1) baR1d(root->x-20,root->y-
10,root->x+20,root->y+10,0,0); else

fillellipse(root->x,root->y,20,20);
outtextxy(root->x,root->y,root-
>name); for(i=0;i<root->nc;i++)
{
IT
display(root->link[i]);
}
}
}
```

Output:

```
enter name of dir/file(under
null):sld How many users(forsld):2
enter name of dir/file(under
sld):tld hoe many files(fortld):2
enter name of dir/file(under tld):hir
enter name of dir/file(under tld):dag
enter name of dir/file(under sld):bin
hoe many files(forbin):2
enter name of dir/file(under bin):exe
enter name of dir/file(under bin):obj
```

c) Hierarchical

Program:

```
#include<stdio.h>
#include<graphics.h>
struct tree_element
```

```
{
Dept
char name[20];
int x,y,ftype,lx,rx,nc,level;
struct tree_element *link[5];
};
typedef struct tree_element
node; void main()
{
int gd=DETECT,gm;
node *root;
root=NULL;
clrscr();
create(&root,0,"root",0,639,320);
clrscr();
initgraph(&gd,&gm,"c:\tc\BGI");
display(root);
getch();
closegraph();
}
create(node **root,int lev,char *dname,int lx,int rx,int x)
{
int i,gap;
if(*root==NULL)
{
(*root)=(node *)malloc(sizeof(node));
printf("Enter name of dir/file(under %s) :
",dname); fflush(stdin);
gets((*root)->name);
printf("enter 1 for Dir/2 for
file :"); scanf("%d",&(*root)-
>ftype); (*root)->level=lev;
(*root)->y=50+lev*50;
(*root)->x=x;
(*root)->lx=lx;
(*root)->rx=rx;
for(i=0;i<5;i++)
(*root)->link[i]=NULL;
if((*root)->ftype==1)
{
printf("No of sub directories/files(for %s):",(*root)-
>name); scanf("%d",&(*root)->nc);

if((*root)->nc==0)
gap=rx-lx;
else gap=(rx-lx)/(*root)-
>nc; for(i=0;i<(*root)-
>nc;i++)
create(&((*root)->link[i]),lev+1,(*root)-
>name,lx+gap*i,lx+gap*i+gap,lx+gap*i+gap/2);
}
else (*root)-
>nc=0;
}
}
display(node *root)
{
int i;
settextstyle(2,0,4);
settextjustify(1,1);
setfillstyle(1,BLUE);
setcolor(14); if(root
!=NULL)
{
for(i=0;i<root->nc;i++)
{
line(root->x,root->y,root->link[i]->x,root->link[i]->y);
}
if(root->ftype==1) baR1d(root->x-20,root->y-10,root->x+20,root->y+10,0,0);
else
fillellipse(root->x,root->y,20,20);
outtextxy(root->x,root->y,root-
>name); for(i=0;i<root->nc;i++)
{
display(root->link[i]);
}
}
}
```

Output:

```
Enter Name of dir/file (under root):
ROOT Enter 1 for Dir / 2 For File : 1
No of subdirectories / files (for ROOT) :2
Enter Name of dir/file (under ROOT):
USER 1 Enter 1 for Dir /2 for file:1
No of subdirectories /files (for USER 1):
1 Enter Name of dir/file (under USER
1): SUBDIR Enter 1 for Dir /2 for file:1
No of subdirectories /files (for SUBDIR):
2 Enter Name of dir/file (under USER
1):JAVA Enter 1 for Dir /2 for file:1
No of subdirectories /files (for JAVA): 0
Enter Name of dir/file (under
SUBDIR): VB Enter 1 for Dir /2 for file:1
No of subdirectories /files (for VB): 0
Enter Name of dir/file (under
ROOT):USER2 Enter 1 for Dir /2 for file:1
No of subdirectories /files (for USER2):
2 Enter Name of dir/file (under
ROOT): A Enter 1 for Dir /2 for file:2
Enter Name of dir/file (under USER2): SUBDIR
2 Enter 1 for Dir /2 for file:1
```

```
No of subdirectories /files (for SUBDIR 2):
2 Enter Name of dir/file (under
SUBDIR2):PPL Enter 1 for Dir /2 for file:1
No of subdirectories /files (for PPL):
2 Enter Name of dir/file (under
PPL): B Enter 1 for Dir /2 for file:2
Enter Name of dir/file (under
PPL): C Enter 1 for Dir /2 for file:2
```

d) DAG

Program:

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<string.h>
struct  tree_element
IT
{
char name[20];

int x,y,ftype,lx,rx,nc,level;
struct tree_element *link[5];
};
typedef struct tree_element
node; typedef struct
{
of
char from[20];
char to[20];
}link;  link
L[10];   int
nofl;
node * root;
void main()
{

JBIET
int gd=DETECT,gm;
root=NULL;
clrscr();
create(&root,0,"root",0,639,320);
read_links();
clrscr();
initgraph(&gd,&gm,"c:\tc\BGI");
draw_link_lines();
Dept
display(root);
getch();
closegraph();
}
read_links()
{
int i;
printf("how many links");
scanf("%d",&nofl);
for(i=0;i<nofl;i++)
{
printf("File/dir:");
Computer Networks & Operating Systems Lab Manual
fflush(stdin);
gets(L[i].from);
printf("user name:");
fflush(stdin);
gets(L[i].to);
}
}
draw_link_lines()
{
int i,x1,y1,x2,y2;
for(i=0;i<nofl;i++)
{
search (root,L[i].from,&x1,&y1);
search(root,L[i].to,&x2,&y2);
setcolor(LIGHTGREEN);
setlinestyle(3,0,1);
line(x1,y1,x2,y2);
setcolor(YELLOW);
setlinestyle(0,0,1);
}
}
search(node *root,char *s,int *x,int *y)
{
int i;
if(root!=NULL)
{
if(strcmpi(root->name,s)==0)
IT
{
*x=root->x;
*y=root->y;
return;
}
else
{
of
for(i=0;i<root->nc;i++)
search(root->link[i],s,x,y);
}
}
}
create(node **root,int lev,char *dname,int lx,int rx,int x)
```

```
{
int i,gap;
if(*root==NULL)
JBIET
{
(*root)=(node *)malloc(sizeof(node));
printf("enter name of dir/file(under
%s):",dname); fflush(stdin);
gets((*root)->name);
printf("enter 1 for dir/ 2 for
file:"); scanf("%d",&(*root)-
>ftype); (*root)->level=lev;
Dept
(*root)->y=50+lev*50;
(*root)->x=x;
(*root)->lx=lx;
(*root)->rx=rx;
for(i=0;i<5;i++)
(*root)->link[i]=NULL;
if((*root)->ftype==1)
{
printf("no of sub directories /files (for %s):",(*root)-
>name); scanf("%d",&(*root)->nc);
if((*root)->nc==0)
gap=rx-lx;
Computer Networks & Operating Systems Lab Manual
else
gap=(rx-lx)/(*root)->nc;
for(i=0;i<(*root)->nc;i++)
create( & ( (*root)->link[i] ) , lev+1 ,
(*root)->name,lx+gap*i,lx+gap*i+gap,lx+gap*i+gap/2);
}
else (*root)->nc=0;
}
}
/* displays the constructed tree in graphics
mode */ display(node *root)
{
int i;
settextstyle(2,0,4);
settextjustify(1,1);
setfillstyle(1,BLUE);
setcolor(14); if(root
!=NULL)
{
for(i=0;i<root->nc;i++)
{
line(root->x,root->y,root->link[i]->x,root->link[i]->y);
}
if(root->ftype==1) baR1d(root->x-20,root->y-10,root-
>x+20,root->y+10,0,0);
IT
else
fillellipse(root->x,root->y,20,20);
outtextxy(root->x,root->y,root-
>name); for(i=0;i<root->nc;i++)
{
display(root->link[i]);
}}}
```

of
JBIET
Dept
Computer Networks & Operating Systems Lab Manual

Output:

```
Enter Name of dir/file (under root): ROOT
Enter 1 for Dir / 2 For File : 1
No of subdirectories / files (for ROOT) :2
Enter Name of dir/file (under ROOT): USER 1
Enter 1 for Dir /2 for file:1
No of subdirectories /files (for USER 1): 2
Enter Name of dir/file (under USER1): VB
Enter 1 for Dir /2 for file:1
No of subdirectories /files (for VB): 2
Enter Name of dir/file (under VB): A
Enter 1 for Dir /2 for file:2
Enter Name of dir/file (under VB): B
Enter 1 for Dir /2 for file:2
Enter Name of dir/file (under USER1): C
Enter 1 for Dir /2 for file:2
Enter Name of dir/file (under ROOT): USER2
Enter 1 for Dir /2 for file:1
No of subdirectories /files (for USER2): 1
Enter Name of dir/file (under USER2):JAVA
Enter 1 for Dir /2 for file:1
No of subdirectories /files (for JAVA):2
Enter Name of dir/file (under JAVA):D
Enter 1 for Dir /2 for file:2
Enter Name of dir/file (under JAVA):HTML
IT
Enter 1 for Dir /2 for file:1
No of subdirectories /files (for HTML):0
How many links:2
File/Dir: B
User Name: USER 2
File/Dir: HTML
User Name: USER1
```

of
ASTRA
Dept

Applications:

1. Used in indexing system

Viva Questions:

1. What are the modes of a file?
2. What are the file permissions?
3. Disadvantage of 1level directory?
4. Access methods of files?
5. What is mean by FMS?

Sample Questions:

1. The directory can be viewed as a___________that translates file names into their directory entries.
2. What are the two level directory structures?
3. How many types of path names?
4. When two users keep a subdirectory in their own directories, the structure being referred to?
5. The operating system________the links when traversing directory trees, to preserve the acyclic structure of the system.

11 Simulate Banker’s Algorithm for Deadlock Avoidance

Preamble:

Deadlock can be avoided if certain information about processes are available to the operating system before allocation of resources, such as which resources a process will consume in its lifetime. For every resource request, the system sees whether granting the request will mean that the system will enter an *unsafe* state, meaning a state that could result in deadlock. The system then only grants requests that will lead to *safe* states.[1] In order for the system to be able to determine whether the next state will be safe or unsafe

Aim:

To implement deadlock avoidance by using banker’s algorithm.

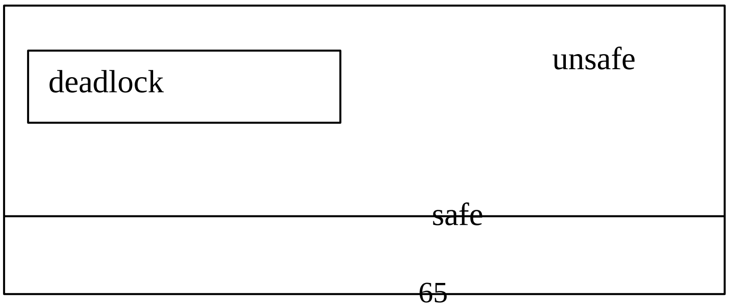
Theory:

Methods for Handling Deadlock

- Never let deadlock occur
- Prevention: break one of the 4 condition
- Avoidance: resources give advance notice of maximum use
- Let deadlock occur and do something about it
- Detection: search for cycles periodically
- Recovery: preempt process or resources
- Don’t worry about it(UNIX and other OS)
- Cheap: just reboot (it happens rarely)

Deadlock: Avoidance

- Process give advance notice maximum usage of resources
- Process make actual request when they need a resource
- Avoidance algorithm: allocate request only if it yields a safe state
- Conceptually the process could be run in this order



Banker’s Algorithm

- Multiple instances of resource types IMPLIES cannot use resource allocation graph
- Banks do not allocate cash unless they can satisfy customer needs when a new process enters the system
- Declare in advance maximum need for each resource type
- Cannot exceed the total resources of the type
- Later, process make actual request for some resources
- Otherwise, suspend process until other process release enough resources

Banker: Example

Initially:

Available

A	B	C
10	5	7

Later snapshot:

	Max	-	*Allocation*	=	*Need*	Available
	A B C		**A B C**		**A B C**	**A B C**
P0	7 5 3		0 1 0		7 4 3	3 3 2
P1	3 2 2		2 0 0		1 2 2	
P2	9 0 2		3 0 2		6 0 0	
P3	2 2 2		2 1 1		0 1 1	
P4	4 3 3		0 0 2		4 3 1	

Safety Algorithm:

```
STEP 1: initialize
            Work: =Available;
             For I = 1,2............n
                   Finish[i] = false
STEP 2: find I such that both
           a.   finish[i] is false
           b.   need I <=work
           if no such I, goto STEP 4
 STEP 3:
            Work: = work+ allocation i
             Finish [i] =true
             Goto STEP 2
STEP 4:
```

If finish[i] = true for all I, system is in safe state

Banker: Safety Example

Using the previous Example, P1, P2, P3, P4,P0 satisfies criteria

	Max -	*Allocation* =	*need* <=	*work* Available
	A B C	**A B C**	**A B C**	**A B C**
P1	3 2 2	2 0 0	1 2 2	3 3 2
P2	3 3 2			
P3	2 2 2	2 1 1	0 1 1	5 3 2
P4	4 3 3	0 0 2	4 3 1	7 4 3
P2	9 0 2	3 0 2	6 0 0	7 4 5
P0	7 5 3	0 1 0	7 4 3	10 4 7
				10 5

7<<<initial system

Resources request Algorithm:

Say P1 request (1, 0, 2)
Compare to Need 1: (1,0, 2) <=(1, 2, 2)
Compare to available: (1, 0,2) <= (3, 3, 2)
Pretend to allocate resources:

	Max -	*Allocation* =	*Need*	Available
	A B C	**A B C**	**A B C**	
P0	7 5 3	010	7 4 3	2 3 0<<<
P1	3 2 2	3 0 2<<<	0 2 0<<<	
P2	9 0 2	3 0 2	6 0 0	
P3	2 2 2	2 1 1	0 1 1	
P4	4 3 3	0 0 2	4 3 1	

Is this safe? Yes: P1, P3, P4, P0, P2
Can P4 get (3, 3, 0)? No,(3,3,0) > (2, 3, 0) Available
Can P0 get (0, 2, 0)? (0,2, 0) < (2,3, 0) Available
Pretend: Available goes to (2, 1. 0)
But All needs greater than Available than Available IMPLIES NOT SAFE

Algorithm:

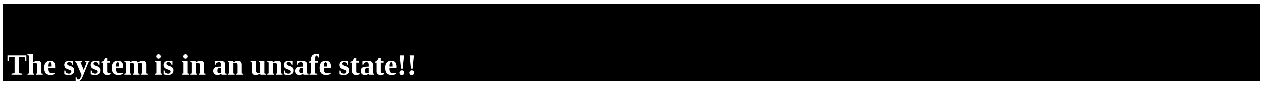
1. Start the program.
2. Get the values of resources and processes.
3. Get the avail value.
4. After allocation find the need value.
5. Check whether its possible to allocate.
6. If it is possible then the system is in safe state.

7. Else system is not in safety state.
8. If the new request comes then check that the system is in safety.
9. Or not if we allow the request.
10. Stop the program.

Program:

```
#include< stdio.h >
#include< conio.h >
void main()
{
int clm[7][5],req[7][5],alloc[7][5],rsrc[5],avail[5],comp[7];
int first,p,r,i,j,prc,count,t;
clrscr();
count=0;
for(i=1;i<=7;i++)
comp[i]=0;
printf(“Enter the no of processes:\n”);
scanf(“%d”,&p);
printf(“Enter the no of resources:\n”);
scanf(“%d”,&r);
printf(“Enter the claim for each process:”);
for(i=1;i<=p;i++)
{
printf(“\nFor process %d”,i);
for(j=1;j<=r;j++)
{
scanf(“%d”,&clm[i][j]);
}
}
printf(“Enter the allocation for each process:\n”);
for(i=1;i<=p;i++)
{
printf(“\nFor process “,i);
for(j=1;j<=r;j++)
{
scanf(“%d”,&alloc[i][j]);
}
}
printf(“Enter total no of each resource:”);
for(j=1;j<=r;j++)
scanf(“%d”,&rsrc[j]);
for(j=1;j<=r;j++)
{
int total=0;
avail[j]=0;
```

```
for(i=1;i<=p;i++)
{total+=alloc[i][j];}
avail[j]=rsrc[j]-total;
}
do
{
for(i=1;i<=p;i++)
{
 for(j=1;j<=r;j++)
{
 req[i][j]=clm[i][j]-alloc[i][j];
}
}
printf(“\n\nAvailable resorces is:”);
for(j=1;j<=r;j++)
{ printf(” “,avail[j]); }
printf(“\nClaim matrix:\t\tAllocation matrix:\n”);
for(i=1;i<=p;i++)
{
for(j=1;j<=r;j++)
{
printf(“%d”,clm[i][j]);
}
printf(“\t\t\t”);
for(j=1;j<=r;j++)
{
 printf(“%d”,alloc[i][j]);
  }
printf(“\n”);
}
prc=0;
for(i=1;i<=p;i++)
{
if(comp[i]==0)//if not completed
{
prc=i;
for(j=1;j<=r;j++)
{
if(avail[j]
{
prc=0;
break;
}
}
}
if(prc!=0)
break;
}
if(prc!=0)
{
printf("\nProcess ",prc,"runs to completion!");
count++;
for(j=1;j<=r;j++)
{
avail[j]+=alloc[prc][j];
alloc[prc][j]=0;
clm[prc][j]=0;
comp[prc]=1;
}
}
}
while(count!=p&&prc!=0);
if(count==p)
printf("\nThe system is in a safe state!!");
else
printf("\nThe system is in an unsafe state!!");
getch();
}
——————————————————————————-
OUT PUT:
——————————————————————————-
Enter the no of processes:
2
Enter the no of resources:
3

Enter the claim for each process:

For process I : 2  4  5

For process II: 2  5  3

Enter the total no of each resource : 5 5 2

Available resource is :

Claim matrix:           allocation matrix:
245                         123
253                         234
```



```
The system is in an unsafe state!!
```

Applications:

1. Used in traffic.

Viva Questions:

1. Explain Banker Algorithm?
2. How we can detect?
3. How we can rectify?
4. What are the methods used to detect deadlock?
5. What is mutual exclusion?

Sample Questions:

1. The Banker's algorithm is_______________than the resource allocation graph algorithm.
2. A deadlock avoidance algorithm dynamically examines the____________, to ensure that a circular wait condition can never exist.
3. When does a system is in a safe state only if there exists a?
4. The data structures available in the Banker's algorithm are?
5. If no cycle exists in the resource allocation graph then?

12 Simulate Bankers Algorithm for Deadlock prevention

Preamble:

Deadlock prevention algorithms are used in concurrent programming when multiple processes must acquire more than one shared resource. If two or more concurrent processes obtain multiple resources indiscriminately, a situation can occur where each process has a resource needed by another process. As a result, none of the processes can obtain all the resources it needs, so all processes are blocked from further execution. This situation is called a deadlock. A deadlock prevention algorithm organizes resource usage by each process to ensure that at least one process is always able to get all the resources it needs.

Aim:

To implement deadlock prevention by using Banker's Algorithm.

Procedure:

- Break mutual exclusion:
- Read-only files are shareable
- But some resources
- Are intrinsically non-shareable (printers)
- Break hold and wait:
- Request all resources in advance
- Request (tape, disk, printer)
- Release all resources before requesting New batch
- Request (tape, disk), release (tape, disk), request (disk, printer)
- Disadvantages: low resources utilization, starvation

Deadlock: Prevention

- Break no preemption
- Process 1 request resources already allocated to process
- process I forfeits its current resources
- if process 2 is waiting for other resources: process 2 forfeits
- used for resources whose state is easily saved / restored
- CPU registers and memory space
- Break circular wait
- Process request resources in increasing
- Order

Expected Output:

```
enter the no of processes::4
enter the max types of resources::3
enter the claim matrix and allocation matrix::
 P1 P2 P3 P4              P1 P2 P3 P4
R1 3  6  3  4          R1 1 6 2  0
R2 2  1  1  2           R2  0  1  1 0
R1 2  3  4  2           R1  0  2  1 2

enter the resource vector::
9
3
6
enter the available vector::
0
1
1
the process p1 claims resources within system limit
the allocated resources are within claim limit
the process p2 claims resources within system limit
the allocated resources are within claim limit
the process p3 claims resources within system limit
the allocated resources are within claim limit
the process p4 claims resources within system limit
the allocated resources are within claim limit
No error in count of resources r1
No error in count of resources r2
No error in count of resources R1
```

Banker's Algorithm:

- Multiple instances of resource types IMPLIES cannot use resource allocation graph
- Banks do not allocate cash unless they can satisfy customer needs when a new process enters the system
- Declare in advance maximum need for each resource type
- Cannot exceed the total resources of the type
- Later, process make actual request for some resources
- Otherwise, suspend process until other process release enough resources

Banker: Example

Initially:
Available

A	B	C
10	5	7

Later snapshot:

	Max	-	*Allocation*	=	*Need*	Available
	A B C		A B C		A B C	A B C
P0	7 5 3		0 1 0		7 4 3	3 3 2
P1	3 2 2		2 0 0		1 2 2	
P2	9 0 2		3 0 2		6 0 0	

P3	2 2 2	2 1 1	0 1 1
P4	4 3 3	0 0 2	4 3 1

Safety Algorithm:

```
 STEP  1:initialize
        Work :=Available;
         For I = 1,2............n
                Finish[i] = false
STEP 2: find I such that both
     c.   finish[i] is false
     d.   need I <=work
     if no such I, goto STEP 4
STEP 3:
        Work := work+ allocation i
         Finish [i] =true
         Goto STEP 2
 STEP 4:
  If finish[i] = true for all I, system is in safe state
```

Program:

```
#include<  stdio.h>
#include< conio.h>
void main()
{
 int allocated[15][15],max[15][15],need[15][15],avail[15],tres[15],work[15],flag[15];
 int pno,rno,i,j,prc,count,t,total;
 count=0;
 clrscr();

 printf("\n Enter number of process:");
 scanf("%d",&pno);
 printf("\n Enter number of resources:");
 scanf("%d",&rno);
 for(i=1;i< =pno;i++)
 {
  flag[i]=0;
 }
 printf("\n Enter total numbers of each resources:");
 for(i=1;i<= rno;i++)
  scanf("%d",&tres[i]);
 printf("\n Enter Max resources for each process:");
 for(i=1;i<= pno;i++)
 {
  printf("\n for process %d:",i);
  for(j=1;j<= rno;j++)
  scanf("%d",&max[i][j]);
 }
 printf("\n Enter allocated resources for each process:");
 for(i=1;i<= pno;i++)
 {
  printf("\n for process %d:",i);
  for(j=1;j<= rno;j++)
  scanf("%d",&allocated[i][j]);
}
 printf("\n available resources:\n");
 for(j=1;j<= rno;j++)
 {
  avail[j]=0;
  total=0;
  for(i=1;i<= pno;i++)
  {
   total+=allocated[i][j];
  }
  avail[j]=tres[j]-total;
  work[j]=avail[j];
  printf("     %d \t",work[j]);
 }
do
 {
 for(i=1;i<= pno;i++)
 {
  for(j=1;j<= rno;j++)
  {
   need[i][j]=max[i][j]-allocated[i][j];
  }
 }
 printf("\n Allocated matrix         Max       need");
 for(i=1;i<= pno;i++)
 {
  printf("\n");
  for(j=1;j<= rno;j++)
  {
   printf("%4d",allocated[i][j]);
  }
  printf("|");
  for(j=1;j<= rno;j++)
  {
   printf("%4d",max[i][j]);
  }
   printf("|");
  for(j=1;j<= rno;j++)
  {
   printf("%4d",need[i][j]);
  }
  }
 prc=0;
  for(i=1;i<= pno;i++)
  {
   if(flag[i]==0)
   {
    prc=i;
    for(j=1;j<= rno;j++)
    {
     if(work[j]< need[i][j])
     {
       prc=0;
       break;
     }
    }
   }
    if(prc!=0)
   break;
  }
   if(prc!=0)
  {
   printf("\n Process %d completed",i);
   count++;
   printf("\n Available matrix:");
   for(j=1;j<= rno;j++)
   {
    work[j]+=allocated[prc][j];
    allocated[prc][j]=0;
    max[prc][j]=0;
    flag[prc]=1;
    printf("  %d",work[j]);
   }
  }
  }
while(count!=pno&&prc!=0);
 if(count==pno)
  printf("\nThe system is in a safe state!!");
 else
  printf("\nThe system is in an unsafe state!!");
  getch();
}
```

Output:

```
Enter number of process:5

 Enter number of resources:3

 Enter total numbers of each resources:10 5 7

 Enter Max resources for each process:
 for process 1:7 5 3

 for process 2:3 2 2

 for process 3:9 0 2

 for process 4:2 2 2

for process 5:4 3 3

Enter allocated resources for each process:
for process 1:0 1 0

for process 2:3 0 2


for process 3:3 0 2

for process 4:2 1 1

for process 5:0 0 2

available resources:
   2      3      0


Allocated matrix        Max       need
  0 1 0|  7 5  3|  7 4 3
  3   0   2|   3   2   2|   0   2   0
  3   0   2|   9   0   2|   6   0   0
  2   1   1|   2   2   2|   0   1   1
  0   0   2|   4   3   3|   4   3   1
Process 2 completed
Available matrix:  5  3   2
Allocated matrix        Max       need
0 1 0|  7 5  3|  7 4 3
  0   0   0|   0   0   0|   0   0   0
  3   0   2|   9   0   2|   6   0   0
  2   1   1|   2   2   2|   0   1   1
  0   0   2|   4   3   3|   4   3   1
Process 4 completed
Available matrix:  7  4   3
Allocated matrix        Max       need
0 1 0|  7 5  3|  7 4 3
  0   0   0|   0   0   0|   0   0   0
  3   0   2|   9   0   2|   6   0   0
  0   0   0|   0   0   0|   0   0   0
  0   0   2|   4   3   3|   4   3   1
Process 1 completed
Available matrix:  7  5   3
Allocated matrix        Max       need
0 0 0|  0 0  0|  0 0 0
  0   0   0|   0   0   0|   0   0   0
  3   0   2|   9   0   2|   6   0   0
  0   0   0|   0   0   0|   0   0   0
  0   0   2|   4   3   3|   4   3   1
Process 3 completed
Available matrix: 10 5   5
Allocated matrix        Max       need
```

```
  0  0  0|  0  0  0|  0  0  0
  0  0  0|  0  0  0|  0  0  0
  0  0  0|  0  0  0|  0  0  0
  0  0  0|  0  0  0|  0  0  0
  0  0  2|  4  3  3|  4  3  1
 Process 5 completed
 Available matrix: 10 5  7
The system is in a safe state!!
```

Applications:

1. Used in traffic.

Viva Questions:

1. Deadlock prevention is a set of methods?

2. For a Hold and wait condition to prevail?

3. for non sharable resources like a printer, mutual exclusion?

4. To ensure that the hold and wait condition never occurs in the system, it must be ensured that?

5. one way to ensure that the circular wait condition never holds is to?

Sample Questions:

1. For a deadlock to arise, which of the following conditions must hold simultaneously?

2. The number of resources requested by a process is?

3. To ensure no preemption, if a process is holding some resources and requests another resource that cannot be immediately allocated to it?

4. Multithreaded programs are?

5. The request and release of resources are____________.

13 FIFO Page Replacement Algorithm

Preamble:

With contiguous allocation, each file has to occupy contiguous blocks on the disk. The location of a file is defined by the disk address of the first block and its length. Both sequential access and direct access are supported by the contiguous allocation.

Aim:

To implement page replacement algorithms FIFO (First In First Out)

Theory:

The simpler page replacement algorithm is a FIFO algorithm. A FIFO replacement algorithm associates with each page the time when that page was brought into memory. When a page must be replaced, the oldest page is chosen. We can create a FIFO queue to hold all pages in memory. We replace the page at the head of the queue when a page is brought into memory; we insert it at the tail of the queue.

7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
7	7	7	2		2	2	4	4	4	0			0	0			7	7	7
	0	0	0		3	3	3	2	2	2			1	1			1	0	0
		1	1		4	0	0	0	3	3			3	2			2	2	1

Algorithm:

FIFO:
Step 1: Create a queue to hold all pages in memory
Step 2: When the page is required replace the page at the head of the queue
Step 3: Now the new page is inserted at the tail of the queue

/* FIFO Page Replacement Algorithm */

Program:

```
#include<stdio.h>
#include<conio.h>
int i,j,nof,nor,flag=0,ref[50],frm[50],pf=0,victim=-1;
void main()
{
 clrscr();
 printf("\n \t\t\t FIFI PAGE REPLACEMENT ALGORITHM");
 printf("\n Enter no.of frames... ");
 scanf("%d",&nof);
 printf("Enter number of reference string..\n");
 scanf("%d",&nor);
 printf("\n Enter the reference string..");
 for(i=0;i<nor;i++)
 scanf("%d",&ref[i]);
 printf("\nThe given reference string:");
 for(i=0;i<nor;i++)
 printf("%4d",ref[i]);
 for(i=1;i<=nof;i++)
 frm[i]=-1;
 printf("\n");
 for(i=0;i<nor;i++)
 {
   flag=0;
   printf("\n\t Reference np%d->\t",ref[i]);
   for(j=0;j<nof;j++)
   {
     if(frm[j]==ref[i])
     {
       flag=1;
       break;
     }}
      if(flag==0)
   {
     pf++;
     victim++;
     victim=victim%nof;
     frm[victim]=ref[i];
     for(j=0;j<nof;j++)
     printf("%4d",frm[j]);
   }
 }
 printf("\n\n\t\t No.of pages faults...%d",pf);
 getch();
}
```

Output:
Page Replacement Algorithm

```
Enter no.of frames... 4
Enter number of reference string..
6
 Enter the reference string..
5 6 4 1 2 3

The given reference string:
        ......................................  5  6  4   1   2   3

         Reference np5->          5 -1 -1 -1
         Reference np6->           5 6 -1 -1
         Reference np4->            5 6 4 -1
         Reference np1->          5 6  4   1
         Reference np2->          2 6  4   1
         Reference np3->          2 3  4   1

              No.of pages faults...6
```

Applications:

1. Disk swaping.

Viva Questions:

1. Which page replacement algorithms suffer from Belady’s Anomaly?
2. A memory page containing a heavily used variable that was initialized very early and is in constant use is removed, then the page replacement algorithm used is?
3. A FIFO replacement algorithm associates with each page the _________
4. How many page faults does the FIFO page replacement algorithm produce?
5. There is a set of page replacement algorithms that can never exhibit Belady’s Anomaly, called?

Sample Questions:

1. The essential content(s) in each entry of a page table is / are?
2. Segment replacement algorithms are more complex than page replacement algorithms because?
3. A page fault occurs when?
4. When a process begins execution with no pages in memory?
5. What do mean by Page Replacement algorithm? FIFO Stands for?

14 LRU Page Replacement Algorithm

Preamble:

For this, first we note the page number that caused the "success". Then we shift the page from right position towards left position up to the location where that noted page number was residing. Now, right most position is free. So we enter this noted page number into this empty frame location.

Aim:

To implement page replacement algorithm LRU (Least Recently Used)

Theory:

LRU (Lease Recently Used)

Here we select the page that has not been used for the longest period of time.

The implementation of the LRU is same as that of the FIFO with only one difference. When "Success" occurs in LRU, then that particular page number should be kept at the right most position.

For this, first we note the page number that caused the "success". Then we shift the page from right position towards left position up to the location where that noted page number was residing. Now, right most position is free. So we enter this noted page number into this empty frame location.

In this the right most position is the array index of (n-l), and the leftmost position is array index of “0” where n-number of frames available in the system.

Eg. LRU

Suppose consider the following contents of frames array (i.e. from FIFO output example),

5 6 8 1 7

and the next incoming page number is 8 then there is a success. Now we have to n the page number and its position. ,

Page number is 5

Position is 2 (starting from left as zeroth position)

Now we shift the array contents from (n-l)th position i.e. 4th position up to 2nd position that means here only two entries will be shifted and it will look as

5 6 1 7 □

the entry with □ is empty. So we have to fill this by the noted page number i.e. 8.

Then the frames array contents now look as:

5 6 1 7 8

With this operation, the most recently used page is kept at the right position and Lease Recently Used page is kept at the left position.

When a fault occurs, then the entire array is shifted towards left and the new page is entered at the right page. Here the operation is same as that of the FIFO. Except that, the Least Recently Used page only will go off.

Algorithm:

Step 1: Create a queue to hold all pages in memory

Step 2: When the page is required replace the page at the head of the queue

Step 3: Now the new page is inserted at the tail of the queue

Step 4: Create a stack

Step 5: When the page fault occurs replace page present at the bottom of the stack

/* LRU Page Replacement Algorithm */

Program:

```
#include<stdio.h>
#include<conio.h>
int i,j,nof,nor,flag=0,ref[50],frm[50],pf=0,victim=-1;
int recent[10],lrucal[50],count=0;
int lruvictim();
void main()
{
  clrscr();
  printf("\n\t\t\t LRU PAGE REPLACEMENT ALGORITHM");
  printf("\n Enter no.of Frames... ");
  scanf("%d",&nof);

  printf(" Enter no.of reference string..");
  scanf("%d",&nor);

  printf("\n Enter reference string..");
  for(i=0;i<nor;i++)
  scanf("%d",&ref[i]);

  printf("\n\n\t\t LRU PAGE REPLACEMENT ALGORITHM ");
  printf("\n\t     The       given       reference       string:");
  printf("\n ................................................ ");
  for(i=0;i<nor;i++)
  printf("%4d",ref[i]);
  for(i=1;i<=nof;i++)
  {
    frm[i]=-1;
    lrucal[i]=0;
  }
  for(i=0;i<10;i++)
   recent[i]=0;
   printf("\n");
   for(i=0;i<nor;i++)
   {
     flag=0;
     printf("\n\t Reference NO %d->\t",ref[i]);
     for(j=0;j<nof;j++)
     {
        if(frm[j]==ref[i])
       {
              flag=1;
              break;
       }
     }
        if(flag==0)
     {
       count++;
       if(count<=nof)
       victim++;
       else
       victim=lruvictim();
       pf++;
       frm[victim]=ref[i];
       for(j=0;j<nof;j++)
       printf("%4d",frm[j]);
     }
     recent[ref[i]]=i;
   }
   printf("\n\n\t No.of page faults...%d",pf);
   getch();
}
int lruvictim()
{
  int i,j,temp1,temp2;
  for(i=0;i<nof;i++)
  {
    temp1=frm[i];
    lrucal[i]=recent[temp1];
  }
  temp2=lrucal[0];
  for(j=1;j<nof;j++)
  {
    if(temp2>lrucal[j])
    temp2=lrucal[j];
  }
  for(i=0;i<nof;i++)
  if(ref[temp2]==frm[i])
  return i;
  return 0;
}
```

Output:

```
LRU Page Replacement Algorithm
Enter no.of Frames... 3
Enter no.of reference string.6
Enter reference string.
6 5 4 2 3 1

    LRU Page Replacement Algorithm

     The given reference string:
       ………………….6  5  4   2   3  1

     Reference NO 6->        6 -1  -1
     Reference NO 5->        6 5  -1
     Reference NO 4->        6 5  4
     Reference NO 2->        2 5  4
     Reference NO 3->        2 3  4
     Reference NO 1->        2 3  1

     No.of page faults...6
```

Applications:

1. Used in memory hardware reference.

Viva Questions:

1. In the stack implementation of the LRU algorithm, a stack can be maintained in a manner?
2. When using counters to implement LRU, we replace the page with the?
3. Increasing the RAM of a computer typically improves performance because?
4. Applying the LRU page replacement to the following reference string:
   1 2 4 5 2 1 2 4
   the main memory can accommodate 3 pages and it already has pages 1 and 2. Page 1 came in before page 2.How many page faults will occur?
5. LRU page – replacement algorithm associates with each page the ________

Sample Questions:

1. The two methods how LRU page replacement policy can be implemented in hardware are?
2. LRU stands for?
3. For 3 page frames, the following is the reference string:
   7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1
   How many page faults does the LRU page replacement algorithm produce?
4. User's________that their processes are running on a paged system?
5. The aim of creating page replacement algorithms is to?

15 Optimal (LFU) Page Replacement Algorithms

Preamble:

The LFU page replacement algorithm requires that the page with the smallest count can be replaced. This algorithm suffers from the situation in which a page is used heavily during the initial page of the process, but never used again. One solution is to shift the counts right by one bit at regular intervals forming an exponential decaying average usage count.

Aim:

To implement page replacement algorithms Optimal (The page which is not used for longest time)

Procedure:

Optimal algorithm here we select the page that will not be used for the longest period of time.

Algorithm:

OPTIMAL:

Step 1: Create a array

Step 2: When the page fault occurs replace page that will not be used for the longest
period of time

/* Optimal (LFU) Page Replacement Algorithm */

Program:

```
#include<stdio.h>
#include<conio.h>
int i,j,nof,nor,flag=0,ref[50],frm[50],pf=0,victim=-1;
int recent[10],optcal[50],count=0;
int optvictim();
void main()
{
   clrscr();
   printf("\n OPTIMAL PAGE REPLACEMENT ALGORITHN");
   printf("\n ................................ ");
   printf("\nEnter the no.of frames");
   scanf("%d",&nof);
   printf("Enter the no.of reference string");
   scanf("%d",&nor);
   printf("Enter the reference string");
   for(i=0;i<nor;i++)
       scanf("%d",&ref[i]);
   clrscr();
   printf("\n OPTIMAL PAGE REPLACEMENT ALGORITHM");
   printf("\n .............................. ");
   printf("\nThe given string");
   printf("\n ...................\n");
   for(i=0;i<nor;i++)
       printf("%4d",ref[i]);
   for(i=0;i<nof;i++)
   {
       frm[i]=-1;
       optcal[i]=0;
   }
   for(i=0;i<10;i++)
       recent[i]=0;
   printf("\n");
   for(i=0;i<nor;i++)
   {
      flag=0;
      printf("\n\tref no %d ->\t",ref[i]);
      for(j=0;j<nof;j++)
      {
            if(frm[j]==ref[i])
            {
               flag=1;
               break;
            }
      }
      if(flag==0)
      {
            count++;
            if(count<=nof)
                victim++;
            else
                victim=optvictim(i);
            pf++;
            frm[victim]=ref[i];
            for(j=0;j<nof;j++)
               printf("%4d",frm[j]);
      }
   }
   printf("\n Number of page faults: %d",pf);
   getch();
}
int optvictim(int index)
{
   int i,j,temp,notfound;
   for(i=0;i<nof;i++)
   {
      notfound=1;
      for(j=index;j<nor;j++)
            if(frm[i]==ref[j])
            {
                notfound=0;
                optcal[i]=j;
                break;
             }
      if(notfound==1)
               return i;
   }
   temp=optcal[0];
   for(i=1;i<nof;i++)
       if(temp<optcal[i])
               temp=optcal[i];
   for(i=0;i<nof;i++)
       if(frm[temp]==frm[i])
               return i;
 return 0;
}
```

Output:

Optimal (LFU) Page Replacement Algorithms

```
 Enter no.of Frames... 3
 Enter no.of reference string.. 6

 Enter reference string..
 6 5 4 2 3 1
```

Optimal (LFU) Page Replacement Algorithms

```
         The given reference string:
          …………………. 6   5  4   2  3   1

         Reference NO 6->          6 -1  -1
         Reference NO 5->          6 5  -1
         Reference NO 4->          6 5  4
         Reference NO 2->          2 5  4
         Reference NO 3->          2 3  4
         Reference NO 1->          2 3  1

         No.of page faults...6
```

Applications:

1. Bench Marking.

Viva Questions

1. What is mean by segmentation?

2. What is mean by reference string?

3. LRU Stands for?

4. What is optimal page replacement algorithm?

5. What is paging?

Sample Questions:

1. Optimal page – replacement algorithm is difficult to implement, because

2. LRU page – replacement algorithm associates with each page the ________

3, The implementation of the LFU and the MFU algorithm is very uncommon because

4. The reason for using the LFU page replacement algorithm is

5. The essential content(s) in each entry of a page table is / are

16 Implement Paging

Preamble:

Paging is an efficient memory management scheme because it is non-contiguous memory allocation method. The basic idea of paging is the physical memory (main memory) is divided into fixed sized blocks called frames, the logical address space is divided into fixed sized blocks, called pages, but page size and frame size should be equal. The size of the frame or a page is depending on operating system

Aim:
To implement the Memory management policy- Paging.

Theory:

The LFU page replacement algorithm requires that the page with the smallest count can be replaced. This algorithm suffers from the situation in which a page is used heavily during the initial page of the process, but never used again. One solution is to shift the counts right by one bit at regular intervals forming an exponential decaying average usage count.

1. first-fit method

These algorithms can be easily implemented using singly linked list operations. In this we consider simulation of 640 (kb) memory allocation for jobs which is a MS-DOS recognized memory.

data structure§ required:

There is structure for Node definition in the singly linked list. The structure define is as follows...

```
struct job_node {
int job_no;
int start_Ioc;
int end_Ioc;
int flag;
struct job_node *next;
}
typedef struct job_node NODE;
```

In above definition the different fields signifies the following...

Field 1. job_no : indicates the number' of the job present.

2. starat_Ioc : indicates the starting location of the job in the memory.

3. end_Ioc : indicates the ending location in the memory.

The difference between the starting location and ending location must be in multiples of 00 only. (A number indicates in Kilobytes).

4. flag: signifies whether the job is present i.e. memory is free or not.

5. *next : the pointer to the structure of type struct job_node.

And there should be a counter for counting the number of valid jobs i.e. number of jobs actually present. One pointer of NODE type is necessary for keeping track of the starting node of singly linked list. The functions required are for:

1. Adding jobs,
2. Deleting jobs,
3. Displaying jobs (display memory map), :.
4. Displaying the job tables.

There should be always be at least two nodes on the list or we can say that initially there are two nodes on the list. The starting node consists of allocated memory for operating system (20KB).

The second node consists of the free memory of 620KB. (Initially there is only one 'hole') That means the total memory should be considered is 640KB which is a MS- DOS recognized memory.

Implemetation:

Adding the Jobs:

When a job is to be added then we will first read number and size. Then we check for the availability of memory i.e. simply we check the singly linked list. For example we wish to add a job of size of70kb initially the operations are as follows.

First we check the starting node, the OS is present, and this we come to know because the flag bit is set to TRUE. Then we go to the next node which is free -" because the flag in that node FALSE. So we will insert a node in between the two nodes and we give values for newly created node as follows...

job_no = 1
start loc = 20
end loc = 90
flag = TRUE

And in the last node we have to reduce the available memory size by 70 (KB)

Now there are three node in the list. When we want to insert a job of 100 (KB), same procedure as above is followed. Job counter is incremented accordingly.

Deletion of jobs:

When we want to delete job, we make the flag FALSE and decrement the counter accordingly. Now the free memory area present in this node i.e. it is a 'hole'. Let us say that the 70 (KB) job is deleted. Then the memory map is as follows.

```
Os   OS  0   0   0   0   0  0    0   2
2     2   2   2  2   2   2  2    2   0
0     0   0   0  0   0   0  0    0   0
0     0   0   0  0   0   0  0    0   0
0     0   0   0  0   0   0  0    0   0
0     0   0   0  0   0   0  0    0   0
0     0   0   0
```

The above map shows every 10(kb) in the memory and which job is present in that 10 (kb). The job table is as follows:

JOB ALLOCATION TABLE

JOB-NO	**JOB SIZE**	**JOB LOCATION**
2	100(KB)	90(KB)

FREE AREA TABLE

PARTITION-NO	**AREA SIZE**	**AREA LOCATION**
3	320(KB)	420(KB)

Since the job 1 is deleted it is not displayed. In this way the jobs are deleted and added.

As the jobs are added and deleted, at any given instant of time there will be a set a holes and allocated blocks, the set of holes are searched and if such a hole is found then the new job is inserted into that hole.

Results, Menus and Program Output should be as follows:

Program for First-Fit Method

Menu

I> Adding job
2> Deleting job
3> Display of Memory Map
4> Display Job Table
5> Exit

enter your choice:

Suppose when the jobs of sizes 70 (kb), 100(kb), 120(kb), 5O(kb), and 60(kb) added successively and serially. Then the job table for choice 4 in the menu, the output is:

JOB ALLOCATION TABLE

JOB-NO	JOB SIZE	JOB LOCATION
1	70(KB)	20(KB)
2	lOO(KB)	90(KB)
3	120(KB)	190(KB)
4	5O(KB)	300{KB)
5	60(KB)	360(KB)

FREE AREA TABLE

PARTITION NO	AREA SIZE	AREA LOCATION
6	220(KB)	420(KB)

Now the memory map is as follows:

```
Os  os  1  1  1  1  1  1  1  2
2    2  2  2  2  2  2  2  2  2
3    3  3  3  3  3  3  3  3  3
3    4  4  4  4  4  5  5  5  5
5    5  0  0  0  0  0  0  0  0
0    0  0  0  0  0  0  0  0  0
0    0  0  0
```

Now, let us say, when the job 1 and 4 are deleted. Then the job allocation table is as follows:

JOB ALLOCATION TABLE

JOB-NO	JOB SIZE	JOB LOCATION
2	100(KB)	90(KB)
3	l20(KB)	190(KB)
5	60(KB)	360(KB)

FREE AREA TABLE

PARTITION NO	AREA SIZE	AREA LOCATION
1	70(KB)	20(KB)
4	60(KB)	310(KB)
6	220(KB)	420(KB)

Deleting a job means not actually deleting Node or not Freeing the Node, but changing the flag to FALSE and decrementing the counter. When the list is traversed, and the flag is checked then the partition is treated as free partition (No job is present in that partition) since the flag is FALSE. For deleting a job is simply we'll read the job no from user. Now when a job is of 60 (KB) is to inserted then it will be inserted at the partition 1 according to first-fit method.

When a job of 80 (KB) or more of this size is to be inserted then the last hole is split into two holes of which one would be of size 80(kb) and the job is inserted into this new hole.

Algorithm:

```
Step 1: Read all the necessary input from the keyboard.
Step 2: Pages - Logical memory is broken into fixed - sized blocks.
Step 3: Frames – Physical memory is broken into fixed – sized blocks.
Step 4: Calculate the physical address using the following
            Physical address = ( Frame number * Frame size ) + offset
Step 5: Display the physical address.
Step 6: Stop the process.
```

/* Memory Allocation with Paging Technique */

```
#include <stdio.h>
#include <conio.h>
struct pstruct
{
            int fno;
            int pbit;
}ptable[10];

int pmsize,lmsize,psize,frame,page,ftable[20],frameno;

void info()
```

```
{
        printf("\n\nMEMORY MANAGEMENT USING PAGING\n\n");
        printf("\n\nEnter the Size of Physical memory: ");
        scanf("%d",&pmsize);
        printf("\n\nEnter the size of Logical memory: ");
        scanf("%d",&lmsize);
        printf("\n\nEnter the partition size: ");
        scanf("%d",&psize);
        frame = (int) pmsize/psize;
        page = (int) lmsize/psize;
        printf("\nThe physical memory is divided into %d no.of frames\n",frame);
        printf("\nThe Logical memory is divided into %d no.of pages",page);
}

void assign()
{
        int i;
        for (i=0;i<page;i++)
        {
        ptable[i].fno = -1;
        ptable[i].pbit= -1;
        }
        for(i=0; i<frame;i++)
                ftable[i] = 32555;
        for (i=0;i<page;i++)
        {
        printf("\n\nEnter the Frame number where page %d must be placed: ",i);
                scanf("%d",&frameno);
                ftable[frameno] = i;
                if(ptable[i].pbit == -1)
                {
                        ptable[i].fno = frameno;
                        ptable[i].pbit = 1;
                }
        }
        getch();
    //  clrscr();
        printf("\n\nPAGE TABLE\n\n");
        printf("PageAddress FrameNo. PresenceBit\n\n");
        for (i=0;i<page;i++)
                printf("%d\t\t%d\t\t%d\n",i,ptable[i].fno,ptable[i].pbit);
        printf("\n\n\n\tFRAME TABLE\n\n");
        printf("FrameAddress PageNo\n\n");
        for(i=0;i<frame;i++)
                printf("%d\t\t%d\n",i,ftable[i]);
}
void cphyaddr()
{
        int laddr,paddr,disp,phyaddr,baddr;
        getch();
    //  clrscr();
        printf("\n\n\n\tProcess to create the Physical Address\n\n");
        printf("\nEnter the Base Address: ");
        scanf("%d",&baddr);
        printf("\nEnter theLogical Address: ");
        scanf("%d",&laddr);

        paddr = laddr / psize;
        disp = laddr % psize;
         if(ptable[paddr].pbit == 1 )
                phyaddr = baddr + (ptable[paddr].fno*psize) + disp;
        printf("\nThe Physical Address where the instruction present: %d",phyaddr);
}
void main()
{
        clrscr();
        info();
        assign();
        cphyaddr();
        getch();
}
```

Output:

```
MEMORY MANAGEMENT USING PAGING
 Enter the Size of Physical memory: 16

Enter the size of Logical memory: 8

Enter the partition size: 2

The physical memory is divided into 8 no.of frames

The Logical memory is divided into 4 no.of pages

Enter the Frame number where page 0 must be placed: 5

Enter the Frame number where page 1 must be placed: 6

Enter the Frame number where page 2 must be placed: 7

Enter the Frame number where page 3 must be placed: 2


 PAGE TABLE

 PageAddress        FrameNo.          PresenceBit

       0               5                  1
       1               6                  1
       2               7                  1
       3               2                  1

FRAME TABLE
```

```
 FrameAddress          PageNo

       0              32555
       1              32555
       2                3
       3              32555
       4              32555
       5                0
       6                1
       7                2

Process to create the Physical Address
Enter the Base Address: 1000
Enter theLogical Address: 3

The Physical Address where the instruction present: 1013
```

Applications:

1. Used in Fragmentation.

Viva Questsions:

1. With paging there is no_________fragmentation.

2. Paging increases the_______time.

3. The operating system maintains a_______table that keeps track of how many frames have been allocated, how many are there, and how many are available.

4. Smaller page tables are implemented as a set of________.

5. The size of a page is typically

Sample Questions:

1. Each entry in a Translation look-aside buffer (TLB) consists of

2. The percentage of times a page number is found in the TLB is known as

3. Memory protection in a paged environment is accomplished by

4. When there is a large logical address space, the best way of paging would be?

5. In paged memory systems, if the page size is increased, then the internal fragmentation generally?

17Segmentation

Aim:
To implement the memory management policy-segmentation.

Algorithm:

```
Step 1:  Start the program.
Step 2:  Get the number of segments.
Step 3:  get the base address and length for each segment.
Step 4: Get the logical address.
Step 5: check whether the segment number is within the limit, if not display the error
        message.
Step 6: Check whether the byte reference is within the limit, if not display the error
        message.
Step 7:  Calculate the physical memory and display it.
Step 8: Stop the program.
```

/*Memory Segment Table*/

Program:

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
int sost;
void gstinfo();
void ptladdr();

struct segtab
{
        int sno;
        int baddr;
        int limit;
        int val[10];
}st[10];

void gstinfo()
{
        int i,j;
        printf("\n\tEnter the size of the segment table: ");
        scanf("%d",&sost);

        for(i=1;i<=sost;i++)
                {
                printf("\n\tEnter the information about segment: %d",i);
                st[i].sno = i;
                printf("\n\tEnter the base Address: ");
                scanf("%d",&st[i].baddr);
                printf("\n\tEnter the Limit: ");
                scanf("%d",&st[i].limit);
                for(j=0;j<st[i].limit;j++)
                        {
                        printf("Enter the %d address Value: ",(st[i].baddr + j));
                        scanf("%d",&st[i].val[j]);
                        }
                }
}

void ptladdr()
{
        int i,swd,d=0,n,s,disp,paddr;
        clrscr();
        printf("\n\n\t\t\t SEGMENT TABLE \n\n");
        printf("\n\t      SEG.NO\tBASE ADDRESS\t      LIMIT \n\n");
        for(i=1;i<=sost;i++)
                printf("\t\t%d    \t\t%d\t\t%d\n\n",st[i].sno,st[i].baddr,st[i].limit);
        printf("\n\nEnter the logical Address: ");
        scanf("%d",&swd);
        n=swd;
        while (n != 0)
        {
                n=n/10;
                d++;
        }

        s = swd/pow(10,d-1);
        disp = swd%(int)pow(10,d-1);

        if(s<=sost)
        {
                if(disp < st[s].limit)
                {
                        paddr = st[s].baddr + disp;
                        printf("\n\t\tLogical Address is: %d",swd);
                        printf("\n\t\tMapped Physical address is: %d",paddr);
                        printf("\n\tThe value is: %d",( st[s].val[disp] ) );
                }
                else
                        printf("\n\t\tLimit of segment %d is high\n\n",s);
        }

        else
                printf("\n\t\tInvalid Segment Address \n");
        }

        void main()
        {
        char ch;
        clrscr();
```

```
gstinfo();
do
{
ptladdr();
printf("\n\t Do U want to Continue(Y/N)");
flushall();
scanf("%c",&ch);
}while (ch == 'Y' || ch == 'y' );

getch();
}
```

OUTPUT:

```
        Enter the size of the segment table: 3
        Enter the information about segment: 1
        Enter the base Address: 4
        Enter the Limit: 5
Enter the 4 address Value: 11
Enter the 5 address Value: 12
Enter the 6 address Value: 13
Enter the 7 address Value: 14
Enter the 8 address Value: 15

        Enter the information about segment: 2
        Enter the base Address: 5

        Enter the Limit: 4
Enter the 5 address Value: 21
Enter the 6 address Value: 31
Enter the 7 address Value: 41
Enter the 8 address Value: 51

Enter the information about segment: 3
 Enter the base Address: 3

        Enter the Limit: 4
Enter the 3 address Value: 31
Enter the 4 address Value: 41
Enter the 5 address Value: 41
Enter the 6 address Value: 51

                  SEGMENT TABLE

        SEG.NO    BASE ADDRESS        LIMIT

          1            4                5

          2             5               4

          3            3                4
 Enter the logical Address: 3
Logical Address  is:  3
Mapped Physical address is: 3
The value is: 31
Do U want to Continue(Y/N)


        SEGMENT TABLE


SEG.NO   BASE ADDRESS      LIMIT

  1          4               5

  2          5               4

  3          3               4


Enter the logical Address: 1


 Logical Address  is: 1

 Mapped Physical address is: 4

 The value is: 11

 Do U want to Continue(Y/N)
```

Applications:

1. Used in shared library.
2. Used in coding.

Viva Questions:

1. What is best-fit method?
2. What is worst-fit method?
3. The segment limit contains the?
4. How we can calculate Area size?
5. How we can calculate Area locations?

Sample Questions:

1. In segmentation, each address is specified by?
2. In paging the user provides only_________, which is partitioned by the hardware into __________ and_.
3. Each entry in a segment table has a

4. The segment base contains the

5. When the entries in the segment tables of two different processes point to the same physical location?

Lab Manual
Data Engineering with Python

R24: (2024-2026)

R24LCC 252

Data Engineering with Python Lab

Credits: 1.5

With effect from the academic year 2024-2026

Data Engg. With Python Lab (R24 LCC252)

Credits: 1.5

Instruction CIE	Duration of SEE SEE
3P hrs per week 25 marks	3 hours 50 marks

Course objectives:

- Understand the process of Importing and Exporting the data.
- Learn how to collect, store and manage data from multiple data sources.
- Know the insights of data using statistical methods
- Identify different techniques for data analysis and data visualization.
- Put into practice the ETL (extract, transform, load) pipeline which will extract raw data, clean the data, perform transformations on data, load data and visualize the data.

Course Outcomes: students would be able to:

- Demonstrate various data types in python and develop programs using files, exception
- handling, functions, classes in Python.
- Examine the process for importing and exporting the data.
- Apply appropriate data collection and pre-processing methods.
- Identify different data analysis Techniques suitable for a given applications
- Demonstrate data visualization techniques for Data Analysis.

Libraries

In this course students are expected to extract, transform and load input data that can be text files, CSV files, XML files, JSON, HTML files, SQL databases, NoSQL databases etc.,. For doing this, they should learn the following Python libraries/modules:
pandas, numpy, BeautifulSoup, pymysql, pymongo, nltk, matplotlib

Datasets

For this laboratory, appropriate publicly available datasets, can be studied and used. Example:
MNIST (http://yann.lecun.com/exdb/mnist/),
UCI Machine Learning Repository(https://archive.ics.uci.edu/ml/datasets.html),
Kaggle(https://www.kaggle.com/datasets)
Twitter Data

Exercises

1. Write programs to parse text files, CSV, HTML, XML and JSON documents and extract relevant data. After retrieving data check any anomalies in the data, missing values etc.
2. Write programs for reading and writing binary files
3. Write programs for searching, splitting, and replacing strings based on pattern matching using regular expressions

4. Design a relational database for a small application and populate the database. Using SQL do the CRUD (create, read, update and delete) operations.
5. Create a Python MongoDB client using the Python module pymongo. Using a collection object practice functions for inserting, searching, removing, updating, replacing, and aggregating documents, as well as for creating indexes
6. Write programs to create numpy arrays of different shapes and from different sources, reshape and slice arrays, add array indexes, and apply arithmetic, logic, and aggregation functions to some or all array elements
7. Write programs to use the pandas data structures: Frames and series as storage containers and for a variety of data-wrangling operations, such as:
   - Single-level and hierarchical indexing
   - Handling missing data
   - Arithmetic and Boolean operations on entire columns and tables
   - Database-type operations (such as merging and aggregation)
   - Plotting individual columns and whole tables
   - Reading data from files and writing data to files

Additional Exercises (for learning and practice) :

1. Introduction to Python Programming:
   a) Running instructions in Interactive interpreter and a Python Script.
   b) Write a program to purposefully raise Indentation Error and Correct it
   c) Write a program to compute distance between two points taking input from the user
   d) Write a program add python that takes 2numbers as command line arguments and prints its sum.
   e) Program to display the following information: Your name, Full Address, Mobile Number, College Name, Course Subjects
   f) Write a Program for checking whether the given number is an even number or not.
2. Control Structures, Lists
   a) Program to find the largest three integers using if-else
   b) Program that receives a series of positive numbers and display the numbers in order and their sum
   c) Program to find the product of two matrices and
   d) Program to display two random numbers that are to be added, the program should allow the student to enter the answer.
      If the answer is correct, a message of congratulations should be displayed.
      If the answer is incorrect, the correct answer should be displayed.
   e) Using a for loop, write a program that prints out the decimal equivalents of 1/2,1/3,1/4, .1/10.
   f) Write a program using a while loop that asks the user for a number, and prints a countdown from that number to zero.
3. Functions and Recursion - Write recursive and non-recursive functions for the following

a) To find GCD of two integers
b) To find the factorial of positive integer
c) To print Fibonacci Sequence up to given number n
d) To display prime number from 2 to n.
e) Function that accepts two arguments: a list and a number n. It displays all of the numbers in the list that are greater than n
f) Functions that accept a string as an argument and return the number of vowels and consonants that the string contains

4. Files, Exceptions, Lists, Sets, Random Numbers
   a) Program to write a series of random numbers in a file from 1 to n and display.
   b) Program to write the content in a file and display it with a line number followed by a colon
   c) Program to display a list of all unique words in a text file
   d) Program to analyze the two text files using set operations
   e) Write a program to print each line of a file in reverse order.
   f) Write a program to count frequency of characters in a given file. Can you use character frequency total whether the given file is a Python program file, C program file or a text file?
   g) Write a program combine lists that combines these lists in to a dictionary.
5. Object Oriented Programming
   a) Program to implement the inheritance
   b) Program to implement the polymorphism
6. Demonstrate data analysis using NumPy
   a) Create an array of 10 zeros
   b) Create an array of even integers upto 50
   c) Create a 3x3 matrix
   d) Generate an array of 25 random numbers sampled from a standard normal distribution.
   e) Create an array of 20 linearly spaced points between 0 and 1
   f) Demonstrate slicing and indexing operations
   g) Get the sum of all columns in matrix
7. Write a Program in Python to create and combine student and subject data frames in Pandas.
8. Create a data frame ‘Book’ that contains three vectors [Name, Price, Author]. Convert this data frame into a matrix and list the object using the operator ‘as’.
9. Performing Exploratory data analysis on web scraped data of 2021-22 NBA player stats (http://www.basketball-reference.com/)
   a. Perform data cleaning
   b. Handle missing values by replacing with 0
   c. Write to CSV file
   d. Which player scored the most points per game?
   e. Which player had the highest 3-point field goals per game?

    f. Demonstrate Group By() function

10. Data visualization through Sea born for the above program 9.
    a. Box plot of points scored grouped by position
    b. Compute the correlation matrix
11. To determine the mean of a set of numbers. To plot the numbers in a bar plot and have a straight line run through the plot at the mean.
12. To determine the median of a set of numbers. To plot the numbers in a barplot and have a straight line run through the plot at the median.
13. To determine the standard deviation. To plot the numbers in a bar plot and have a straight line run through the plot at the mean and another straight line run through the plot at mean + standard deviation.

More dataset to perform data analysis

Source of the Data: https://www.kaggle.com/chirin/africa-economic-banking-and-systemic-crisis-data/downloads/africa-economic-banking-and-systemic-crisis-data.zip/1

Data set: https://www.kaggle.com/khalidative/crimeanalysis

Table of Contents

1. **Write programs to parse text files, CSV, HTML, XML and JSON documents and extract relevant data. After retrieving data check any anomalies in the data, missing values etc.**

A. Text File

ReadWriteTextFile.py

```
# Open a file
fo = open("text.txt", "w")
fo.write( "Python is a great language.\nYeah its great!!\n")
# Close opend file
fo.close()
fo = open("text.txt", "r+")
str = fo.read();
print("Read String is : ", str)
# Close opend file
fo.close()
```

Output

```
Python is a great language.
Yeah its great!!
```

Text.txt

```
Python is a great language.
Yeah its great!!
```

B. CSV Files

Iris.csv Dataset

```
Id,SepalLengthCm,SepalWidthCm,PetalLengthCm,PetalWidthCm,Species
1,5.1,3.5,1.4,0.2,Iris-setosa
2,4.9,3.0,1.4,0.2,Iris-setosa
3,4.7,3.2,1.3,0.2,Iris-setosa
4,4.6,3.1,1.5,0.2,Iris-setosa
5,5.0,3.6,1.4,0.2,Iris-setosa
…
146,6.7,3.0,5.2,2.3,Iris-virginica
147,6.3,2.5,5.0,1.9,Iris-virginica
148,6.5,3.0,5.2,2.0,Iris-virginica
149,6.2,3.4,5.4,2.3,Iris-virginica
150,5.9,3.0,5.1,1.8,Iris-virginica
```

ReadCSV.py

```
import csv
filename = 'Iris.csv'
fields = []
rows = []
# Reading csv file
with open(filename, 'r') as csvfile:
    # Creating a csv reader object
    csvreader = csv.reader(csvfile, delimiter=',', quotechar='"', skipinitialspace=True)
    # Extracting field names in the first row
    fields = next(csvreader)
    # Extracting each data row one by one
    for row in csvreader:
        rows.append(row)
# Printing out the first 5 rows
for row in rows[:5]:
    print(row)
```

Output

```
['1', '5.1', '3.5', '1.4', '0.2', 'Iris-setosa']
['2', '4.9', '3.0', '1.4', '0.2', 'Iris-setosa']
['3', '4.7', '3.2', '1.3', '0.2', 'Iris-setosa']
['4', '4.6', '3.1', '1.5', '0.2', 'Iris-setosa']
['5', '5.0', '3.6', '1.4', '0.2', 'Iris-setosa']
```

WriteCSV.py

```
import csv
# Field names
fields = ['Name', 'Goals', 'Assists', 'Shots']
# Rows of data in the csv file
rows = [['Emily', '12', '18', '112'],
        ['Katie', '8', '24', '96'],
        ['Arjun', '16', '9', '101'],
        ['Mike', '3', '14', '82']]
filename = "soccer.csv"
# Writing to csv file
with open(filename, 'w+') as csvfile:
    # Creating a csv writer object
    csvwriter = csv.writer(csvfile)
    # Writing the fields
    csvwriter.writerow(fields)
    # Writing the data rows
    csvwriter.writerows(rows)
```

Output

Soccer.csv

```
Name,Goals,Assists,Shots
Emily,12,18,112
Katie,8,24,96
Arjun,16,9,10
1
Mike,3,14,82
```

ReadCSVPandas.py

```
import pandas as pd
filename = “Iris.csv”
# Read in the data
data = pd.read_csv(filename)
# Print the first 5 rows
print(data.head(5))
# Write the data to file
data.to_csv(“new_data.csv”, sep=”,”, index=False)
```

Output

```
Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm      Species
0   1           5.1          3.5          1.4          0.2  Iris-setosa
1   2           4.9          3.0          1.4          0.2  Iris-setosa
2   3           4.7          3.2          1.3          0.2  Iris-setosa
3   4           4.6          3.1          1.5          0.2  Iris-setosa
4   5           5.0          3.6          1.4          0.2  Iris-setosa
```

new_data.csv

```
Id SepalLengthCm SepalWidthCm PetalLengthCm PetalWidthCm Species

 1           5.1           3.5           1.4          0.2 Iris-setosa

 2           4.9             3           1.4          0.2 Iris-setosa

 3           4.7           3.2           1.3          0.2 Iris-setosa

 4           4.6           3.1           1.5          0.2 Iris-setosa
```

…

C. HTML

ReadWriteHTMLFile.py

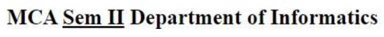
```
# Creating an HTML file
Func = open("hypertext.html", "w")
# Adding input data to the HTML file
Func.write("<html>\n<head>\n<title> \nOutput Data in an HTML file \
    </title>\n</head> <body><h1>Welcome to <u>Aurora College</u></h1>\
    \n<h2>MCA <u>Sem II</u> Department of Informatics</h2> \n</body></html>")
# Saving the data into the HTML file
Func.close()
Func = open("hypertext.html", "r")
# Adding input data to the HTML file
str = Func.read()
print("Read from HTML file - ", str)
# Saving the data into the HTML file
Func.close()
```

Output

```
Read from HTML file -  <html>
<head>
<title>
Output Data in an HTML file                    </title>
</head> <body><h1>Welcome to <u>Aurora College</u></h1>
<h2>MCA <u>Sem II</u> Department of Informatics</h2>
</body></html>
```

Hypertext.html

```
<html>
<head>
<title>
Output Data in an HTML file                    </title>
</head> <body><h1>Welcome to <u>Aurora College</u></h1>
<h2>MCA <u>Sem II</u> Department of Informatics</h2>
</body></html>
```



MCA Sem II Department of Informatics

D. XML and JSON

WriteXMLPandas.py

```
import pandas as pd
from dicttoxml import dicttoxml
import json
# Building our dataframe
data = {'Name': ['Emily', 'Katie', 'Arjun',
        'Mike'], 'Goals': [12, 8, 16, 3],
        'Assists': [18, 24, 9, 14],
        'Shots': [112, 96, 101, 82]
        }
df = pd.DataFrame(data, columns=data.keys())
# Converting the dataframe to a dictionary and Then save it to file
data_dict = df.to_dict(orient="records")
with open('output.json', "w+") as f:
    json.dump(data_dict, f, indent=4)
# Converting the dataframe to XML and Then save it to file
xml_data = dicttoxml(data_dict).decode()
with open("output.xml", "w+") as f:
    f.write(xml_data)
```

Output

Output.json

```
[
    {
        "Name": "Emily",
        "Goals": 12,
        "Assists": 18,
        "Shots": 112
    },
    {
        "Name": "Katie",
        "Goals": 8,
        "Assists": 24,
        "Shots": 96
    },
    {
        "Name": "Arjun",
        "Goals": 16,
        "Assists": 9,
        "Shots": 101
    },
    {
        "Name": "Mike",
        "Goals": 3,
        "Assists": 14,
        "Shots": 82
    }
]
```

Output.xml

```
<?xml version="1.0" encoding="UTF-8" ?><root><item type="dict"><Name
type="str">Emily</Name><Goals type="int">12</Goals><Assists
type="int">18</Assists><Shots type="int">112</Shots></item><item type="dict"><Name
type="str">Katie</Name><Goals type="int">8</Goals><Assists
type="int">24</Assists><Shots type="int">96</Shots></item><item type="dict"><Name
type="str">Arjun</Name><Goals type="int">16</Goals><Assists
type="int">9</Assists><Shots type="int">101</Shots></item><item type="dict"><Name
type="str">Mike</Name><Goals type="int">3</Goals><Assists
type="int">14</Assists><Shots type="int">82</Shots></item></root>
```

E. JSON

first.json

```
{
  "squadName": "Super hero squad",
  "homeTown": "Metro City",
  "formed": 2016,
  "secretBase": "Super tower",
  "active": true,
  "members": [
    {
      "name": "Molecule Man",
      "age": 29,
      "secretIdentity": "Dan Jukes",
      "powers": ["Radiation resistance", "Turning tiny", "Radiation blast"]
    },
    {
      "name": "Madame Uppercut",
      "age": 39,
      "secretIdentity": "Jane Wilson",
      "powers": [
        "Million tonne punch",
        "Damage resistance",
        "Superhuman reflexes"
      ]
    },
    {
      "name": "Eternal Flame",
      "age": 1000000,
      "secretIdentity": "Unknown",
      "powers": [
        "Immortality",
        "Heat Immunity",
        "Inferno",
        "Teleportation",
        "Interdimensional travel"
      ]
    }
  ]
}
```

WriteJSONPandas.py

```
import json
import pandas as pd
# Read the data from file, # We now have a Python dictionary
with open('first.json') as f:
    data_listofdict = json.load(f)
# We can do the same thing with pandas
data_df = pd.read_json('first.json', orient='records')
# We can write a dictionary to JSON like so, # Use 'indent' and 'sort_keys' to make the JSON, #
file look nice
with open('new_data1.json', 'w+') as json_file:
    json.dump(data_listofdict, json_file, indent=4, sort_keys=True)
# And again the same thing with pandas
export = data_df.to_json('new_data2.json', orient='records')
```

Output

new_data1.json

```
{
    "active": true,
    "formed": 2016,
    "homeTown": "Metro City",
    "members": [
        {
            "age": 29,
            "name": "Molecule Man",
            "powers": [
                "Radiation resistance",
                "Turning tiny",
                "Radiation blast"
            ],
            "secretIdentity": "Dan Jukes"
        },
        {
            "age": 39,
            "name": "Madame Uppercut",
            "powers": [
                "Million tonne punch",
                "Damage resistance",
                "Superhuman reflexes"
            ],
            "secretIdentity": "Jane Wilson"
        },
        {
            "age": 1000000,
            "name": "Eternal Flame",
            "powers": [
                "Immortality",
                "Heat Immunity",
                "Inferno",
                "Teleportation",
                "Interdimensional travel"
            ],
            "secretIdentity": "Unknown"
        }
    ],
    "secretBase": "Super tower",
    "squadName": "Super hero squad"
}
```

new_data2.json

```
[{"squadName":"Super hero squad","homeTown":"Metro
City","formed":2016,"secretBase":"Super tower","active":true,"members":{"name":"Molecule
Man","age":29,"secretIdentity":"Dan Jukes","powers":["Radiation resistance","Turning
tiny","Radiation blast"]}},{"squadName":"Super hero squad","homeTown":"Metro
City","formed":2016,"secretBase":"Super tower","active":true,"members":{"name":"Madame
Uppercut","age":39,"secretIdentity":"Jane Wilson","powers":["Million tonne punch","Damage
resistance","Superhuman reflexes"]}},{"squadName":"Super hero squad","homeTown":"Metro
City","formed":2016,"secretBase":"Super tower","active":true,"members":{"name":"Eternal
Flame","age":1000000,"secretIdentity":"Unknown","powers":["Immortality","Heat
Immunity","Inferno","Teleportation","Interdimensional travel"]}}]
```

ArrayJSON.json

```
[
  {
    "name": "Molecule Man",
    "age": 29,
    "secretIdentity": "Dan Jukes",
    "powers": ["Radiation resistance", "Turning tiny", "Radiation blast"]
  },
  {
    "name": "Madame Uppercut",
    "age": 39,
    "secretIdentity": "Jane Wilson",
    "powers": [
      "Million tonne punch", "Damage resistance", "Superhuman reflexes"]
  }
]
```

WriteArrayJSONPandas.py

```
import json
import pandas as pd
# Read the data from file, # We now have a Python dictionary
with open('ArrayJSON.json') as f:
    data_listofdict = json.load(f)
# We can do the same thing with pandas
data_df = pd.read_json('arrayJSON.json', orient='records')
# We can write a dictionary to JSON like so, # Use 'indent' and 'sort_keys' to make the JSON, # file look nice
with open('new_data3.json', 'w+') as json_file:
    json.dump(data_listofdict, json_file, indent=4, sort_keys=True)
# And again the same thing with pandas
export = data_df.to_json('new_data4.json', orient='records')
```

Output

new_data3.json

```
[
    {
        "age": 29,
        "name": "Molecule Man",
        "powers": [
            "Radiation resistance",
            "Turning tiny",
            "Radiation blast"
        ],
        "secretIdentity": "Dan Jukes"
    },
    {
        "age": 39,
        "name": "Madame Uppercut",
        "powers": [
            "Million tonne punch",
            "Damage resistance",
            "Superhuman reflexes"
        ],
        "secretIdentity": "Jane Wilson"
    }
]
```

new_data4.json

```
[{"name":"Molecule Man","age":29,"secretIdentity":"Dan Jukes","powers":["Radiation
resistance","Turning tiny","Radiation blast"]},{"name":"Madame
Uppercut","age":39,"secretIdentity":"Jane Wilson","powers":["Million tonne punch","Damage
resistance","Superhuman reflexes"]}]
```

2. Write programs for reading and writing binary files

A. Binary File to an Array and ByteArray

ReadWriteBinaryFile.py

```
file = open("binfile.bin","wb")
sentence = bytearray("This is good".encode("ascii"))
file.write(sentence)
file.close()
file = open("binfile.bin","rb")
print(file.read())
file.close()
```

Output

```
b'This is good'
```

binfile.bin

```
This is good
```

ReadWriteBinaryFiletoAray.py

```
#Python read a binary file to an array
file=open("array.bin","wb")
num=[2,4,6,8,10]
array=bytearray(num)
file.write(array)
file.close()
file=open("array.bin","rb")
number=list(file.read())
print (number)
file.close()
```

Output

```
[2, 4, 6, 8, 10]
```

Array.bin

It contains some non readable data

ReadWriteBinaryFileLines.py

```
#Python read a binary file line by line
lines='Welcome to python guides'
file=open("document1.txt","wb")
file.write(lines.encode())
file.close()
file=open("document1.txt","rb")
filelines=file.readline()
print(filelines)
file.close()
#Python read a binary file to Ascii
file = open('test.bin', 'w+b')
sentence = 'Hello Python'
file_encode = sentence.encode('ASCII')
file.write(file_encode)
file.seek(0)
bdata = file.read()
print('Binary sentence', bdata)
new_sentence = bdata.decode('ASCII')
print('ASCII sentence', new_sentence)
```

Ouput

```
b'Welcome to python guides'
Binary sentence b'Hello Python'
ASCII sentence Hello Python
```

document1.txt

```
Welcome to python guides
```

test.bin

```
Hello Python
```

3. Write programs for searching, splitting, and replacing strings based on pattern matching using regular expressions

```
#Regular Expressions
#A regular expression or RegEx is a special text string that helps to find patterns in data. A RegE
x can be used to check if some pattern exists in a different data type. To use RegEx in python firs
t we should import the RegEx module which is called re.

#The re Module
#After importing the module we can use it to detect or find patterns.

import re

'''Methods in re Module
To find a pattern we use different set of re character sets that allows to search for a match in a str
ing.

re.match(): searches only in the beginning of the first line of the string and returns matched objec
ts if found, else returns None.
re.search: Returns a match object if there is one anywhere in the string, including multiline string
s.
re.findall: Returns a list containing all matches
re.split: Takes a string, splits it at the match points, returns a list
re.sub: Replaces one or many matches within a string'''
#Match
# syntax
#re.match(substring, string, re.I)
#substring is a string or a pattern, string is the text we look for a pattern , re.I is case ignore

import re

txt = 'I love to teach python and javaScript'
# It returns an object with span, and match
match = re.match('I love to teach', txt, re.I)
print(match)  # <re.Match object; span=(0, 15), match='I love to teach'>
# We can get the starting and ending position of the match as tuple using span
span = match.span()
print(span)      # (0, 15)
# Lets find the start and stop position from the span
start, end = span
print(start, end)  # 0, 15
```

```
substring = txt[start:end]
print(substring)        # I love to teach
#As you can see from the example above, the pattern we are looking for (or the substring we are
looking for) is I love to teach. The match function returns an object only if the text starts with the
pattern.

import re

txt = 'I love to teach python and javaScript'
match = re.match('I like to teach', txt, re.I)
print(match)  # None
#The string does not string with I like to teach, therefore there was no match and the match meth
od returned None.

#Search
# syntax
#re.match(substring, string, re.I)
# substring is a pattern, string is the text we look for a pattern , re.I is case ignore flag

import re

txt = '''Python is the most beautiful language that a human being has ever created.
I recommend python for a first programming language'''

# It returns an object with span and match
match = re.search('first', txt, re.I)
print(match)  # <re.Match object; span=(100, 105), match='first'>
# We can get the starting and ending position of the match as tuple using span
span = match.span()
print(span)      # (100, 105)
# Lets find the start and stop position from the span
start, end = span
print(start, end)  # 100 105
substring = txt[start:end]
print(substring)        # first
#As you can see, search is much better than match because it can look for the pattern throughout
the text. Search returns a match object with a first match that was found, otherwise it returns Non
e. A much better re function is findall. This function checks for the pattern through the whole stri
ng and returns all the matches as a list.
```

```
#Searching for All Matches Using findall
#findall() returns all the matches as a list

txt = '''Python is the most beautiful language that a human being has ever created.
I recommend python for a first programming language'''

# It return a list
matches = re.findall('language', txt, re.I)
print(matches)  # ['language', 'language']
#As you can see, the word language was found two times in the string. Let us practice some mor
e. Now we will look for both Python and python words in the string:

txt = '''Python is the most beautiful language that a human being has ever created.
I recommend python for a first programming language'''

# It returns list
matches = re.findall('python', txt, re.I)
print(matches)  # ['Python', 'python']
#Since we are using re.I both lowercase and uppercase letters are included. If we do not have the
re.I flag, then we will have to write our pattern differently. Let us check it out:

txt = '''Python is the most beautiful language that a human being has ever created.
I recommend python for a first programming language'''

matches = re.findall('Python|python', txt)
print(matches)  # ['Python', 'python']

#
matches = re.findall('[Pp]ython', txt)
print(matches)  # ['Python', 'python']
#Replacing a Substring
txt = '''Python is the most beautiful language that a human being has ever created.
I recommend python for a first programming language'''

match_replaced = re.sub('Python|python', 'JavaScript', txt, re.I)
print(match_replaced)  # JavaScript is the most beautiful language that a human being has ever c
reated.
# OR
match_replaced = re.sub('[Pp]ython', 'JavaScript', txt, re.I)
```

```
print(match_replaced)  # JavaScript is the most beautiful language that a human being has ever c
reated.
#Let us add one more example. The following string is really hard to read unless we remove the
% symbol. Replacing the % with an empty string will clean the text.

txt = '''%I a%m te%%a%%che%r% a%n%d %% I l%o%ve te%ach%ing.
T%he%re i%s n%o%th%ing as r%ewarding a%s e%duc%at%i%ng a%n%d e%m%p%ow%er%i
ng p%e%o%ple.
I fo%und te%a%ching m%ore i%n%t%er%%es%ting t%h%an any other %jobs.
D%o%es thi%s m%ot%iv%a%te %y%o%u to b%e a t%e%a%cher?'''

matches = re.sub('%', '', txt)
print(matches)
#I am teacher and I love teaching.
#There is nothing as rewarding as educating and empowering people.
#I found teaching more interesting than any other jobs. Does this motivate you to be a teacher?

#Splitting Text Using RegEx Split
txt = '''I am teacher and  I love teaching.
There is nothing as rewarding as educating and empowering people.
I found teaching more interesting than any other jobs.
Does this motivate you to be a teacher?'''
print(re.split('\n', txt)) # splitting using \n - end of line symbol
['I am teacher and  I love teaching.', 'There is nothing as rewarding as educating and empowering
people.', 'I found teaching more interesting than any other jobs.', 'Does this motivate you to be a
teacher?']

#Writing RegEx Patterns
#To declare a string variable we use a single or double quote. To declare RegEx variable r''. The
following pattern only identifies apple with lowercase, to make it case insensitive either we shoul
d rewrite our pattern or we should add a flag.

import re

regex_pattern = r'apple'
txt = 'Apple and banana are fruits. An old cliche says an apple a day a doctor way has been repla
ced by a banana a day keeps the doctor far far away. '
matches = re.findall(regex_pattern, txt)
print(matches)  # ['apple']
```

```
# To make case insensitive adding flag '
matches = re.findall(regex_pattern, txt, re.I)
print(matches)  # ['Apple', 'apple']
# or we can use a set of characters method
regex_pattern = r'[Aa]pple'  # this mean the first letter could be Apple or apple
matches = re.findall(regex_pattern, txt)
print(matches)  # ['Apple', 'apple']
'''
[]: A set of characters
[a-c] means, a or b or c
[a-z] means, any letter from a to z
[A-Z] means, any character from A to Z
[0-3] means, 0 or 1 or 2 or 3
[0-9] means any number from 0 to 9
[A-Za-z0-9] any single character, that is a to z, A to Z or 0 to 9
\: uses to escape special characters
\d means: match where the string contains digits (numbers from 0-9)
\D means: match where the string does not contain digits
. : any character except new line character(\n)
^: starts with
r'^substring' eg r'^love', a sentence that starts with a word love
r'[^abc] means not a, not b, not c.
$: ends with
r'substring$' eg r'love$', sentence that ends with a word love
*: zero or more times
r'[a]*' means a optional or it can occur many times.
+: one or more times
r'[a]+' means at least once (or more)
?: zero or one time
r'[a]?' means zero times or once
{3}: Exactly 3 characters
{3,}: At least 3 characters
{3,8}: 3 to 8 characters
|: Either or
r'apple|banana' means either apple or a banana
(): Capture and group'''

#Regular Expression cheat sheet

#Let us use examples to clarify the meta characters above
```

```
#Square Bracket - Let us use square bracket to include lower and upper case

regex_pattern = r'[Aa]pple' # this square bracket mean either A or a
txt = 'Apple and banana are fruits. An old cliche says an apple a day a doctor way has been repla
ced by a banana a day keeps the doctor far far away.'
matches = re.findall(regex_pattern, txt)
print(matches)  # ['Apple', 'apple']
#If we want to look for the banana, we write the pattern as follows:

regex_pattern = r'[Aa]pple|[Bb]anana' # this square bracket means either A or a
txt = 'Apple and banana are fruits. An old cliche says an apple a day a doctor way has been repla
ced by a banana a day keeps the doctor far far away.'
matches = re.findall(regex_pattern, txt)
print(matches)  # ['Apple', 'banana', 'apple', 'banana']
#Using the square bracket and or operator , we manage to extract Apple, apple, Banana and bana
na.

#Escape character(\) in RegEx
regex_pattern = r'\d'  # d is a special character which means digits
txt = 'This regular expression example was made on December 6,  2019 and revised on July 8, 20
21'
matches = re.findall(regex_pattern, txt)
print(matches)  # ['6', '2', '0', '1', '9', '8', '2', '0', '2', '1'], this is not what we want
#One or more times(+)
regex_pattern = r'\d+'  # d is a special character which means digits, + mean one or more times
txt = 'This regular expression example was made on December 6,  2019 and revised on July 8, 20
21'
matches = re.findall(regex_pattern, txt)
print(matches)  # ['6', '2019', '8', '2021'] - now, this is better!
#Period(.)
regex_pattern = r'[a].'  # this square bracket means a and . means any character except new line
txt = '''Apple and banana are fruits'''
matches = re.findall(regex_pattern, txt)
print(matches)  # ['an', 'an', 'an', 'a ', 'ar']

regex_pattern = r'[a].+'  # . any character, + any character one or more times
matches = re.findall(regex_pattern, txt)
print(matches)  # ['and banana are fruits']
#Zero or more times(*)
```

```
#Zero or many times. The pattern could may not occur or it can occur many times.

regex_pattern = r'[a].*'  # . any character, * any character zero or more times
txt = '''Apple and banana are fruits'''
matches = re.findall(regex_pattern, txt)
print(matches)  # ['and banana are fruits']
#Zero or one time(?)
#Zero or one time. The pattern may not occur or it may occur once.

txt = '''I am not sure if there is a convention how to write the word e-mail.
Some people write it as email others may write it as Email or E-mail.'''
regex_pattern = r'[Ee]-?mail'  # ? means here that '-' is optional
matches = re.findall(regex_pattern, txt)
print(matches)  # ['e-mail', 'email', 'Email', 'E-mail']
#Quantifier in RegEx
#We can specify the length of the substring we are looking for in a text, using a curly bracket. Le
t us imagine, we are interested in a substring with a length of 4 characters:

txt = 'This regular expression example was made on December 6,  2019 and revised on July 8, 20
21'
regex_pattern = r'\d{4}'  # exactly four times
matches = re.findall(regex_pattern, txt)
print(matches)  # ['2019', '2021']

txt = 'This regular expression example was made on December 6,  2019 and revised on July 8, 20
21'
regex_pattern = r'\d{1, 4}'   # 1 to 4
matches = re.findall(regex_pattern, txt)
print(matches)  # ['6', '2019', '8', '2021']
#Cart ^
#Starts with
txt = 'This regular expression example was made on December 6,  2019 and revised on July 8, 20
21'
regex_pattern = r'^This'  # ^ means starts with
matches = re.findall(regex_pattern, txt)
print(matches)  # ['This']
#Negation
txt = 'This regular expression example was made on December 6,  2019 and revised on July 8, 20
21'
```

```
regex_pattern = r'[^A-Za-
z ]+'  # ^ in set character means negation, not A to Z, not a to z, no space
matches = re.findall(regex_pattern, txt)
print(matches)  # ['6,', '2019', '8', '2021']
```

Output

```
<re.Match object; span=(0, 15), match='I love to teach'>
(0, 15)
0 15
I love to teach
None
<re.Match object; span=(100, 105), match='first'>
(100, 105)
100 105
first
['language', 'language']
['Python', 'python']
['Python', 'python']
['Python', 'python']
JavaScript is the most beautiful language that a human being has ever created.
I recommend JavaScript for a first programming language
JavaScript is the most beautiful language that a human being has ever created.
I recommend JavaScript for a first programming language
I am teacher and  I love teaching.
There is nothing as rewarding as educating and empowering people.
I found teaching more interesting than any other jobs.
Does this motivate you to be a teacher?
['I am teacher and  I love teaching.', 'There is nothing as rewarding as educating and empowering
people.', 'I found teaching more interesting than any other jobs.', 'Does this motivate you to be a
teacher?']
['apple']
['Apple', 'apple']
['Apple', 'apple']
['Apple', 'apple']
['Apple', 'banana', 'apple', 'banana']
['6', '2', '0', '1', '9', '8', '2', '0', '2', '1']
['6', '2019', '8', '2021']
['an', 'an', 'an', 'a ', 'ar']
['and banana are fruits']
['and banana are fruits']
['e-mail', 'email', 'Email', 'E-mail']
['2019', '2021']
[]
['This']
['6,', '2019', '8,', '2021']
```

4. Design a relational database for a small application and populate the database. Using SQL do the CRUD (create, read, update and delete) operations.

A. Create Database

Mysqlcreate.py

```
import mysql.connector
mydb = mysql.connector.connect(
  host="localhost",
  user="root",
  password="manager"
)
print(mydb)
mycursor = mydb.cursor()
mycursor.execute("CREATE DATABASE hsdatabase")
mycursor.execute("SHOW DATABASES")
for x in mycursor:
  print(x)
```

Output

```
<mysql.connector.connection_cext.CMySQLConnection object at 0x0000021342D67ED0>
('hsdatabase',)
('information_schema',)
('mysql',)
('performance_schema',)
('sakila',)
('sys',)
('testdb',)
('world',)
```

B. Create Table

mysqlcreatetable.py

```
import mysql.connector
mydb = mysql.connector.connect(
    host="localhost",
    user="root",
    password="manager",
    database="hsdatabase"
)
mycursor = mydb.cursor()
#mycursor.execute("CREATE TABLE customers (id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(255), address VARCHAR(255))")
mycursor.execute("CREATE TABLE customers (name VARCHAR(255), address VARCHAR(255))")
#mycursor.execute("ALTER TABLE customers ADD COLUMN id INT AUTO_INCREMENT PRIMARY KEY")
mycursor.execute("SHOW TABLES")
for x in mycursor:
  print(x)
```

Output

```
('customers',)
```

C. Insert Values in the Table

mysqlinsert.py

```
import mysql.connector
mydb = mysql.connector.connect(
    host="localhost",
    user="root",
    password="manager",
    database="hsdatabase"
)
mycursor = mydb.cursor()
sql = "INSERT INTO customers (name, address) VALUES (%s, %s)"
val = ("Arjun", "Highway 21")
mycursor.execute(sql, val)
mydb.commit()
print(mycursor.rowcount, "record inserted it's id is.",mycursor.lastrowid)
###
sql = "INSERT INTO customers (name, address) VALUES (%s, %s)"
val = [
  ('Arjuarao', 'Lowstreet 4'),
  ('Amy', 'Apple st 652'),
  ('Hannah', 'Mountain 21'),
  ('Michael', 'Valley 345'),
  ('Sandy', 'Ocean blvd 2'),
  ('Betty', 'Green Grass 1'),
  ('Richard', 'Sky st 331'),
  ('Susan', 'One way 98'),
  ('Vicky', 'Yellow Garden 2'),
  ('Ben', 'Park Lane 38'),
  ('William', 'Central st 954'),
  ('Chuck', 'Main Road 989'),
  ('Viola', 'Sideway 1633')
]
mycursor.executemany(sql, val)
mydb.commit()
print(mycursor.rowcount, "was inserted.")
```

Output

```
1 record inserted it's id is. 0
13 was inserted.
```

D. Insert Values in the Table

mysqlselect.py

```
import mysql.connector
mydb = mysql.connector.connect(
    host="localhost",
    user="root",
    password="manager",
    database="hsdatabase"
)
mycursor = mydb.cursor()
mycursor.execute("SELECT * FROM customers")
#mycursor.execute("SELECT name, address FROM customers")
myresult = mycursor.fetchall()
#myresult = mycursor.fetchone()
for x in myresult:
  print(x)
```

Output

```
('Arjunarao', 'Highway 21')
('Peter', 'Lowstreet 4')
('Amy', 'Apple st 652')
('Hannah', 'Mountain 21')
('Michael', 'Valley 345')
('Sandy', 'Ocean blvd 2')
('Betty', 'Green Grass 1')
('Richard', 'Sky st 331')
('Susan', 'One way 98')
('Vicky', 'Yellow Garden 2')
('Ben', 'Park Lane 38')
('William', 'Central st 954')
('Chuck', 'Main Road 989')
('Viola', 'Sideway 1633')
```

E. Update Table Values

mysqlupdate.py

```
import mysql.connector
mydb = mysql.connector.connect(
    host="localhost",
    user="root",
    password="manager",
    database="hsdatabase"
)
mycursor = mydb.cursor()
sql = "SELECT * FROM customers WHERE name ='Ben'"
mycursor.execute(sql)
myresult = mycursor.fetchall()
print('Before Update')
for x in myresult:
  print(x)
#sql = "UPDATE customers SET address = 'Canyon 123' WHERE name = 'Ben'"
#mycursor.execute(sql)
sql = "UPDATE customers SET address = %s WHERE name = %s"
val = ("Valley 345", "Ben")
mycursor.execute(sql,val)
mydb.commit()
print(mycursor.rowcount, "record(s) affected")
sql = "SELECT * FROM customers WHERE name ='Ben'"
mycursor.execute(sql)
myresult = mycursor.fetchall()
print('After Update')
for x in myresult:
  print(x)
```

Output

```
Before Update
('Ben', 'Canyon 123')
1 record(s) affected
After Update
('Ben', 'Valley 345')
```

F. Delete Table Values

mysqldelete.py

```
import mysql.connector
mydb = mysql.connector.connect(
    host="localhost",
    user="root",
    password="manager",
    database="hsdatabase"
)
mycursor = mydb.cursor()
mycursor.execute("SELECT name FROM customers")
myresult = mycursor.fetchall()
print('Before Delete')
for x in myresult:
  print(x)
sql = "DELETE FROM customers WHERE name = %s"
nam = ("Ben", )
mycursor.execute(sql, nam)
mydb.commit()
print(mycursor.rowcount, "record(s) deleted")
mycursor.execute("SELECT name FROM customers")
myresult = mycursor.fetchall()
print('After Delete')
for x in myresult:
print(x)
```

Output

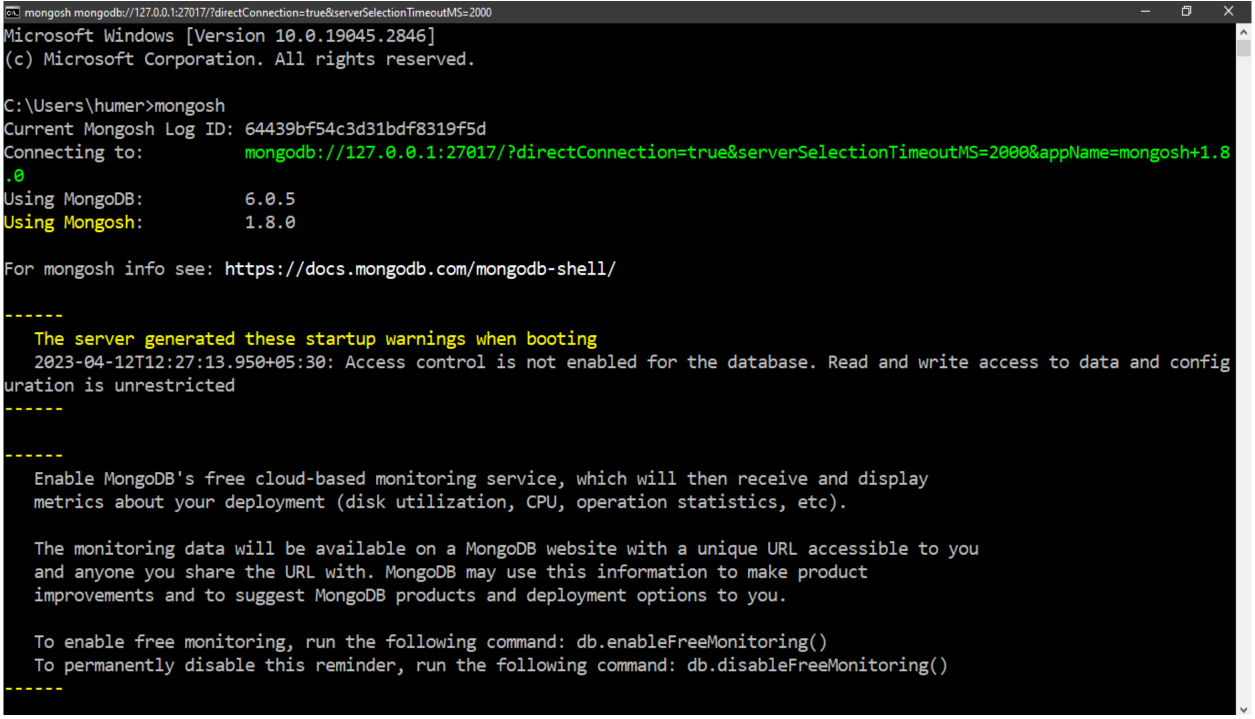
```
Before Delete
('Arjun',)
('Peter',)
('Amy',)
('Hannah',)
('Michael',)
('Sandy',)
('Betty',)
('Richard',)
('Susan',)
('Vicky',)
('Ben',)
('William',)
('Chuck',)
```

```
('Viola',)
1 record(s) deleted
After Delete
('Arjun',)
('Peter',)
('Amy',)
('Hannah',)
('Michael',)
('Sandy',)
('Betty',)
('Richard',)
('Susan',)
('Vicky',)
('William',)
('Chuck',)
('Viola',)
```

MongoDB - How to Create a Demo Database on MongoDB?

Now that you've successfully installed MongoDB and mongoose shell in your system, here's a step-by-step guide to creating a demo database to check if the database is appropriately working.

Step 1: Open the command prompt on your desktop, and type `mongosh` to start the MongoDB server. The screen will look like this.


mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

```
Microsoft Windows [Version 10.0.19045.2846]
(c) Microsoft Corporation. All rights reserved.

C:\Users\humer>mongosh
Current Mongosh Log ID: 64439bf54c3d31bdf8319f5d
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1.8
.0
Using MongoDB:          6.0.5
Using Mongosh:          1.8.0

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

------
   The server generated these startup warnings when booting
   2023-04-12T12:27:13.950+05:30: Access control is not enabled for the database. Read and write access to data and config
uration is unrestricted
------

------
   Enable MongoDB's free cloud-based monitoring service, which will then receive and display
   metrics about your deployment (disk utilization, CPU, operation statistics, etc).

   The monitoring data will be available on a MongoDB website with a unique URL accessible to you
   and anyone you share the URL with. MongoDB may use this information to make product
   improvements and to suggest MongoDB products and deployment options to you.

   To enable free monitoring, run the following command: db.enableFreeMonitoring()
   To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
------
```

Step 2: Type `show dbs` to see the existing databases.

Step 3: To create a new database, just type `use ` For example, `use Knowledgehut` and hit enter.

Step 4: To create a collection, you can type the following command. `db.createCollection(“”)`. For example, we can create a collection of students by typing `db.createCollection(“students”)`

Step 5: Then, you can insert new data by typing `db.students.insertOne({"name":" Arjun", "title":" Doe"})`.

Step 6: Now you can type `show dbs` to see all the databases, including the new one you just created.

5. Create a Python MongoDB client using the Python module pymongo. Using a collection object practice functions for inserting, searching, removing, updating, replacing, and aggregating documents, as well as for creating indexes

A. Create Database

createmongo.py

```
#Creating a Database
import pymongo

#To create a database in MongoDB, start by creating a MongoClient object, then specify a
connection URL with the correct ip address and the name of the database you want to create.
#MongoDB will create the database if it does not exist, and make a connection to it.
myclient = pymongo.MongoClient("mongodb://localhost:27017/")

#Create a database called "mydatabase"
mydb = myclient["mydatabase"]
#MongoDB waits until you have created a collection (table), with at least one document (record)
before it actually creates the database (and collection).

#Return a list of your system's databases
print(myclient.list_database_names())

#Or you can check a specific database by name
#Check if "mydatabase" exists
dblist = myclient.list_database_names()
if "mydatabase" in dblist:
  print("The database exists.")
```

Output

```
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\venv\Scripts\python.exe
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\mongocreate.py

['MongoFirst', 'admin', 'config', 'local', 'mca']

Process finished with exit code 0
```

B. Create Collection

mongocreatecollection.py

```
#Creating a Collection
import pymongo

#To create a collection in MongoDB, use database object and specify the name of the collection
you want to create.
#MongoDB will create the collection if it does not exist.
myclient = pymongo.MongoClient("mongodb://localhost:27017/")
mydb = myclient["mydatabase"]

mycol = mydb["customers"]
#In MongoDB, a collection is not created until it gets content, so if this is your first time creating
a collection, you should complete the next chapter (create document) before you check if the
collection exists!
#Return a list of all collections in your database
print(mydb.list_collection_names())

#Check if the "customers" collection exists
collist = mydb.list_collection_names()
if "customers" in collist:
  print("The collection exists.")
```

Output

```
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\venv\Scripts\python.exe
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\mongocreatecollection.py

[]

Process finished with exit code 0
```

C. Inserting documents

mongoinsertdocument.py

```
#Python MongoDB Insert Document
#A document in MongoDB is the same as a record in SQL databases.
#To insert a record, or document as it is called in MongoDB, into a collection, we use the insert_one() method.
#The first parameter of the insert_one() method is a dictionary containing the name(s) and value(s) of each field in the document you want to insert.
import pymongo
myclient = pymongo.MongoClient("mongodb://localhost:27017/")
mydb = myclient["mydatabase"]
mycol = mydb["customers"]

#insert single document/record
mydict = { "name": "Arjun", "address": "Highway 37" }
print("insert_one")
x = mycol.insert_one(mydict)

#The insert_one() method returns a InsertOneResult object, which has a property, inserted_id, that holds the id of the inserted document.
print(x.inserted_id)

#Insert Multiple Documents
#To insert multiple documents into a collection in MongoDB, we use the insert_many() method.
#The first parameter of the insert_many() method is a list containing dictionaries with the data you want to insert
mylist = [
  { "name": "Amy", "address": "Apple st 652"},
  { "name": "Hannah", "address": "Mountain 21"},
  { "name": "Michael", "address": "Valley 345"},
  { "name": "Sandy", "address": "Ocean blvd 2"},
  { "name": "Betty", "address": "Green Grass 1"},
  { "name": "Richard", "address": "Sky st 331"},
  { "name": "Susan", "address": "One way 98"},
  { "name": "Vicky", "address": "Yellow Garden 2"},
  { "name": "Ben", "address": "Park Lane 38"},
  { "name": "William", "address": "Central st 954"},
  { "name": "Chuck", "address": "Main Road 989"},
  { "name": "Viola", "address": "Sideway 1633"}
]
print("insert_many with default IDs")
x = mycol.insert_many(mylist)

#print list of the _id values of the inserted documents:
print(x.inserted_ids)

#Insert Multiple Documents, with Specified IDs
#If you do not want MongoDB to assign unique ids for you document, you can specify the _id
field when you insert the document(s).
#Remember that the values has to be unique. Two documents cannot have the same _id.
mylist = [
  { "_id": 1, "name": "Arjun", "address": "Highway 37"},
  { "_id": 2, "name": "Peter", "address": "Lowstreet 27"},
  { "_id": 3, "name": "Amy", "address": "Apple st 652"},
  { "_id": 4, "name": "Hannah", "address": "Mountain 21"},
  { "_id": 5, "name": "Michael", "address": "Valley 345"},
  { "_id": 6, "name": "Sandy", "address": "Ocean blvd 2"},
  { "_id": 7, "name": "Betty", "address": "Green Grass 1"},
  { "_id": 8, "name": "Richard", "address": "Sky st 331"},
  { "_id": 9, "name": "Susan", "address": "One way 98"},
  { "_id": 10, "name": "Vicky", "address": "Yellow Garden 2"},
  { "_id": 11, "name": "Ben", "address": "Park Lane 38"},
  { "_id": 12, "name": "William", "address": "Central st 954"},
  { "_id": 13, "name": "Chuck", "address": "Main Road 989"},
  { "_id": 14, "name": "Viola", "address": "Sideway 1633"}
]
print("insert_many with Specified IDs")
x = mycol.insert_many(mylist)

#print list of the _id values of the inserted documents:
print(x.inserted_ids)
```

Output

```
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\venv\Scripts\python.exe
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\mongoinsertdocument.py

insert_one

6443dd82c52e7b229cc86124

insert_many with default IDs
```

```
[ObjectId('6443dd82c52e7b229cc86125'), ObjectId('6443dd82c52e7b229cc86126'),
ObjectId('6443dd82c52e7b229cc86127'), ObjectId('6443dd82c52e7b229cc86128'),
ObjectId('6443dd82c52e7b229cc86129'), ObjectId('6443dd82c52e7b229cc8612a'),
ObjectId('6443dd82c52e7b229cc8612b'), ObjectId('6443dd82c52e7b229cc8612c'),
ObjectId('6443dd82c52e7b229cc8612d'), ObjectId('6443dd82c52e7b229cc8612e'),
ObjectId('6443dd82c52e7b229cc8612f'), ObjectId('6443dd82c52e7b229cc86130')]

insert_many with Specified IDs

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]

Process finished with exit code 0
```

D. Updating documents

mongoupdate.py

```
#Python MongoDB Update
#Update Collection - You can update a record, or document as it is called in MongoDB, by using the update_one() method.
#The first parameter of the update_one() method is a query object defining which document to update.
#Note: If the query finds more than one record, only the first occurrence is updated.
#The second parameter is an object defining the new values of the document.

#Change the address from "Valley 345" to "Canyon 123":

import pymongo

myclient = pymongo.MongoClient("mongodb://localhost:27017/")
mydb = myclient["mydatabase"]
mycol = mydb["customers"]

myquery = { "address": "Valley 345" }
newvalues = { "$set": { "address": "Canyon 123" } }

mycol.update_one(myquery, newvalues)
print("Updating single document")
#print "customers" after the update:
for x in mycol.find():
  print(x)

#Update Many - To update all documents that meets the criteria of the query, use the update_many() method.
#Update all documents where the address starts with the letter "S":

myquery = { "address": { "$regex": "^S" } }
newvalues = { "$set": { "name": "Minnie" } }

x = mycol.update_many(myquery, newvalues)
print("Updating multiple documents")
print(x.modified_count, "documents updated.")
```

Output

```
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\venv\Scripts\python.exe
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\mongoupdate.py

Updating single document
{'_id': ObjectId('6443dd82c52e7b229cc86124'), 'name': 'Arjun', 'address': 'Highway 37'}
{'_id': ObjectId('6443dd82c52e7b229cc86125'), 'name': 'Amy', 'address': 'Apple st 652'}
{'_id': ObjectId('6443dd82c52e7b229cc86126'), 'name': 'Hannah', 'address': 'Mountain 21'}
{'_id': ObjectId('6443dd82c52e7b229cc86127'), 'name': 'Michael', 'address': 'Canyon 123'}
{'_id': ObjectId('6443dd82c52e7b229cc86128'), 'name': 'Sandy', 'address': 'Ocean blvd 2'}
{'_id': ObjectId('6443dd82c52e7b229cc86129'), 'name': 'Betty', 'address': 'Green Grass 1'}
{'_id': ObjectId('6443dd82c52e7b229cc8612a'), 'name': 'Richard', 'address': 'Sky st 331'}
{'_id': ObjectId('6443dd82c52e7b229cc8612b'), 'name': 'Susan', 'address': 'One way 98'}
{'_id': ObjectId('6443dd82c52e7b229cc8612c'), 'name': 'Vicky', 'address': 'Yellow Garden 2'}
{'_id': ObjectId('6443dd82c52e7b229cc8612d'), 'name': 'Ben', 'address': 'Park Lane 38'}
{'_id': ObjectId('6443dd82c52e7b229cc8612e'), 'name': 'William', 'address': 'Central st 954'}
{'_id': ObjectId('6443dd82c52e7b229cc8612f'), 'name': 'Chuck', 'address': 'Main Road 989'}
{'_id': ObjectId('6443dd82c52e7b229cc86130'), 'name': 'Viola', 'address': 'Sideway 1633'}
{'_id': 1, 'name': 'Arjun', 'address': 'Highway 37'}
{'_id': 2, 'name': 'Peter', 'address': 'Lowstreet 27'}
{'_id': 3, 'name': 'Amy', 'address': 'Apple st 652'}
{'_id': 4, 'name': 'Hannah', 'address': 'Mountain 21'}
{'_id': 5, 'name': 'Michael', 'address': 'Valley 345'}
{'_id': 6, 'name': 'Sandy', 'address': 'Ocean blvd 2'}
{'_id': 7, 'name': 'Betty', 'address': 'Green Grass 1'}
{'_id': 8, 'name': 'Richard', 'address': 'Sky st 331'}
{'_id': 9, 'name': 'Susan', 'address': 'One way 98'}
{'_id': 10, 'name': 'Vicky', 'address': 'Yellow Garden 2'}
{'_id': 11, 'name': 'Ben', 'address': 'Park Lane 38'}
{'_id': 12, 'name': 'William', 'address': 'Central st 954'}
{'_id': 13, 'name': 'Chuck', 'address': 'Main Road 989'}
{'_id': 14, 'name': 'Viola', 'address': 'Sideway 1633'}
Updating multiple documents
4 documents updated.
Process finished with exit code 0
```

E. Finding documents

mongofind.py

```
#Python MongoDB Find
#In MongoDB we use the find() and find_one() methods to find data in a collection.
#Just like the SELECT statement is used to find data in a table in a MySQL database.

#Find One - To select data from a collection in MongoDB, we can use the find_one() method.
The find_one() method returns the first occurrence in the selection.
import pymongo

myclient = pymongo.MongoClient("mongodb://localhost:27017/")
mydb = myclient["mydatabase"]
mycol = mydb["customers"]
print("Find_one")
x = mycol.find_one()
print(x)

#Find All - To select data from a table in MongoDB, we can also use the find() method.
#The find() method returns all occurrences in the selection.
#The first parameter of the find() method is a query object. In this example we use an empty
query object, which selects all documents in the collection.
print("Find ")
for x in mycol.find():
  print(x)

#Return Only Some Fields - The second parameter of the find() method is an object describing
which fields to include in the result.
#This parameter is optional, and if omitted, all fields will be included in the result.
print("Find with parameters")
for x in mycol.find({},{ "_id": 0, "name": 1, "address": 1 }):
  print(x)

#You are not allowed to specify both 0 and 1 values in the same object (except if one of the
fields is the _id field). If you specify a field with the value 0, all other fields get the value 1, and
vice versa
print("Find with address parameters")
for x in mycol.find({},{ "address": 0 }):
print(x)
```

```
#You get an error if you specify both 0 and 1 values in the same object (except if one of the
fields is the _id field)
print("find with error")
for x in mycol.find({},{ "name": 1, "address": 0 }):
  print(x)
```

Output

```
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\venv\Scripts\python.exe
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\mongofind.py

Find_one
{'_id': ObjectId('6443dd82c52e7b229cc86124'), 'name': 'Arjun', 'address': 'Highway 37'}
Find
{'_id': ObjectId('6443dd82c52e7b229cc86124'), 'name': 'Arjun', 'address': 'Highway 37'}
{'_id': ObjectId('6443dd82c52e7b229cc86125'), 'name': 'Amy', 'address': 'Apple st 652'}
{'_id': ObjectId('6443dd82c52e7b229cc86126'), 'name': 'Hannah', 'address': 'Mountain 21'}
{'_id': ObjectId('6443dd82c52e7b229cc86127'), 'name': 'Michael', 'address': 'Canyon 123'}
{'_id': ObjectId('6443dd82c52e7b229cc86128'), 'name': 'Sandy', 'address': 'Ocean blvd 2'}
{'_id': ObjectId('6443dd82c52e7b229cc86129'), 'name': 'Betty', 'address': 'Green Grass 1'}
{'_id': ObjectId('6443dd82c52e7b229cc8612a'), 'name': 'Minnie', 'address': 'Sky st 331'}
{'_id': ObjectId('6443dd82c52e7b229cc8612b'), 'name': 'Susan', 'address': 'One way 98'}
{'_id': ObjectId('6443dd82c52e7b229cc8612c'), 'name': 'Vicky', 'address': 'Yellow Garden 2'}
{'_id': ObjectId('6443dd82c52e7b229cc8612d'), 'name': 'Ben', 'address': 'Park Lane 38'}
{'_id': ObjectId('6443dd82c52e7b229cc8612e'), 'name': 'William', 'address': 'Central st 954'}
{'_id': ObjectId('6443dd82c52e7b229cc8612f'), 'name': 'Chuck', 'address': 'Main Road 989'}
{'_id': ObjectId('6443dd82c52e7b229cc86130'), 'name': 'Minnie', 'address': 'Sideway 1633'}
{'_id': 1, 'name': 'Arjun', 'address': 'Highway 37'}
{'_id': 2, 'name': 'Peter', 'address': 'Lowstreet 27'}
{'_id': 3, 'name': 'Amy', 'address': 'Apple st 652'}
{'_id': 4, 'name': 'Hannah', 'address': 'Mountain 21'}
{'_id': 5, 'name': 'Michael', 'address': 'Valley 345'}
{'_id': 6, 'name': 'Sandy', 'address': 'Ocean blvd 2'}
{'_id': 7, 'name': 'Betty', 'address': 'Green Grass 1'}
{'_id': 8, 'name': 'Minnie', 'address': 'Sky st 331'}
{'_id': 9, 'name': 'Susan', 'address': 'One way 98'}
{'_id': 10, 'name': 'Vicky', 'address': 'Yellow Garden 2'}
{'_id': 11, 'name': 'Ben', 'address': 'Park Lane 38'}
{'_id': 12, 'name': 'William', 'address': 'Central st 954'}
{'_id': 13, 'name': 'Chuck', 'address': 'Main Road 989'}
{'_id': 14, 'name': 'Minnie', 'address': 'Sideway 1633'}
Find with parameters
```

```
{'name': 'Arjun', 'address': 'Highway 37'}
{'name': 'Amy', 'address': 'Apple st 652'}
{'name': 'Hannah', 'address': 'Mountain 21'}
{'name': 'Michael', 'address': 'Canyon 123'}
{'name': 'Sandy', 'address': 'Ocean blvd 2'}
{'name': 'Betty', 'address': 'Green Grass 1'}
{'name': 'Minnie', 'address': 'Sky st 331'}
{'name': 'Susan', 'address': 'One way 98'}
{'name': 'Vicky', 'address': 'Yellow Garden 2'}
{'name': 'Ben', 'address': 'Park Lane 38'}
{'name': 'William', 'address': 'Central st 954'}
{'name': 'Chuck', 'address': 'Main Road 989'}
{'name': 'Minnie', 'address': 'Sideway 1633'}
{'name': 'Arjun', 'address': 'Highway 37'}
{'name': 'Peter', 'address': 'Lowstreet 27'}
{'name': 'Amy', 'address': 'Apple st 652'}
{'name': 'Hannah', 'address': 'Mountain 21'}
{'name': 'Michael', 'address': 'Valley 345'}
{'name': 'Sandy', 'address': 'Ocean blvd 2'}
{'name': 'Betty', 'address': 'Green Grass 1'}
{'name': 'Minnie', 'address': 'Sky st 331'}
{'name': 'Susan', 'address': 'One way 98'}
{'name': 'Vicky', 'address': 'Yellow Garden 2'}
{'name': 'Ben', 'address': 'Park Lane 38'}
{'name': 'William', 'address': 'Central st 954'}
{'name': 'Chuck', 'address': 'Main Road 989'}
{'name': 'Minnie', 'address': 'Sideway 1633'}
Find with address parameters
{'_id': ObjectId('6443dd82c52e7b229cc86124'), 'name': 'Arjun'}
{'_id': ObjectId('6443dd82c52e7b229cc86125'), 'name': 'Amy'}
{'_id': ObjectId('6443dd82c52e7b229cc86126'), 'name': 'Hannah'}
{'_id': ObjectId('6443dd82c52e7b229cc86127'), 'name': 'Michael'}
{'_id': ObjectId('6443dd82c52e7b229cc86128'), 'name': 'Sandy'}
{'_id': ObjectId('6443dd82c52e7b229cc86129'), 'name': 'Betty'}
{'_id': ObjectId('6443dd82c52e7b229cc8612a'), 'name': 'Minnie'}
{'_id': ObjectId('6443dd82c52e7b229cc8612b'), 'name': 'Susan'}
{'_id': ObjectId('6443dd82c52e7b229cc8612c'), 'name': 'Vicky'}
{'_id': ObjectId('6443dd82c52e7b229cc8612d'), 'name': 'Ben'}
{'_id': ObjectId('6443dd82c52e7b229cc8612e'), 'name': 'William'}
{'_id': ObjectId('6443dd82c52e7b229cc8612f'), 'name': 'Chuck'}
```

```
{'_id': ObjectId('6443dd82c52e7b229cc86130'), 'name': 'Minnie'}
{'_id': 1, 'name': 'Arjun'}
{'_id': 2, 'name': 'Peter'}
{'_id': 3, 'name': 'Amy'}
{'_id': 4, 'name': 'Hannah'}
{'_id': 5, 'name': 'Michael'}
{'_id': 6, 'name': 'Sandy'}
{'_id': 7, 'name': 'Betty'}
{'_id': 8, 'name': 'Minnie'}
{'_id': 9, 'name': 'Susan'}
{'_id': 10, 'name': 'Vicky'}
{'_id': 11, 'name': 'Ben'}
{'_id': 12, 'name': 'William'}
{'_id': 13, 'name': 'Chuck'}
{'_id': 14, 'name': 'Minnie'}
find with error
Traceback (most recent call last):
  File "C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\mongofind.py", line 35, in
<module>
    for x in mycol.find({},{ "name": 1, "address": 0 }):
  File "C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\venv\Lib\site-
packages\pymongo\cursor.py", line 1248, in next
    if len(self.__data) or self._refresh():
                           ^^^^^^^^^^^^^^^
  File "C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\venv\Lib\site-
packages\pymongo\cursor.py", line 1165, in _refresh
    self.__send_message(q)
  File "C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\venv\Lib\site-
packages\pymongo\cursor.py", line 1052, in __send_message
    response = client._run_operation(
               ^^^^^^^^^^^^^^^^^^^^^^
  File "C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\venv\Lib\site-
packages\pymongo\_csot.py", line 105, in csot_wrapper
    return func(self, *args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\venv\Lib\site-
packages\pymongo\mongo_client.py", line 1330, in _run_operation
    return self._retryable_read(
           ^^^^^^^^^^^^^^^^^^^^^
  File "C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\venv\Lib\site-packages\pymongo\_csot.py", line 105, in csot_wrapper
    return func(self, *args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\venv\Lib\site-packages\pymongo\mongo_client.py", line 1448, in _retryable_read
    return func(session, server, sock_info, read_pref)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\venv\Lib\site-packages\pymongo\mongo_client.py", line 1326, in _cmd
    return server.run_operation(
           ^^^^^^^^^^^^^^^^^^^^^
  File "C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\venv\Lib\site-packages\pymongo\server.py", line 134, in run_operation
    _check_command_response(first, sock_info.max_wire_version)
  File "C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\venv\Lib\site-packages\pymongo\helpers.py", line 181, in _check_command_response
    raise OperationFailure(errmsg, code, response, max_wire_version)
pymongo.errors.OperationFailure: Cannot do exclusion on field address in inclusion projection, full error: {'ok': 0.0, 'errmsg': 'Cannot do exclusion on field address in inclusion projection', 'code': 31254, 'codeName': 'Location31254'}

Process finished with exit code 1
```

F. Querying documents

mongoquery.py

```
#Python MongoDB Query
#Filter the Result - When finding documents in a collection, you can filter the result by using a query object.
#The first argument of the find() method is a query object, and is used to limit the search.
#Find document(s) with the address "Park Lane 38"
import pymongo

myclient = pymongo.MongoClient("mongodb://localhost:27017/")
mydb = myclient["mydatabase"]
mycol = mydb["customers"]

myquery = { "address": "Park Lane 38" }

mydoc = mycol.find(myquery)
print("Query with address")
for x in mydoc:
  print(x)

#Advanced Query - To make advanced queries you can use modifiers as values in the query object.
#E.g. to find the documents where the "address" field starts with the letter "S" or higher (alphabetically), use the greater than modifier: {"$gt": "S"}
#Find documents where the address starts with the letter "S" or higher

myquery = { "address": { "$gt": "S" } }
mydoc = mycol.find(myquery)
print("Advanced Query")
for x in mydoc:
  print(x)

#Filter With Regular Expressions - You can also use regular expressions as a modifier.
#Regular expressions can only be used to query strings - To find only the documents where the "address" field starts with the letter "S", use the regular expression {"$regex": "^S"}
#Find documents where the address starts with the letter "S"
myquery = { "address": { "$regex": "^S" } }

mydoc = mycol.find(myquery)
print("Filter With Regular Expressions")
```

```
for x in mydoc:
  print(x)
```

Output

```
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\venv\Scripts\python.exe
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\mongoquery.py

Query with address
{'_id': ObjectId('6443dd82c52e7b229cc8612d'), 'name': 'Ben', 'address': 'Park Lane 38'}
{'_id': 11, 'name': 'Ben', 'address': 'Park Lane 38'}
Advanced Query
{'_id': ObjectId('6443dd82c52e7b229cc8612a'), 'name': 'Minnie', 'address': 'Sky st 331'}
{'_id': ObjectId('6443dd82c52e7b229cc8612c'), 'name': 'Vicky', 'address': 'Yellow Garden 2'}
{'_id': ObjectId('6443dd82c52e7b229cc86130'), 'name': 'Minnie', 'address': 'Sideway 1633'}
{'_id': 5, 'name': 'Michael', 'address': 'Valley 345'}
{'_id': 8, 'name': 'Minnie', 'address': 'Sky st 331'}
{'_id': 10, 'name': 'Vicky', 'address': 'Yellow Garden 2'}
{'_id': 14, 'name': 'Minnie', 'address': 'Sideway 1633'}
Filter With Regular Expressions
{'_id': ObjectId('6443dd82c52e7b229cc8612a'), 'name': 'Minnie', 'address': 'Sky st 331'}
{'_id': ObjectId('6443dd82c52e7b229cc86130'), 'name': 'Minnie', 'address': 'Sideway 1633'}
{'_id': 8, 'name': 'Minnie', 'address': 'Sky st 331'}
{'_id': 14, 'name': 'Minnie', 'address': 'Sideway 1633'}

Process finished with exit code 0
```

G. Sorting documents

mongosort.py

```
#Python MongoDB Sort
#Sort the Result - Use the sort() method to sort the result in ascending or descending order.
#The sort() method takes one parameter for "fieldname" and one parameter for "direction"
(ascending is the default direction).
#Sort the result alphabetically by name
import pymongo
myclient = pymongo.MongoClient("mongodb://localhost:27017/")
mydb = myclient["mydatabase"]
mycol = mydb["customers"]
mydoc = mycol.find().sort("name")
print("Sort by name in ascending order")
for x in mydoc:
  print(x)
#Sort Descending - Use the value -1 as the second parameter to sort descending.
#sort("name", 1) #ascending and sort("name", -1) #descending
#Sort the result reverse alphabetically by name
mydoc = mycol.find().sort("name", -1)
print("Sort by name in descending order")
for x in mydoc:
  print(x)
```

Output

```
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\venv\Scripts\python.exe
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\mongosort.py

Sort by name in ascending order
{'_id': ObjectId('6443dd82c52e7b229cc86125'), 'name': 'Amy', 'address': 'Apple st 652'}
{'_id': 3, 'name': 'Amy', 'address': 'Apple st 652'}
{'_id': ObjectId('6443dd82c52e7b229cc8612d'), 'name': 'Ben', 'address': 'Park Lane 38'}
{'_id': 11, 'name': 'Ben', 'address': 'Park Lane 38'}
{'_id': ObjectId('6443dd82c52e7b229cc86129'), 'name': 'Betty', 'address': 'Green Grass 1'}
{'_id': 7, 'name': 'Betty', 'address': 'Green Grass 1'}
{'_id': ObjectId('6443dd82c52e7b229cc8612f'), 'name': 'Chuck', 'address': 'Main Road 989'}
{'_id': 13, 'name': 'Chuck', 'address': 'Main Road 989'}
{'_id': ObjectId('6443dd82c52e7b229cc86126'), 'name': 'Hannah', 'address': 'Mountain 21'}
{'_id': 4, 'name': 'Hannah', 'address': 'Mountain 21'}
{'_id': ObjectId('6443dd82c52e7b229cc86124'), 'name': 'Arjun', 'address': 'Highway 37'}
{'_id': 1, 'name': 'Arjun', 'address': 'Highway 37'}
```

```
{'_id': ObjectId('6443dd82c52e7b229cc86127'), 'name': 'Michael', 'address': 'Canyon 123'}
{'_id': 5, 'name': 'Michael', 'address': 'Valley 345'}
{'_id': ObjectId('6443dd82c52e7b229cc8612a'), 'name': 'Minnie', 'address': 'Sky st 331'}
{'_id': ObjectId('6443dd82c52e7b229cc86130'), 'name': 'Minnie', 'address': 'Sideway 1633'}
{'_id': 8, 'name': 'Minnie', 'address': 'Sky st 331'}
{'_id': 14, 'name': 'Minnie', 'address': 'Sideway 1633'}
{'_id': 2, 'name': 'Peter', 'address': 'Lowstreet 27'}
{'_id': ObjectId('6443dd82c52e7b229cc86128'), 'name': 'Sandy', 'address': 'Ocean blvd 2'}
{'_id': 6, 'name': 'Sandy', 'address': 'Ocean blvd 2'}
{'_id': ObjectId('6443dd82c52e7b229cc8612b'), 'name': 'Susan', 'address': 'One way 98'}
{'_id': 9, 'name': 'Susan', 'address': 'One way 98'}
{'_id': ObjectId('6443dd82c52e7b229cc8612c'), 'name': 'Vicky', 'address': 'Yellow Garden 2'}
{'_id': 10, 'name': 'Vicky', 'address': 'Yellow Garden 2'}
{'_id': ObjectId('6443dd82c52e7b229cc8612e'), 'name': 'William', 'address': 'Central st 954'}
{'_id': 12, 'name': 'William', 'address': 'Central st 954'}
Sort by name in descending order
{'_id': ObjectId('6443dd82c52e7b229cc8612e'), 'name': 'William', 'address': 'Central st 954'}
{'_id': 12, 'name': 'William', 'address': 'Central st 954'}
{'_id': ObjectId('6443dd82c52e7b229cc8612c'), 'name': 'Vicky', 'address': 'Yellow Garden 2'}
{'_id': 10, 'name': 'Vicky', 'address': 'Yellow Garden 2'}
{'_id': ObjectId('6443dd82c52e7b229cc8612b'), 'name': 'Susan', 'address': 'One way 98'}
{'_id': 9, 'name': 'Susan', 'address': 'One way 98'}
{'_id': ObjectId('6443dd82c52e7b229cc86128'), 'name': 'Sandy', 'address': 'Ocean blvd 2'}
{'_id': 6, 'name': 'Sandy', 'address': 'Ocean blvd 2'}
{'_id': 2, 'name': 'Peter', 'address': 'Lowstreet 27'}
{'_id': ObjectId('6443dd82c52e7b229cc8612a'), 'name': 'Minnie', 'address': 'Sky st 331'}
{'_id': ObjectId('6443dd82c52e7b229cc86130'), 'name': 'Minnie', 'address': 'Sideway 1633'}
{'_id': 8, 'name': 'Minnie', 'address': 'Sky st 331'}
{'_id': 14, 'name': 'Minnie', 'address': 'Sideway 1633'}
{'_id': ObjectId('6443dd82c52e7b229cc86127'), 'name': 'Michael', 'address': 'Canyon 123'}
{'_id': 5, 'name': 'Michael', 'address': 'Valley 345'}
{'_id': ObjectId('6443dd82c52e7b229cc86124'), 'name': 'Arjun', 'address': 'Highway 37'}
{'_id': 1, 'name': 'Arjun', 'address': 'Highway 37'}
{'_id': ObjectId('6443dd82c52e7b229cc86126'), 'name': 'Hannah', 'address': 'Mountain 21'}
{'_id': 4, 'name': 'Hannah', 'address': 'Mountain 21'}
{'_id': ObjectId('6443dd82c52e7b229cc8612f'), 'name': 'Chuck', 'address': 'Main Road 989'}
{'_id': 13, 'name': 'Chuck', 'address': 'Main Road 989'}
{'_id': ObjectId('6443dd82c52e7b229cc86129'), 'name': 'Betty', 'address': 'Green Grass 1'}
{'_id': 7, 'name': 'Betty', 'address': 'Green Grass 1'}
{'_id': ObjectId('6443dd82c52e7b229cc8612d'), 'name': 'Ben', 'address': 'Park Lane 38'}
```

```
{'_id': 11, 'name': 'Ben', 'address': 'Park Lane 38'}
{'_id': ObjectId('6443dd82c52e7b229cc86125'), 'name': 'Amy', 'address': 'Apple st 652'}
{'_id': 3, 'name': 'Amy', 'address': 'Apple st 652'}

Process finished with exit code 0
```

H. Deleting documents

mongodelete.py

```
#Python MongoDB Delete Document
#Delete Document - To delete one document, we use the delete_one() method.
#The first parameter of the delete_one() method is a query object defining which document to delete.
#Note: If the query finds more than one document, only the first occurrence is deleted.

#Delete the document with the address "Mountain 21"
import pymongo

myclient = pymongo.MongoClient("mongodb://localhost:27017/")
mydb = myclient["mydatabase"]
mycol = mydb["customers"]

myquery = { "address": "Mountain 21" }

mycol.delete_one(myquery)
print("Single document deleted")

#Delete Many Documents - To delete more than one document, use the delete_many() method.
#The first parameter of the delete_many() method is a query object defining which documents to delete.
#Delete all documents were the address starts with the letter S
myquery = { "address": {"$regex": "^S"} }
x = mycol.delete_many(myquery)
print("Deleting multiple documents")
print(x.deleted_count, " documents deleted.")

#Delete All Documents in a Collection - To delete all documents in a collection, pass an empty query object to the delete_many() method:
#Delete all documents in the "customers" collection
print("Delete all documents in a collection")
x = mycol.delete_many({})
print(x.deleted_count, " documents deleted.")
```

Output

```
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\venv\Scripts\python.exe
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\mongodelete.py

Single document deleted
Deleting multiple documents
4  documents deleted.
Delete all documents in a collection
22  documents deleted.

Process finished with exit code 0
```

I. Drop Collection

mongodropcollection.py

```
#Python MongoDB Drop Collection
#Delete Collection - You can delete a table, or collection as it is called in MongoDB, by using the drop() method.
#Delete the "customers" collection:

import pymongo
myclient = pymongo.MongoClient("mongodb://localhost:27017/")
mydb = myclient["mydatabase"]
mycol = mydb["customers"]

print(mycol.drop())

#The drop() method returns true if the collection was dropped successfully, and false if the collection does not exist.
```

Output

```
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\venv\Scripts\python.exe
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\mongodropcollection.py

None

Process finished with exit code 0
```

6. Write programs to create numpy arrays of different shapes and from different sources, reshape and slice arrays, add array indexes, and apply arithmetic, logic, and aggregation functions to some or all array elements

#Python for Statistical Analysis

#Statistics -

Statistics is the discipline that studies the collection, organization, displaying, analysing, interpretation and presentation of data. Statistics is a branch of Mathematics that is recommended to be a prerequisite for data science and machine learning. Statistics is a very broad field but we will focus in this section only on the most relevant part. After completing this challenge, you may go onto the web development, data analysis, machine learning and data science path. Whatever path you may follow, at some point in your career you will get data which you may work on. Having some statistical knowledge will help you to make decisions based on data, data tells as they say.

#Data -

What is data? Data is any set of characters that is gathered and translated for some purpose, usually analysis. It can be any character, including text and numbers, pictures, sound, or video. If data is not put in a context, it doesn't make any sense to a human or computer. To make sense from data we need to work on the data using different tools.

#The work flow of data analysis, data science or machine learning starts from data. Data can be provided from some data source or it can be created. There are structured and unstructured data.

#Data can be found in small or big format. Most of the data types we will get have been covered in the file handling section.

#NumPy - In the first section we defined Python as a great general-purpose programming language on its own, but with the help of other popular libraries as(numpy, scipy, matplotlib, pandas etc) it becomes a powerful environment for scientific computing.

#NumPy is the core library for scientific computing in Python. It provides a high-performance multidimensional array object, and tools for working with arrays.

```
# How to import numpy
import numpy as np
# How to check the version of the numpy package
print('numpy:', np.__version__)
# Checking the available methods
print(dir(np))
#Creating numpy array using
#Creating int numpy arrays
# Creating python List
python_list = [1,2,3,4,5]
```

```
# Checking data types
print('Type:', type (python_list)) # <class 'list'>
print(python_list) # [1, 2, 3, 4, 5]
two_dimensional_list = [[0,1,2], [3,4,5], [6,7,8]]
print(two_dimensional_list)  # [[0, 1, 2], [3, 4, 5], [6, 7, 8]]
# Creating Numpy(Numerical Python) array from python list

numpy_array_from_list = np.array(python_list)
print(type (numpy_array_from_list))  # <class 'numpy.ndarray'>
print(numpy_array_from_list) # array([1, 2, 3, 4, 5])
#Creating float numpy arrays
#Creating a float numpy array from list with a float data type parameter

# Python list
python_list = [1,2,3,4,5]

numy_array_from_list2 = np.array(python_list, dtype=float)
print(numy_array_from_list2) # array([1., 2., 3., 4., 5.])
#Creating boolean numpy arrays
#Creating a boolean a numpy array from list

numpy_bool_array = np.array([0, 1, -1, 0, 0], dtype=bool)
print(numpy_bool_array) # array([False,  True,  True, False, False])
#Creating multidimensional array using numpy
#A numpy array may have one or multiple rows and columns

two_dimensional_list = [[0,1,2], [3,4,5], [6,7,8]]
numpy_two_dimensional_list = np.array(two_dimensional_list)
print(type (numpy_two_dimensional_list))
print(numpy_two_dimensional_list)

#Converting numpy array to list
# We can always convert an array back to a python list using tolist().
np_to_list = numpy_array_from_list.tolist()
print(type (np_to_list))
print('one dimensional array:', np_to_list)
print('two dimensional array: ', numpy_two_dimensional_list.tolist())

#Creating numpy array from tuple
# Numpy array from tuple
# Creating tuple in Python
python_tuple = (1,2,3,4,5)
print(type (python_tuple)) # <class 'tuple'>
print('python_tuple: ', python_tuple) # python_tuple:  (1, 2, 3, 4, 5)

numpy_array_from_tuple = np.array(python_tuple)
print(type (numpy_array_from_tuple)) # <class 'numpy.ndarray'>
print('numpy_array_from_tuple: ', numpy_array_from_tuple) # numpy_array_from_tuple:  [1 2 3
 4 5]
#Shape of numpy array
#The shape method provide the shape of the array as a tuple. The first is the row and the second i
s the column. If the array is just one dimensional it returns the size of the array.

nums = np.array([1, 2, 3, 4, 5])
print(nums)
print('shape of nums: ', nums.shape)
print(numpy_two_dimensional_list)
print('shape of numpy_two_dimensional_list: ', numpy_two_dimensional_list.shape)
three_by_four_array = np.array([[0, 1, 2, 3],
        [4,5,6,7],
        [8,9,10, 11]])
print(three_by_four_array.shape)

#Data type of numpy array
#Type of data types: str, int, float, complex, bool, list, None

int_lists = [-3, -2, -1, 0, 1, 2,3]
int_array = np.array(int_lists)
float_array = np.array(int_lists, dtype=float)

print(int_array)
print(int_array.dtype)
print(float_array)
print(float_array.dtype)

#Size of a numpy array
#In numpy to know the number of items in a numpy array list we use size

numpy_array_from_list = np.array([1, 2, 3, 4, 5])
two_dimensional_list = np.array([[0, 1, 2],
                              [3, 4, 5],
                              [6, 7, 8]])

print('The size:', numpy_array_from_list.size) # 5
print('The size:', two_dimensional_list.size)  # 3

#Mathematical Operation using numpy
#NumPy array is not like exactly like python list. To do mathematical operation in Python list we
have to loop through the items but numpy can allow to do any mathematical operation without l
ooping. Mathematical Operation:

'''Addition (+)
Subtraction (-)
Multiplication (*)
Division (/)
Modules (%)
Floor Division(//)
Exponential(**)
Addition
# Mathematical Operation
# Addition'''
numpy_array_from_list = np.array([1, 2, 3, 4, 5])
print('original array: ', numpy_array_from_list)
ten_plus_original = numpy_array_from_list  + 10
print(ten_plus_original)

# Subtraction
numpy_array_from_list = np.array([1, 2, 3, 4, 5])
print('original array: ', numpy_array_from_list)
ten_minus_original = numpy_array_from_list  - 10
print(ten_minus_original)

# Multiplication
numpy_array_from_list = np.array([1, 2, 3, 4, 5])
print('original array: ', numpy_array_from_list)
ten_times_original = numpy_array_from_list * 10
print(ten_times_original)

# Division
numpy_array_from_list = np.array([1, 2, 3, 4, 5])
```

```
print('original array: ', numpy_array_from_list)
ten_times_original = numpy_array_from_list / 10
print(ten_times_original)

# Modulus; Finding the remainder
numpy_array_from_list = np.array([1, 2, 3, 4, 5])
print('original array: ', numpy_array_from_list)
ten_times_original = numpy_array_from_list % 3
print(ten_times_original)

# Floor division: the division result without the remainder
numpy_array_from_list = np.array([1, 2, 3, 4, 5])
print('original array: ', numpy_array_from_list)
ten_times_original = numpy_array_from_list // 10
print(ten_times_original)

# Exponential is finding some number the power of another:
numpy_array_from_list = np.array([1, 2, 3, 4, 5])
print('original array: ', numpy_array_from_list)
ten_times_original = numpy_array_from_list  ** 2
print(ten_times_original)

#Checking data types - Int,  Float numbers
numpy_int_arr = np.array([1,2,3,4])
numpy_float_arr = np.array([1.1, 2.0,3.2])
numpy_bool_arr = np.array([-3, -2, 0, 1,2,3], dtype='bool')

print(numpy_int_arr.dtype)
print(numpy_float_arr.dtype)
print(numpy_bool_arr.dtype)

#Converting types - We can convert the data types of numpy array

#Int to Float
numpy_int_arr = np.array([1,2,3,4], dtype = 'float')
numpy_int_arr

#Float to Int
numpy_int_arr = np.array([1., 2., 3., 4.], dtype = 'int')
numpy_int_arr
```

```
#Int ot boolean
np.array([-3, -2, 0, 1,2,3], dtype='bool')

#Int to str
#numpy_float_list.astype('int').astype('str')
#array(['1', '2', '3'], dtype='<U21')
#Multi-dimensional Arrays - 2 Dimension Array
two_dimension_array = np.array([(1,2,3),(4,5,6), (7,8,9)])
print(type (two_dimension_array))
print(two_dimension_array)
print('Shape: ', two_dimension_array.shape)
print('Size:', two_dimension_array.size)
print('Data type:', two_dimension_array.dtype)

#Getting items from a numpy array
# 2 Dimension Array
two_dimension_array = np.array([[1,2,3],[4,5,6], [7,8,9]])
first_row = two_dimension_array[0]
second_row = two_dimension_array[1]
third_row = two_dimension_array[2]
print('First row:', first_row)
print('Second row:', second_row)
print('Third row: ', third_row)

first_column= two_dimension_array[:,0]
second_column = two_dimension_array[:,1]
third_column = two_dimension_array[:,2]
print('First column:', first_column)
print('Second column:', second_column)
print('Third column: ', third_column)
print(two_dimension_array)

#Slicing Numpy array - Slicing in numpy is similar to slicing in python list

two_dimension_array = np.array([[1,2,3],[4,5,6], [7,8,9]])
first_two_rows_and_columns = two_dimension_array[0:2, 0:2]
print(first_two_rows_and_columns)

#How to reverse the rows and the whole array?
```

```
two_dimension_array[::]

#Reverse the row and column positions
two_dimension_array = np.array([[1,2,3],[4,5,6], [7,8,9]])
two_dimension_array[::-1,::-1]

#How to represent missing values ?
print(two_dimension_array)
two_dimension_array[1,1] = 55
two_dimension_array[1,2] =44
print(two_dimension_array)
# Numpy Zeroes
# numpy.zeros(shape, dtype=float, order='C')
numpy_zeroes = np.zeros((3,3),dtype=int,order='C')
numpy_zeroes

# Numpy Zeroes
numpy_ones = np.ones((3,3),dtype=int,order='C')
print(numpy_ones)

twoes = numpy_ones * 2
# Reshape
# numpy.reshape(), numpy.flatten()
first_shape  = np.array([(1,2,3), (4,5,6)])
print(first_shape)
reshaped = first_shape.reshape(3,2)
print(reshaped)

flattened = reshaped.flatten()
flattened

## Horitzontal Stack
np_list_one = np.array([1,2,3])
np_list_two = np.array([4,5,6])

print(np_list_one + np_list_two)

print('Horizontal Append:', np.hstack((np_list_one, np_list_two)))

## Vertical Stack
```

```
print('Vertical Append:', np.vstack((np_list_one, np_list_two)))

#Generating Random Numbers - Generate a random float  number
random_float = np.random.random()
random_float

# Generate a random float  number
random_floats = np.random.random(5)
random_floats

# Generating a random integers between 0 and 10

random_int = np.random.randint(0, 11)
random_int

# Generating a random integers between 2 and 11, and creating a one row array
random_int = np.random.randint(2,10, size=4)
random_int

# Generating a random integers between 0 and 10
random_int = np.random.randint(2,10, size=(3,3))
random_int

#Generationg random numbers - np.random.normal(mu, sigma, size)
normal_array = np.random.normal(79, 15, 80)
normal_array

#Numpy and Statistics
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()
plt.hist(normal_array, color="grey", bins=50)

#Matrix in numpy
four_by_four_matrix = np.matrix(np.ones((4,4), dtype=float))
four_by_four_matrix

np.asarray(four_by_four_matrix)[2] = 2
four_by_four_matrix
```

```
#Numpy numpy.arange()
#What is Arrange? Sometimes, you want to create values that are evenly spaced within a defined
interval. For instance, you want to create values from 1 to 10; you can use numpy.arange() funct
ion

# creating list using range(starting, stop, step)
lst = range(0, 11, 2)
lst
range(0, 11, 2)
for l in lst:
    print(l)

# Similar to range arange numpy.arange(start, stop, step)
whole_numbers = np.arange(0, 20, 1)
whole_numbers

natural_numbers = np.arange(1, 20, 1)
natural_numbers
odd_numbers = np.arange(1, 20, 2)
odd_numbers

even_numbers = np.arange(2, 20, 2)
even_numbers

#Creating sequence of numbers using linspace
# numpy.linspace()
# numpy.logspace() in Python with Example
# For instance, it can be used to create 10 values from 1 to 5 evenly spaced.
np.linspace(1.0, 5.0, num=10)

# not to include the last value in the interval
np.linspace(1.0, 5.0, num=5, endpoint=False)

# LogSpace
# LogSpace returns even spaced numbers on a log scale. Logspace has the same parameters as np
.linspace.
# Syntax:
# numpy.logspace(start, stop, num, endpoint)

np.logspace(2, 4.0, num=4)
```

```
# to check the size of an array
x = np.array([1,2,3], dtype=np.complex128)
x

x.itemsize
# indexing and Slicing NumPy Arrays in Python
np_list = np.array([(1,2,3), (4,5,6)])
np_list

print('First row: ', np_list[0])
print('Second row: ', np_list[1])

print('First column: ', np_list[:,0])
print('Second column: ', np_list[:,1])
print('Third column: ', np_list[:,2])

#NumPy Statistical Functions with Example -
```

NumPy has quite useful statistical functions for finding minimum, maximum, mean, median, percentile,standard deviation and variance, etc from the given elements in the array. The functions are explained as follows − Statistical function Numpy is equipped with the robust statistical function as listed below

```
#Numpy Functions
#Min - np.min()
#Max - np.max()
#Mean - np.mean()
#Median - np.median()
#Varience
#Percentile
#Standard deviation - np.std()
np_normal_dis = np.random.normal(5, 0.5, 100)
np_normal_dis
## min, max, mean, median, sd
print('min: ', two_dimension_array.min())
print('max: ', two_dimension_array.max())
print('mean: ',two_dimension_array.mean())
# print('median: ', two_dimension_array.median())
print('sd: ', two_dimension_array.std())

print(two_dimension_array)
print('Column with minimum: ', np.amin(two_dimension_array,axis=0))
print('Column with maximum: ', np.amax(two_dimension_array,axis=0))
print('=== Row ==')
print('Row with minimum: ', np.amin(two_dimension_array,axis=1))
print('Row with maximum: ', np.amax(two_dimension_array,axis=1))

#How to create repeating sequences?
a = [1,2,3]

# Repeat whole of 'a' two times
print('Tile:   ', np.tile(a, 2))

# Repeat each element of 'a' two times
print('Repeat: ', np.repeat(a, 2))

#How to generate random numbers?
# One random number between [0,1)
one_random_num = np.random.random()
one_random_in = np.random
print(one_random_num)

# Random numbers between [0,1) of shape 2,3
r = np.random.random(size=[2,3])
print(r)

print(np.random.choice(['a', 'e', 'i', 'o', 'u'], size=10))

## Random numbers between [0, 1] of shape 2, 2
rand = np.random.rand(2,2)
rand

rand2 = np.random.randn(2,2)
rand2

# Random integers between [0, 10) of shape 2,5
rand_int = np.random.randint(0, 10, size=[5,3])
rand_int

from scipy import stats
```

```
np_normal_dis = np.random.normal(5, 0.5, 1000) # mean, standard deviation, number of samples
np_normal_dis
## min, max, mean, median, sd
print('min: ', np.min(np_normal_dis))
print('max: ', np.max(np_normal_dis))
print('mean: ', np.mean(np_normal_dis))
print('median: ', np.median(np_normal_dis))
print('mode: ', stats.mode(np_normal_dis))
print('sd: ', np.std(np_normal_dis))

plt.hist(np_normal_dis, color="grey", bins=21)
plt.show()

# numpy.dot(): Dot Product in Python using Numpy
# Dot Product
# Numpy is powerful library for matrices computation. For instance, you can compute the dot product with np.dot

# Syntax
# numpy.dot(x, y, out=None)
## Linear algebra
### Dot product: product of two arrays
f = np.array([1,2,3])
g = np.array([4,5,3])
### 1*4+2*5 + 3*6
np.dot(f, g)  # 23
#NumPy Matrix Multiplication with np.matmul()
### Matmul: matruc product of two arrays
h = [[1,2],[3,4]]
i = [[5,6],[7,8]]
### 1*5+2*7 = 19
np.matmul(h, i)

## Determinant 2*2 matrix
### 5*8-7*6np.linalg.det(i)
np.linalg.det(i)

Z = np.zeros((8,8))
Z[1::2,::2] = 1
```

```
Z[::2,1::2] = 1
Z

new_list = [ x + 2 for x in range(0, 11)]
new_list

np_arr = np.array(range(0, 11))
np_arr + 2

#We use linear equation for quantities which have linear relationship. Let's see the example belo
w:

temp = np.array([1,2,3,4,5])
pressure = temp * 2 + 5
pressure

plt.plot(temp,pressure)
plt.xlabel('Temperature in oC')
plt.ylabel('Pressure in atm')
plt.title('Temperature vs Pressure')
plt.xticks(np.arange(0, 6, step=0.5))
plt.show()

#To draw the Gaussian normal distribution using numpy. As you can see below, the numpy can g
enerate random numbers. To create random sample, we need the mean(mu), sigma(standard devi
ation), mumber of data points.

mu = 28
sigma = 15
samples = 100000

x = np.random.normal(mu, sigma, samples)
ax = sns.distplot(x);
ax.set(xlabel="x", ylabel='y')
plt.show()

#Summary - To summarize, the main differences with python lists are:
#Arrays support vectorized operations, while lists don’t.
#Once an array is created, you cannot change its size. You will have to create a new array or ove
rwrite the existing one.
```

```
#Every array has one and only one dtype. All items in it should be of that dtype.
#An equivalent numpy array occupies much less space than a python list of lists.
#numpy arrays support boolean indexing.
```

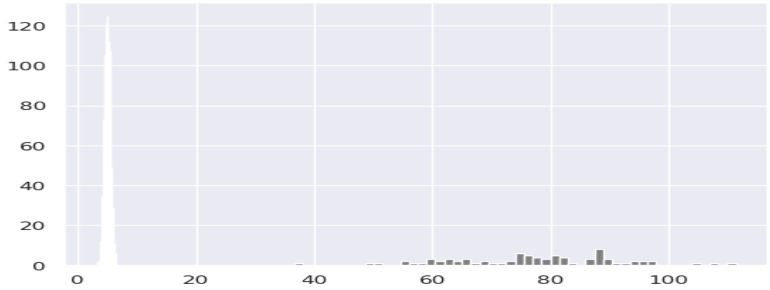
Output

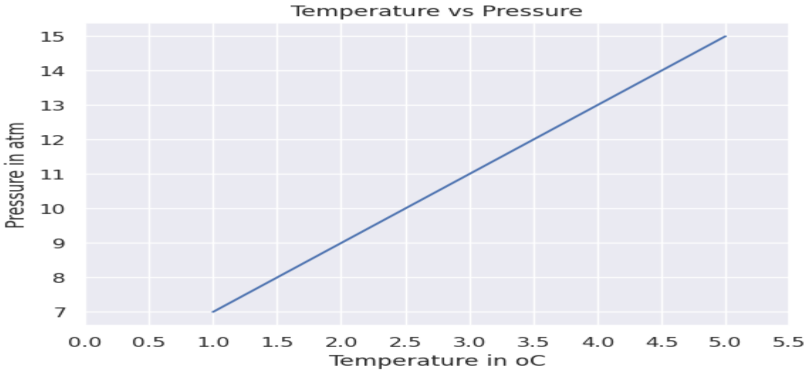
```
numpy: 1.22.4
['ALLOW_THREADS', 'AxisError', 'BUFSIZE', 'CLIP', 'ComplexWarning', 'DataSource',
'ERR_CALL', 'ERR_DEFAULT', 'ERR_IGNORE', 'ERR_LOG', 'ERR_PRINT', 'ERR_RAISE',
'ERR_WARN', 'FLOATING_POINT_SUPPORT', 'FPE_DIVIDEBYZERO', 'FPE_INVALID',
'FPE_OVERFLOW', 'FPE_UNDERFLOW', 'False_', 'Inf', 'Infinity', 'MAXDIMS',
'MAY_SHARE_BOUNDS', 'MAY_SHARE_EXACT', 'ModuleDeprecationWarning', 'NAN',
'NINF', 'NZERO', 'NaN', 'PINF', 'PZERO', 'RAISE', 'RankWarning', 'SHIFT_DIVIDEBYZERO',
'SHIFT_INVALID', 'SHIFT_OVERFLOW', 'SHIFT_UNDERFLOW', 'ScalarType', 'Tester',
'TooHardError', 'True_', 'UFUNC_BUFSIZE_DEFAULT', 'UFUNC_PYVALS_NAME',
'VisibleDeprecationWarning', 'WRAP', '_CopyMode', '_NoValue', '_UFUNC_API',
'__NUMPY_SETUP__', '__all__', '__builtins__', '__cached__', '__config__',
'__deprecated_attrs__', '__dir__', '__doc__', '__expired_functions__', '__file__', '__getattr__',
'__git_version__', '__loader__', '__name__', '__package__', '__path__', '__spec__', '__version__',
'_add_newdoc_ufunc', '_distributor_init', '_financial_names', '_from_dlpack', '_globals', '_mat',
'_pytesttester', '_version', 'abs', 'absolute', 'add', 'add_docstring', 'add_newdoc',
'add_newdoc_ufunc', 'alen', 'all', 'allclose', 'alltrue', 'amax', 'amin', 'angle', 'any', 'append',
'apply_along_axis', 'apply_over_axes', 'arange', 'arccos', 'arccosh', 'arcsin', 'arcsinh', 'arctan',
'arctan2', 'arctanh', 'argmax', 'argmin', 'argpartition', 'argsort', 'argwhere', 'around', 'array',
'array2string', 'array_equal', 'array_equiv', 'array_repr', 'array_split', 'array_str', 'asanyarray',
'asarray', 'asarray_chkfinite', 'ascontiguousarray', 'asfarray', 'asfortranarray', 'asmatrix', 'asscalar',
'atleast_1d', 'atleast_2d', 'atleast_3d', 'average', 'bartlett', 'base_repr', 'binary_repr', 'bincount',
'bitwise_and', 'bitwise_not', 'bitwise_or', 'bitwise_xor', 'blackman', 'block', 'bmat', 'bool8', 'bool_',
'broadcast', 'broadcast_arrays', 'broadcast_shapes', 'broadcast_to', 'busday_count', 'busday_offset',
'busdaycalendar', 'byte', 'byte_bounds', 'bytes0', 'bytes_', 'c_', 'can_cast', 'cast', 'cbrt', 'cdouble',
'ceil', 'cfloat', 'char', 'character', 'chararray', 'choose', 'clip', 'clongdouble', 'clongfloat',
'column_stack', 'common_type', 'compare_chararrays', 'compat', 'complex128', 'complex256',
'complex64', 'complex_', 'complexfloating', 'compress', 'concatenate', 'conj', 'conjugate',
'convolve', 'copy', 'copysign', 'copyto', 'core', 'corrcoef', 'correlate', 'cos', 'cosh', 'count_nonzero',
'cov', 'cross', 'csingle', 'ctypeslib', 'cumprod', 'cumproduct', 'cumsum', 'datetime64',
'datetime_as_string', 'datetime_data', 'deg2rad', 'degrees', 'delete', 'deprecate',
'deprecate_with_doc', 'diag', 'diag_indices', 'diag_indices_from', 'diagflat', 'diagonal', 'diff',
'digitize', 'disp', 'divide', 'divmod', 'dot', 'double', 'dsplit', 'dstack', 'dtype', 'e', 'ediff1d', 'einsum',
'einsum_path', 'emath', 'empty', 'empty_like', 'equal', 'errstate', 'euler_gamma', 'exp', 'exp2',
'expand_dims', 'expm1', 'expm1x', 'extract', 'eye', 'fabs', 'fastCopyAndTranspose', 'fft',
'fill_diagonal', 'find_common_type', 'finfo', 'fix', 'flatiter', 'flatnonzero', 'flexible', 'flip', 'fliplr',
'flipud', 'float128', 'float16', 'float32', 'float64', 'float_', 'float_power', 'floating', 'floor',
'floor_divide', 'fmax', 'fmin', 'fmod', 'format_float_positional', 'format_float_scientific',
'format_parser', 'frexp', 'frombuffer', 'fromfile', 'fromfunction', 'fromiter', 'frompyfunc',
'fromregex', 'fromstring', 'full', 'full_like', 'gcd', 'generic', 'genfromtxt', 'geomspace',
'get_array_wrap', 'get_include', 'get_printoptions', 'getbufsize', 'geterr', 'geterrcall', 'geterrobj',
'gradient', 'greater', 'greater_equal', 'half', 'hamming', 'hanning', 'heaviside', 'histogram',
'histogram2d', 'histogram_bin_edges', 'histogramdd', 'hsplit', 'hstack', 'hypot', 'i0', 'identity', 'iinfo',
'imag', 'in1d', 'index_exp', 'indices', 'inexact', 'inf', 'info', 'infty', 'inner', 'insert', 'int0', 'int16',
'int32', 'int64', 'int8', 'int_', 'intc', 'integer', 'interp', 'intersect1d', 'intp', 'invert', 'is_busday', 'isclose',
'iscomplex', 'iscomplexobj', 'isfinite', 'isfortran', 'isin', 'isinf', 'isnan', 'isnat', 'isneginf', 'isposinf',
'isreal', 'isrealobj', 'isscalar', 'issctype', 'issubclass_', 'issubdtype', 'issubsctype', 'iterable', 'ix_',
'kaiser', 'kernel_version', 'kron', 'lcm', 'ldexp', 'left_shift', 'less', 'less_equal', 'lexsort', 'lib', 'linalg',
'linspace', 'little_endian', 'load', 'loadtxt', 'log', 'log10', 'log1p', 'log2', 'logaddexp', 'logaddexp2',
'logical_and', 'logical_not', 'logical_or', 'logical_xor', 'logspace', 'longcomplex', 'longdouble',
'longfloat', 'longlong', 'lookfor', 'ma', 'mask_indices', 'mat', 'math', 'matmul', 'matrix', 'matrixlib',
'max', 'maximum', 'maximum_sctype', 'may_share_memory', 'mean', 'median', 'memmap',
'meshgrid', 'mgrid', 'min', 'min_scalar_type', 'minimum', 'mintypecode', 'mod', 'modf', 'moveaxis',
'msort', 'multiply', 'nan', 'nan_to_num', 'nanargmax', 'nanargmin', 'nancumprod', 'nancumsum',
'nanmax', 'nanmean', 'nanmedian', 'nanmin', 'nanpercentile', 'nanprod', 'nanquantile', 'nanstd',
'nansum', 'nanvar', 'nbytes', 'ndarray', 'ndenumerate', 'ndim', 'ndindex', 'nditer', 'negative',
'nested_iters', 'newaxis', 'nextafter', 'nonzero', 'not_equal', 'numarray', 'number', 'obj2sctype',
'object0', 'object_', 'ogrid', 'oldnumeric', 'ones', 'ones_like', 'os', 'outer', 'packbits', 'pad', 'partition',
'percentile', 'pi', 'piecewise', 'place', 'poly', 'poly1d', 'polyadd', 'polyder', 'polydiv', 'polyfit',
'polyint', 'polymul', 'polynomial', 'polysub', 'polyval', 'positive', 'power', 'printoptions', 'prod',
'product', 'promote_types', 'ptp', 'put', 'put_along_axis', 'putmask', 'quantile', 'r_', 'rad2deg',
'radians', 'random', 'ravel', 'ravel_multi_index', 'real', 'real_if_close', 'rec', 'recarray', 'recfromcsv',
'recfromtxt', 'reciprocal', 'record', 'remainder', 'repeat', 'require', 'reshape', 'resize', 'result_type',
'right_shift', 'rint', 'roll', 'rollaxis', 'roots', 'rot90', 'round', 'round_', 'row_stack', 's_', 'safe_eval',
'save', 'savetxt', 'savez', 'savez_compressed', 'sctype2char', 'sctypeDict', 'sctypes', 'searchsorted',
'select', 'set_numeric_ops', 'set_printoptions', 'set_string_function', 'setbufsize', 'setdiff1d', 'seterr',
'seterrcall', 'seterrobj', 'setxor1d', 'shape', 'shares_memory', 'short', 'show_config', 'sign', 'signbit',
'signedinteger', 'sin', 'sinc', 'single', 'singlecomplex', 'sinh', 'size', 'sometrue', 'sort', 'sort_complex',
'source', 'spacing', 'split', 'sqrt', 'square', 'squeeze', 'stack', 'std', 'str0', 'str_', 'string_', 'subtract',
'sum', 'swapaxes', 'sys', 'take', 'take_along_axis', 'tan', 'tanh', 'tensordot', 'test', 'testing', 'tile',
'timedelta64', 'trace', 'tracemalloc_domain', 'transpose', 'trapz', 'tri', 'tril', 'tril_indices',
'tril_indices_from', 'trim_zeros', 'triu', 'triu_indices', 'triu_indices_from', 'true_divide', 'trunc',
'typecodes', 'typename', 'ubyte', 'ufunc', 'uint', 'uint0', 'uint16', 'uint32', 'uint64', 'uint8', 'uintc',
'uintp', 'ulonglong', 'unicode_', 'union1d', 'unique', 'unpackbits', 'unravel_index', 'unsignedinteger',
'unwrap', 'use_hugepage', 'ushort', 'vander', 'var', 'vdot', 'vectorize', 'version', 'void', 'void0',
'vsplit', 'vstack', 'warnings', 'where', 'who', 'zeros', 'zeros_like']
Type: <class 'list'>
[1, 2, 3, 4, 5]
[[0, 1, 2], [3, 4, 5], [6, 7, 8]]
<class 'numpy.ndarray'>
[1 2 3 4 5]
[1. 2. 3. 4. 5.]
[False  True  True False False]
<class 'numpy.ndarray'>
[[0 1 2]
 [3 4 5]
 [6 7 8]]
<class 'list'>
one dimensional array: [1, 2, 3, 4, 5]
two dimensional array:  [[0, 1, 2], [3, 4, 5], [6, 7, 8]]
<class 'tuple'>
python_tuple:  (1, 2, 3, 4, 5)
<class 'numpy.ndarray'>
numpy_array_from_tuple:  [1 2 3 4 5]
[1 2 3 4 5]
shape of nums:  (5,)
[[0 1 2]
 [3 4 5]
 [6 7 8]]
shape of numpy_two_dimensional_list:  (3, 3)
(3, 4)
[-3 -2 -1  0  1  2  3]
int64
[-3. -2. -1.  0.  1.  2.  3.]
float64
The size: 5
The size: 9
original array:  [1 2 3 4 5]
[11 12 13 14 15]
original array:  [1 2 3 4 5]
[-9 -8 -7 -6 -5]
original array:  [1 2 3 4 5]
[10 20 30 40 50]
original array:  [1 2 3 4 5]
```

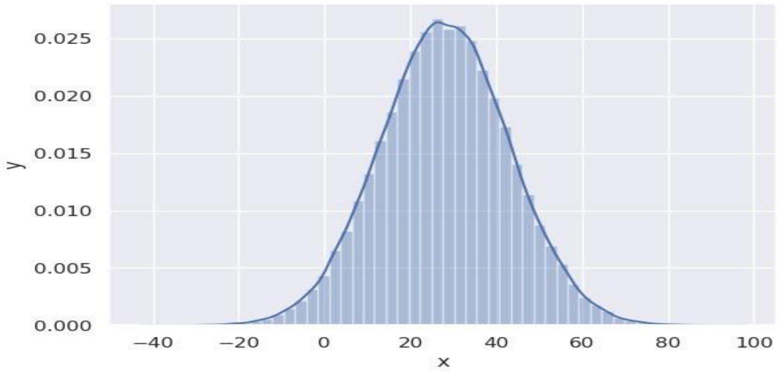
```
[0.1 0.2 0.3 0.4 0.5]
original array:  [1 2 3 4 5]
[1 2 0 1 2]
original array:  [1 2 3 4 5]
[0 0 0 0 0]
original array:  [1 2 3 4 5]
[ 1  4  9 16 25]
int64
float64
bool
<class 'numpy.ndarray'>
[[1 2 3]
 [4 5 6]
 [7 8 9]]
Shape:  (3, 3)
Size: 9
Data type: int64
First row: [1 2 3]
Second row: [4 5 6]
Third row:  [7 8 9]
First column: [1 4 7]
Second column: [2 5 8]
Third column:  [3 6 9]
[[1 2 3]
 [4 5 6]
 [7 8 9]]
[[1 2]
 [4 5]]
[[1 2 3]
 [4 5 6]
 [7 8 9]]
[[ 1  2  3]
 [ 4 55 44]
 [ 7  8  9]]
[[1 1 1]
 [1 1 1]
 [1 1 1]]
[[1 2 3]
 [4 5 6]]
[[1 2]
```

```
 [3 4]
 [5 6]]
[5 7 9]
Horizontal Append: [1 2 3 4 5 6]
Vertical Append: [[1 2 3]
 [4 5 6]]
0
2
4
6
8
10
First row:  [1 2 3]
Second row:  [4 5 6]
First column:  [1 4]
Second column:  [2 5]
Third column:  [3 6]
min:  1
max:  55
mean:  14.777777777777779
sd:  18.913709183069525
[[ 1  2  3]
 [ 4 55 44]
 [ 7  8  9]]
Column with minimum:  [1 2 3]
Column with maximum:  [ 7 55 44]
=== Row ==
Row with minimum:  [1 4 7]
Row with maximum:  [ 3 55  9]
Tile:    [1 2 3 1 2 3]
Repeat:  [1 1 2 2 3 3]
0.6221433907437441
[[0.10204494 0.76399148 0.60378347]
 [0.14595641 0.82338019 0.90873077]]
['e' 'e' 'a' 'o' 'o' 'i' 'e' 'e' 'o' 'u']
min:  3.2302617205929933
max:  6.476754600683039
mean:  4.989011809431645
median:  4.987293688381718
mode:  ModeResult(mode=array([3.23026172]), count=array([1]))
```

```
sd:  0.4838372968678121
<ipython-input-14-07cddcfeda53>:417: FutureWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the default behavior of `mode` typically preserves the axis it acts along. In SciPy 1.11.0, this behavior will change: the default value of `keepdims` will become False, the `axis` over which the statistic is taken will be eliminated, and the value None will no longer be accepted. Set `keepdims` to True or False to avoid this warning.
  print('mode: ', stats.mode(np_normal_dis))
```







7. Write programs to use the pandas data structures: Frames and series as storage containers and for a variety of data-wrangling operations, such as:

A. **Single-level and hierarchical indexing**
B. **Handling missing data**
C. **Arithmetic and Boolean operations on entire columns and tables**
D. **Database-type operations (such as merging and aggregation)**
E. **Plotting individual columns and whole tables**
F. **Reading data from files and writing data to files**

```
#Pandas
# ## 1. Setup
# ### Import
# Before moving on to learn pandas first we need to install it and import it. If you install
[Anaconda distributions](https://www.anaconda.com/) on your local machine or using [Google
Colab](https://research.google.com/colaboratory) then pandas will already be available there,
otherwise, you follow this installation process from [pandas official's
website](https://pandas.pydata.org/docs/getting_started/install.html).
# Importing libraries
import numpy as np
import pandas as pd
# we can set numbers for how many rows and columns will be displayed
pd.set_option('display.min_rows', 10) #default will be 10
pd.set_option('display.max_columns', 20)

# ## 2. Loading Different Data Formats Into a Pandas Data Frame
# ### Reading CSV file
# read csv file
df = pd.read_csv('online_store_customer_data.csv')
df.head(3)
# Loading csv file with skip first 2 rows without header
df_csv = pd.read_csv('online_store_customer_data.csv', skiprows=2, header=None)
df_csv.head(3)
# ### Read CSV file from URL
# Read csv file from url
#from io import StringIO
#import requests
url="https://github.com/norochalise/pandas-tutorial-article-
2022/tree/main/dataset/online_store_customer_data.csv"
#s=requests.get(url).content
```

```
#c=pd.read_csv(io.StringIO(s.decode('utf-8')))
#df_url = pd.read_csv(url)
#s=requests.get(url).text
#df_url = pd.read_csv(StringIO(s))
df_url = pd.read_csv(url, sep = "\t")
df_url.head(3)
# ### Write CSV file
# saving df_url dataframe to csv file
df_url.to_csv('csv_from_url.csv')
df_url.to_csv('demo_text.txt')
# ### Read text file
# read plain text file
df_txt = pd.read_csv("demo_text.txt")
# ### Read Excel file
# read excel file
df_excel = pd.read_excel('excel_file.xlsx', sheet_name='Sheet1')
df_excel
# ### Write Excel file
# save dataframe to the excel file
df_url.to_csv('demo.xlsx')
# ## 3. Data preprocessing
# Data preprocessing is the process of making raw data to clean data. This is the most crucial
part of data the science. In this section, we will explore data first then we remove unwanted
columns, remove duplicates, handle missing data, etc. After this step, we get clean data from raw
data.
# ### 3.1 Data Exploring
# #### Retrieving rows from data frame.
# display first 3 rows
df.head(3)
# display last 6 rows
df.tail(6)
# #### Retrieving sample rows from data frame.
# Display random 7 sample rows
df.sample(7)
# #### Retrieving information about dataframe
df.info()
# display datatypes
df.dtypes
df.dtypes.value_counts()
# #### Display number of rows and columns.
```

```
df.shape
df.columns
# display Age columns first 3 rows data
df['Age'].head(3)
# display first 4 rows of Age, Transaction_date and Gender columns
df[['Age', 'Transaction_date', 'Gender']].head(4)
# #### Retrieving a Range of Rows
# for display 2nd to 6th rows
df[2:7]
# for display starting to 10th
df[:11]
# for display last two rows
df[-2:]

#  ### 3.2 Data Cleaning
# After the explore our datasets may need to clean them for better analysis. Data coming in from
multiple sources so It's possible to have an error in some values. This is where data cleaning
becomes extremely important. In this section, we will delete unwanted columns, rename
columns, correct appropriate data types, etc.
# #### Delete Columns name
# Drop unwanted columns
df.drop(['Transaction_ID'], axis=1, inplace=True)
# #### Change Columns name
# create new df_col dataframe from df.copy() method.
df_col = df.copy()
# rename columns name
df_col.rename(columns={"Transaction_date": "Date", "Gender": "Sex"}, inplace=True)
df_col.head(3)
# #### Adding a new column to a DataFrame
# Add a new ajusted column which value will be amount_spent * 100
df_col['new_col'] = df_col['Amount_spent'] * 100
df_col.head(3)
# #### String value change or replace
df_col.head(3)
# changing Female to Woman and Male to Man in Sex column.
#first argument in loc function is condition and second one is columns name.
df_col.loc[df_col.Sex == "Female", 'Sex'] = 'Woman'
df_col.loc[df_col.Sex == "Male", 'Sex'] = 'Man'
df_col.head(3)
# Now Sex columns values are changed Female to Woman and Male to Man.
# #### Datatypes change
df_col.info()
# In our `Date` columns, it's object type so now we will convert this to date types, and also we will convert `Referal` columns float64 to float32.
# change object type to datefime64 format
df_col['Date'] = df_col['Date'].astype('datetime64[ns]')
# change float64 to float32 of Referal columns
df_col['Referal'] = df_col['Referal'].astype('float32')
df_col.info()

# ### 3.3 Remove duplicate
# Display duplicated entries
df.duplicated().sum()
# duplicate rows dispaly, keep arguments will--- 'first', 'last' and False
duplicate_value = df.duplicated(keep='first')
df.loc[duplicate_value, :]
# dropping ALL duplicate values
df.drop_duplicates(keep = 'first', inplace = True)

# ### 3.4 Handling missing values
# Handling missing values in the common task in the data pre-processing part. For many reasons most of the time we will encounter missing values. Without dealing with this we can't do the proper model building. For this section first, we will find out missing values then we decided how to handle them. We can handle this by removing affected columns or rows or replacing appropriate values there.
# #### Display missing values information
df.isna().sum().sort_values(ascending=False)
# #### Delete Nan rows
# If we have less Nan value then we can delete entire rows by `dropna()` function. For this function, we will add columns name in subset parameter.
# df copy to df_copy
df_new = df.copy()
#Delete Nan rows of Job Columns
df_new.dropna(subset = ["Employees_status"], inplace=True)
# #### Delete entire columns
# If we have a large number of nan values in particular columns then dropping those columns might be a good decision rather than imputing.
df_new.drop(columns=['Amount_spent'], inplace=True)
df_new.isna().sum().sort_values(ascending=False)
# #### Impute missing values
# Sometimes if we delete entire columns that will be not the appropriate approach. Delete columns can affect our model building because we will lose our main features. For imputing we have many approaches so here are some of the most popular techniques.
# **Method 1** - Impute fixed value like 0, 'Unknown' or 'Missing' etc. We inpute Unknown in Gender columns
df['Gender'].fillna('Unknown', inplace=True)
# **Method 2** - Impute Mean, Median and Mode
# Impute Mean in Amount_spent columns
mean_amount_spent = df['Amount_spent'].mean()
df['Amount_spent'].fillna(mean_amount_spent, inplace=True)
#Impute Median in Age column
median_age = df['Age'].median()
df['Age'].fillna(median_age, inplace=True)
# Impute Mode in Employees_status column
mode_emp = df['Employees_status'].mode().iloc[0]
df['Employees_status'].fillna(mode_emp, inplace=True)
# **Method 3** - Imputing forward fill or backfill by `ffill` and `bfill`. In `ffill` missing value impute from the value of the above row and for `bfill` it's taken from the below rows value.
df['Referal'].fillna(method='ffill', inplace=True)
df.isna().sum().sum()
# Now we deal with all missing values with different methods. So now we haven't any null values.

# ## 4. Memory management
# When we work on large datasets, There we get one big issue is a memory problem. We need too large resources for dealing with this. But there are some methods in pandas to deal with this. Here are some methods or strategies to deal with this problem with help of pandas.
# ### Change datatypes
df_memory = df.copy()
memory_usage = df_memory.memory_usage(deep=True)
memory_usage_in_mbs = round(np.sum(memory_usage / 1024 ** 2), 3)
print(f" Total memory taking df_memory dataframe is : {memory_usage_in_mbs:.2f} MB ")
# #### Change object to category datatypes
# Our data frame is small in size. Which is 1.15 MB. Now We will convert our object datatype to category.
# Object datatype to category convert
df_memory[df_memory.select_dtypes(['object']).columns] =
df_memory.select_dtypes(['object']).apply(lambda x: x.astype('category'))
# convert object to category
df_memory.info(memory_usage="deep")
```

```
# Now its reduce 1.15 megabytes to 216.6 kb. It's almost reduced 5.5 times.
# #### Change int64 or float64 to int 32, 16, or 8
# By default, pandas store numeric values to int64 or float64. Which takes more memory. If we have to store small numbers then we can change to 64 to 32, 16, and so on. For example, our Referal columns have only 0 and 1 values so for that we don't need to store at float64. so now we change it to float16.
# Change Referal column datatypes
df_memory['Referal'] = df_memory['Referal'].astype('float32')
# convert object to category
df_memory.info(memory_usage="deep")
# After changing only one column's data types we reduce 216 kb to 179 kb.
# **Note: Before changing datatypes please make sure it's consequences.**

# ## 5. Data Analysis

# ### 5.1. Calculating Basic statistical measurement
df.describe().T
# We know already above code will display only numeric columns basic statistical information. for object or category columns we can use `describe(include=object)` .
df.describe(include=object).T
# We can calculate the mean, median, mode, maximum values, minimum values of individual columns we simply use these functions.
# Calculate Mean
mean = df['Age'].mean()
# Calculate Median
median = df['Age'].median()
#Calculate Mode
mode = df['Age'].mode().iloc[0]
# Calculate standard deviation
std = df['Age'].std()
# Calculate Minimum values
minimum = df['Age'].min()
# Calculate Maximum values
maximum = df.Age.max()
print(f" Mean of Age : {mean}")
print(f" Median of Age : {median}")
print(f" Mode of Age : {mode}")
print(f" Standard deviation of Age : {std:.2f}")
print(f" Maximum of Age : {maximum}")
print(f" Minimum of Age : {minimum}")
```

```
# In pandas we can display the correlation of different numeric columns. For this we can use `.corr()` function.
# calculate correlation
df.corr(numeric_only = True)

# ### 5.2 Basic built in function for data analysis
# #### Number of uniqe values in category column
# for display how many unique values are there in State_names column
df['State_names'].nunique()
# #### Shows all unique values
# for display uniqe values of State_names column
df['State_names'].unique()
# #### Counts of unique values
df['Gender'].value_counts()
# If we want to show with the percentage of occurrence rather number than we use `normalize=True` argument in `value_counts()` function
# Calculate percentage of each category
df['Gender'].value_counts(normalize=True)
df['State_names'].value_counts().sort_values(ascending = False).head(20)
# Sort Values by State_names
df.sort_values(by=['State_names']).head(3)
# For sorting our dataframe by Amount_spent with ascending order:
# Sort Values Amount_spent with ascending order
df.sort_values(by=['Amount_spent']).head(3)
# For sorting our dataframe by Amount_spent with descending order:
# Sort Values Amount_spent with descending order
df.sort_values(by=['Amount_spent'], ascending=False).head(3)
# Alternatively, We can use `nlargest()` and `nsmallest()` functions for displaying largest and smallest values with desired numbers. for example, If we want to display 4 largest Amount_spent rows then we use this:
# nlargest
df.nlargest(4, 'Amount_spent').head(10) # first argument is how many rows you want to disply and second one is columns name
# For 3 smallest Amount_spent rows
# nsmallest
df.nsmallest(3, 'Age').head(10)
# #### Conditional queries on Data
# filtering - Only show Paypal users
condition = df['Payment_method'] == 'PayPal'
df[condition].head(4)
```

```
# We can apply multiple conditional queries like before. For example, if we want to display all
Married female people who lived in New York then we use the following:
# first create 3 condition
female_person = df['Gender'] == 'Female'
married_person = df['Marital_status'] == 'Married'
loc_newyork = df['State_names'] == 'New York'
# we passing condition on our dataframe
df[female_person & married_person & loc_newyork].head(4)
# ### 5.3 Summarizing or grouping data
# #### Groupby
# **Grouping by one column:** For example, if we want to find `maximum` values of `Age`
and `Amount_spent` by `Gender` then we can use this:

df[['Age', 'Amount_spent']].groupby(df['Gender']).max()
# To find `mean`, `count`, and `max` values of `Age` and `Amount_spent` by `Gender` then we
can use `agg()` function with `groupby()` .
# Group by one columns
state_gender_res = df[['Age','Gender','Amount_spent']].groupby(['Gender']).agg(['count', 'mean',
'max'])
state_gender_res
# **Grouping by multiple columns:** To find total count, maximum and minimum values of
Amount_spent by State_names, Gender, and Payment_method then we can pass these columns
names under `groupby()` function and add `.agg()` with `count`, `mean`, `max` argument.
#Group By multiple columns
state_gender_res = df[['State_names','Gender','Payment_method','Amount_spent']].groupby([
'State_names','Gender', 'Payment_method']).agg(['count', 'min', 'max'])
state_gender_res.head(12)
# #### Cross Tabulation (Crosstab)
# For creating a simple crosstab between Maritatal_status and Payment_method columns we just
use `crosstab()` with both column names.
pd.crosstab(df.Marital_status, df.Payment_method)
# We can include subtotals by `margins` parameter:
pd.crosstab(df.Marital_status, df.Payment_method, margins=True, margins_name="Total")
# If We want a display with percentage than `normalize=True` parameter help
pd.crosstab(df.Marital_status, df.Payment_method, normalize=True, margins=True,
margins_name="Total")
# In this crosstab features, we can pass multiple columns names for grouping and analyzing data.
For instance, If we want to see how the `Payment_method` and `Employees_status` are
distributed by `Marital_status` then we will pass these columns' names in `crosstab()` function
```

```
and it will show below.
pd.crosstab(df.Marital_status, [df.Payment_method, df.Employees_status])

# ## 6. Data Visualization
# Visualization is the key to data analysis. The most popular python package for visualization are matplotlib and seaborn but sometimes pandas will be handy for you. Pandas also provide some visualization plots easily. For the basic analysis part, it will be easy to use. For this section, we are exploring some different types of plots using pandas. Here are the plots.

# ### 6.1 Line plot
# For creating a line plot in pandas we use `.plot()` with two columns name for the argument. For example, we create a line plot from one dummy dataset.
dict_line = {
    'year': [2016, 2017, 2018, 2019, 2020, 2021],
    'price': [200, 250, 260, 220, 280, 300]
}
df_line = pd.DataFrame(dict_line)
# use plot() method on the dataframe
df_line.plot('year', 'price');
# The above line chart shows prices over a different time. It shows like price trend.

# ### 6.2 Bar plot
# A bar plot is also known as a bar chart shows quantitative or qualitative values for different category items. In a bar plot data are represented in the form of bars. Bars length or height are used to represent the quantitative value for each item. Bar plot can be plotted horizontally or vertically. For creating these plots look below.
# **For horizontal bar:**
df['Employees_status'].value_counts().plot(kind='bar');
# **For vertical bar:**
df['Employees_status'].value_counts().plot(kind='barh');

# ### 6.3 Pie plot
# A pie plot is also known as a pie chart. A pie plot is a circular graph that represents the total value with its components. The area of a circle represents the total value and the different sectors of the circle represent the different parts. In this plot, the data are expressed as percentages. Each component is expressed as a percentage of the total value.
# In pandas for creating pie plot. We use `kind=pie` in `plot()` function in dataframes columns or series.
df['Segment'].value_counts().plot(kind='pie');
```

```
# ### 6.4 Box Plot
# A box plot is also known as a box and whisker plot. This plot is used to show the distribution of a variable based on its quartiles. Box plot displays the five-number summary of a set of data. The five-number summary is the minimum, first quartile, median, third quartile, and maximum. It will also be popular to identify outliers.
# We can plot this by one column or multiple columns. For multiple columns, we need to pass columns name in `y` variable as a list.
df.plot(y=['Amount_spent'], kind='box');
# In a boxplot, we can plot the distribution of categorical variables against a numerical variable and compare them. Let's plot it with the Employees_status and Amount_spent columns with pandas `boxplot()` method:
import matplotlib.pyplot as plt
#np.warnings.filterwarnings('ignore', category=np.VisibleDeprecationWarning)
#fig, ax = plt.subplots(figsize=(6,6))
#df.boxplot(by ='Employees_status', column =['Amount_spent'],ax=ax, grid = False);

# ### 6.5 Histogram
# A histogram shows the frequency and distribution of quantitative measurement across grouped values for data items. It is commonly used in statistics to show how many of a certain type of variable occurs within a specific range or bucket. Below we will plot a histogram for looking Age distribution.
df.plot(
    y='Age',
    kind='hist',
    bins=10
);
# ### 6.6 KDE plot
# A kernel density estimate (KDE) plot is a method for visualizing the distribution of observations in a dataset, analogous to a histogram. KDE represents the data using a continuous probability density curve in one or more dimensions.
df.plot(
    y='Age',
    xlim=(0, 100),
    kind='kde'
);
# ### 6.7 Scatterplot
# A scatterplot is used to observe and show relationships between two quantitative variables for different category items. Each member of the dataset gets plotted as a point whose x-y coordinates relate to its values for the two variables. Below we will plot a scatterplot to display
```

relationships between Age and Amount_spent columns.

```
df.plot(
    x='Age',
    y='Amount_spent',
    kind='scatter'
)
```

Output

```
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\venv\Scripts\python.exe
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\pandas_tutorial.py

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2512 entries, 0 to 2511
Data columns (total 11 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Transaction_date  2512 non-null   object
 1   Transaction_ID    2512 non-null   int64
 2   Gender            2484 non-null   object
 3   Age               2470 non-null   float64
 4   Marital_status    2512 non-null   object
 5   State_names       2512 non-null   object
 6   Segment           2512 non-null   object
 7   Employees_status  2486 non-null   object
 8   Payment_method    2512 non-null   object
 9   Referal           2357 non-null   float64
 10  Amount_spent      2270 non-null   float64
dtypes: float64(3), int64(1), object(7)
memory usage: 216.0+ KB
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2512 entries, 0 to 2511
Data columns (total 11 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Date              2512 non-null   object
 1   Sex               2484 non-null   object
 2   Age               2470 non-null   float64
 3   Marital_status    2512 non-null   object
 4   State_names       2512 non-null   object
 5   Segment           2512 non-null   object
 6   Employees_status  2486 non-null   object
 7   Payment_method    2512 non-null   object
 8   Referal           2357 non-null   float64
 9   Amount_spent      2270 non-null   float64
 10  new_col           2270 non-null   float64
dtypes: float64(4), object(7)
memory usage: 216.0+ KB
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2512 entries, 0 to 2511
Data columns (total 11 columns):
 #   Column            Non-Null Count  Dtype
- -  -------           - - - - - - - - - -  ------
 0   Date              2512 non-null   datetime64[ns]
 1   Sex               2484 non-null   object
 2   Age               2470 non-null   float64
 3   Marital_status    2512 non-null   object
 4   State_names       2512 non-null   object
 5   Segment           2512 non-null   object
 6   Employees_status  2486 non-null   object
 7   Payment_method    2512 non-null   object
 8   Referal           2357 non-null   float32
 9   Amount_spent      2270 non-null   float64
 10  new_col           2270 non-null   float64
dtypes: datetime64[ns](1), float32(1), float64(3), object(6)
memory usage: 206.2+ KB
 Total memory taking df_memory dataframe is : 1.15 MB
<class 'pandas.core.frame.DataFrame'>
Index: 2500 entries, 0 to 2511
Data columns (total 10 columns):
 #   Column            Non-Null Count  Dtype
- -  -------           - - - - - - - - - -  ------
 0   Transaction_date  2500 non-null   category
 1   Gender            2500 non-null   category
 2   Age               2500 non-null   float64
 3   Marital_status    2500 non-null   category
 4   State_names       2500 non-null   category
 5   Segment           2500 non-null   category
 6   Employees_status  2500 non-null   category
 7   Payment_method    2500 non-null   category
 8   Referal           2500 non-null   float64
 9   Amount_spent      2500 non-null  float64
dtypes: category(7), float64(3)
memory usage: 189.1 KB
<class 'pandas.core.frame.DataFrame'>
Index: 2500 entries, 0 to 2511
Data columns (total 10 columns):
 #   Column           Non-Null Count  Dtype
- -  -------         - - - - - - - - - -  ------
 0   Transaction_date  2500 non-null   category
 1   Gender            2500 non-null   category
 2   Age               2500 non-null   float64
 3   Marital_status    2500 non-null   category
 4   State_names       2500 non-null   category
 5   Segment           2500 non-null   category
 6   Employees_status  2500 non-null   category
 7   Payment_method    2500 non-null   category
 8   Referal           2500 non-null   float32
 9   Amount_spent      2500 non-null  float64
dtypes: category(7), float32(1), float64(2)
memory usage: 179.3 KB
 Mean of Age : 46.636
 Median of Age : 47.0
 Mode of Age : 47.0
 Standard deviation of Age : 18.02
 Maximum of Age : 78.0
 Minimum of Age : 15.0

Process finished with exit code 0
```

Additional Exercises (for learning and practice) :
14. Introduction to Python Programming:

Running instructions in Interactive interpreter and a Python Script.

A commonly used method of running Python code is via an interactive session. We can run the Python interpreter in interactive mode. In an interactive session, you can test every piece of code as you go, making this a useful development tool. Statements written when working with an interactive session are evaluated and executed immediately. When the interactive session ends, the code ceases to exist.

What is a Shell?

An operating system is made up of a bunch of programs. They perform tasks like file handling, memory management, and resource management, and they help your applications run smoothly.

All the work we do on computers, like analyzing data in Excel or playing games, is facilitated by the operating system.

Operating system programs are of two types, called **shell** and **kernel** programs.

Kernel programs are the ones who perform the actual tasks, like creating a file or sending interrupts. Shell is another program, whose job is to take input and decide and execute the required kernel program to do the job and show the output.

The shell is also called the **command processor**.

What is a Terminal?

The terminal is the program that interacts with the shell and allows us to communicate with it via text-based commands. This is why it's also called the command line.

To access the terminal on Windows, hit the Windows logo + R, type cmd, and press Enter.

To access the terminal on Ubuntu, hit Ctrl + Alt + T.

What is the Python Shell?

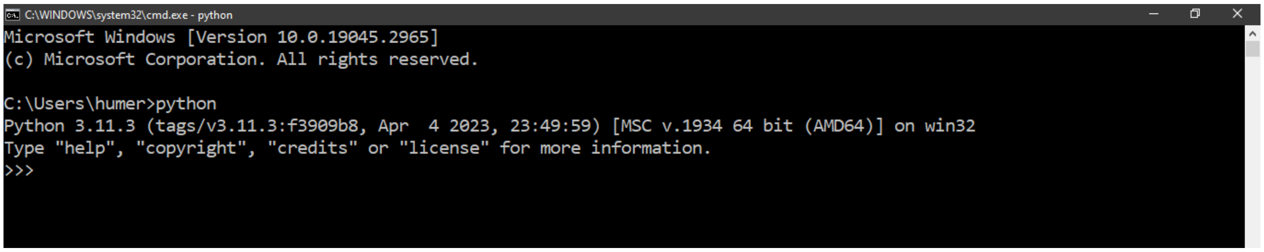
Python is an interpreted language. This means that the Python interpreter reads a line of code, executes that line, then repeats this process if there are no errors.

The Python Shell gives you a command line interface you can use to specify commands directly to the Python interpreter in an interactive manner.

You can get a lot of detailed information regarding the Python shell in the official docs.

How to Use the Python Shell

To start the Python shell, simply type python and hit Enter in the terminal:

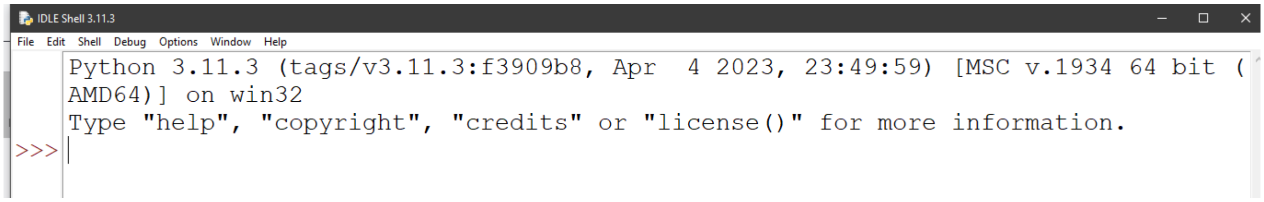


```
Microsoft Windows [Version 10.0.19045.2965]
(c) Microsoft Corporation. All rights reserved.

C:\Users\humer>python
Python 3.11.3 (tags/v3.11.3:f3909b8, Apr  4 2023, 23:49:59) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

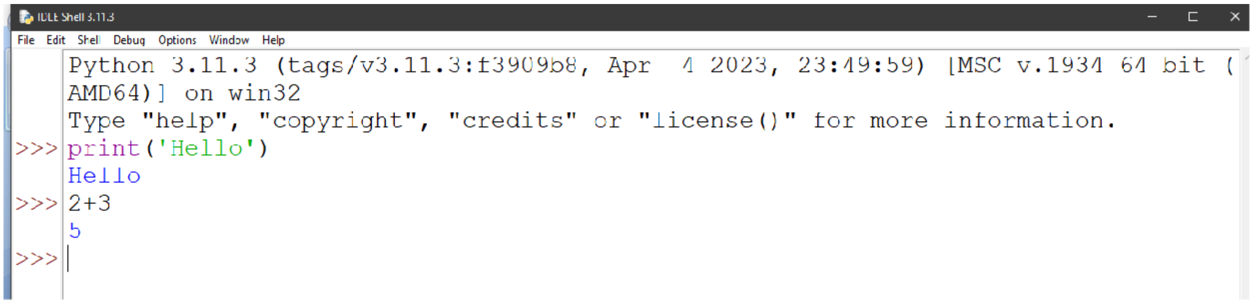
The Python interactive mode: IDLE

To run the Python code, we can use the Python interactive session. We need to start Python interactive session, just open a command-line or terminal in start menu, then type in python, and press enter key.

Here is the example of how to run Python code using interactive shell.



```
Python 3.11.3 (tags/v3.11.3:f3909b8, Apr  4 2023, 23:49:59) [MSC v.1934 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
```



```
Python 3.11.3 (tags/v3.11.3:f3909b8, Apr  4 2023, 23:49:59) [MSC v.1934 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print('Hello')
Hello
>>> 2+3
5
>>>
```

The interactive shell is also called REPL which stands for read, evaluate, print, loop. It'll read each command, evaluate and execute it, print the output for that command if any, and continue this same process repeatedly until you quit the shell.

There are different ways to quit the shell:

you can hit Ctrl+Z on Windows or Ctrl+D on Unix systems to quit use the exit() command

use the quit() command

The IDE

To run a Python script from an IDE, start a project first. Once the project is created add your .py files (or create them in the IDE) and press run.

In the PyCharm IDE:

Start project

Welcome screen opens, click Create New Project.

On the main menu, choose File | New Project.

Select Python interpreter

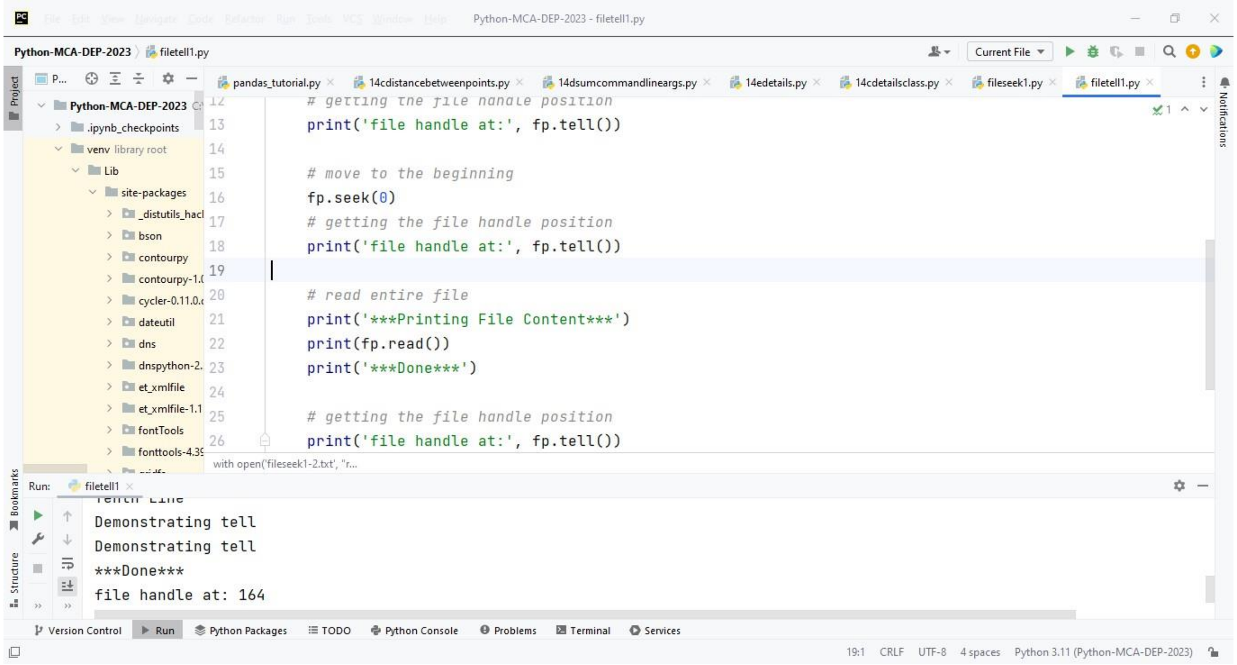
Choose Python version from the list. Use 3.x

Click create

Add new Python file (File new) and add hello.py

Click the green triangle to start the program. Another option is to click right mouse button on your Python file and selecting run.

Other IDEs have a similar process to run a Python program (start project, add file, run button).



```
    # getting the file handle position
    print('file handle at:', fp.tell())

    # move to the beginning
    fp.seek(0)
    # getting the file handle position
    print('file handle at:', fp.tell())

    # read entire file
    print('***Printing File Content***')
    print(fp.read())
    print('***Done***')

    # getting the file handle position
    print('file handle at:', fp.tell())
```

with open('fileseek1-2.txt', "r...

```
Tenth Line
Demonstrating tell
Demonstrating tell
***Done***
file handle at: 164
```

What Can You Do in the Python Shell?

You can do pretty much everything that the Python language allows, from using variables, loops, and conditions to defining functions and more.

The >>> is the shell prompt where you type in your commands. If you have commands that span across several lines – for example when you define loops – the shell prints the ... characters which signifies that a line continues.

Let's see an example:

```
>>>

>>> watch_list = ["stranger_things_s1", "stranger_things_s2",
"stranger_things_s3","stranger_things_s4"]

>>>

>>>
```

Here we defined a list with some TV show names via the Python shell.

Next, let's define a function that accepts a list of shows and randomly returns a show:

```
>>> def weekend_party(show_list):

...     r = random.randint(0, len(show_list)-1)

...     return show_list[r]

...
```

Note the continuation lines (...) of the Python shell here.

Finally to invoke the function from the shell, you simply call the function the way you would do in a script:

```
>>> weekend_party(watch_list)

'stranger_things_s1'

>>>
```

You can inspect Python modules from the shell, as shown below:

```
>>>

>>>
```

```
>>> import numpy

>>> numpy.__version_

'1.20.1'

>>>
```

You can see what methods and attributes a module offers by using the dir() method:

```
>>>

>>> x = dir(numpy)

>>> len(x)

606

>>> x[0:3]

['ALLOW_THREADS', 'AxisError', 'BUFSIZE']
```

Here you can see that Numpy has 606 methods and properties in total.

How to Run Python Scripts

The Python shell is useful for executing simple programs or for debugging parts of complex programs.

But really large Python programs with a lot of complexity are written in files with a .py extension, typically called Python scripts. Then you execute them from the terminal using the Python command.

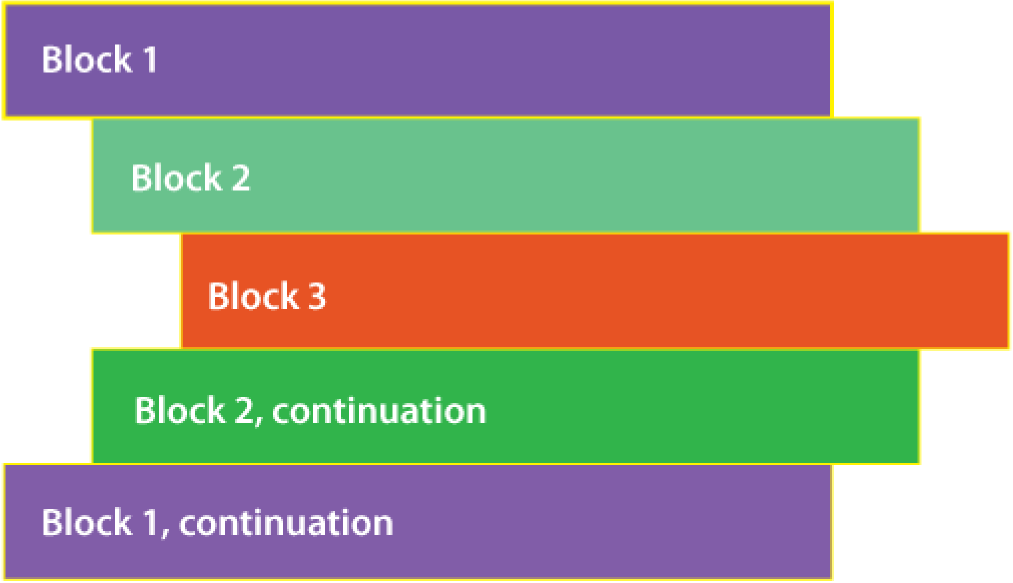
The usual syntax is:

```
python filename.py
```

All the commands we executed previously via the shell, we can also write it in a script and run in this way.

Write a program to purposefully raise Indentation Error and Correct it

What is an indentation in Python Programming Language?



Indentation refers to implementing proper **spaces and tabs** at starting any statement, method and block of code in the python programming language.

Due to these indentations, we can easily identify the beginning point and at the endpoint of any conditional loop, functions, if-else statements and so on.

Suppose we observe the other programming languages like C, C++, and Java. In that case, there is the use of small and curly braces along with the semicolon for specifying the particular block of code and identifying the particular statement.

But here in python, we will not be concerned about using the semicolon and different braces in syntax.

There is no importance of spaces in C. Other programming languages or the compiler will ignore all the whitespace and comments, in the lexical phase itself, in the starting phase; hence, it cannot generate syntax errors.

The role of all whitespaces matters; if a block of code is starting by applying a white space, it must end on the same indent.

All statements must have the same number of whitespaces or tabs so that the distance to the left of the screen belongs to the same block of code. If a block has to be more deeply nested, it is indented further to the right.

Suppose we observe in the python programming language. In that case, there is no concept of the main function, as we compared it with other programming languages. They all have a concept of the main function, which is executed after then other functions. If the main function calls the other function, the pointer will point to execute the other functions and again return to the main function.

Due to the indentation, it is decided that the initial indented block is the body of the main function. If there are other functions, then the initial indented block will call that function. The indentation of the additional function is also different from the main body so that the compiler can distinguish the block of the main and derived function.

AD

An indentation error occurs whenever there is no ending of the particular block, or the number of whitespaces assigned for starting the code block is not the same. At the same time that blocks ends, ultimately, an indentation error occurs.

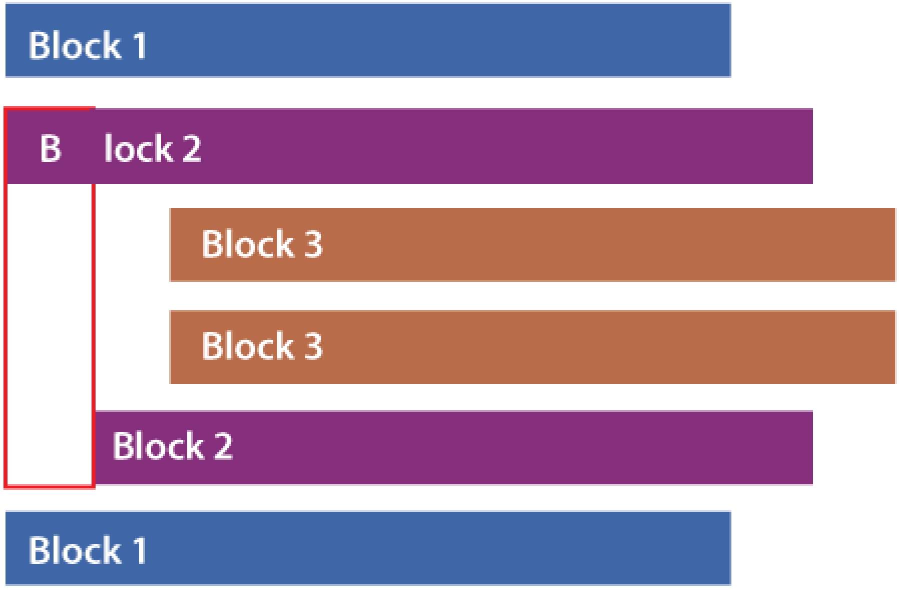
Description of the above figure -

In the above figure, statement one will be executed. Then, the pointer will point to the ' if statement ' and check the condition; if the condition is true, it will enter the block and enter the nested if-else block.

Again in the nested if statement, it will check the condition if it is true, then executes the next statement.

In this way, the execution of the statements will be carried on, now; if we take a case where the block fails, it will not enter into the nested if-else block. It directly exit out form all the blocks and the final statement will execute, which matches with the block 1, initial block.

Now let us see the figure having indentation error -



Error:
indentationerror: unindent does not match any outer indentation level

The figure mentioned above clearly has an indentation error because the block 2 indentation does not match the starting and ending point whitespaces. The beginning of block 2 will contain 0 whitespaces, but while it ends, it contains 2 whitespaces which will produce an error.

Now a doubt arises that, if block 2 below block 3 can act as block 4 a nested block, then it will not generate an indentation error, then the answer of this doubt is no because block 4; a nested block below block 3 will have more whitespaces as compare to the block 3. After all, it is nested inside block 3. And the second reason is that the python interpreter will find the end of block 2, but it will be unable to detect the matching whitespaces from block 2; hence it will generate an indentation error.

How to fix indentation error in a python programming language?

There are points that one must remember while fixing the indentation error :

As we have seen earlier, indentation error occurs only due to the unmatched whitespaces and tabs; hence, we need to fix all the **whitespaces** and **tabs.**

Examples of Python Indentation Error:

Example 1:

# Python program to find the maximum out of two numbers:

```
def max(x,y):  # max function will return the maximum among the two numbers
  if(x>y):
  return x
  else:
    return y
a = int(input("Enter a number: "))
b = int(input("Enter another number: "))
print("Finding the Maximum out of a:", a ,"and b:", b)
c=max(a,b)     # calling the max function
print(c,"is maximum")           # printing the result
```

The output of the above program:

```
IndentationError: expected an indented block
```

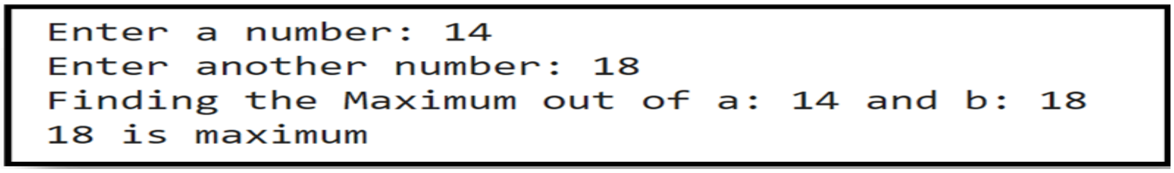
Now Fixing the Indentation Error in the above examples of python Programs :

Example 1:

```
# Python program to find the maximum out of two numbers:
def max(x,y):  # max function will return the maximum among the two numbers
  if(x>y):
    return x
  else:
    return y
a = int(input("Enter a number: "))
b = int(input("Enter another number: "))
print("Finding the Maximum out of a:", a ,"and b:", b)
c=max(a,b)     # calling the max function
```

```
print(c,"is maximum")          # printing the result
```

The output of the above program:



```
Enter a number: 14
Enter another number: 18
Finding the Maximum out of a: 14 and b: 18
18 is maximum
```

a) Write a program to compute distance between two points taking input from the user

Distance can be calculated using the two points (x_1, y_1) and (x_2, y_2), the distance d between these points is given by the formula:

$$\sqrt{(x_2-x_1)^2+(y_2-y_1)^2}$$

for e.g : let x_1 , y_1=10,9 and x_2 , y_2=4,1 then $(x_2-x_1)^2$=$(10-4)^2$ = 6^2 = 36 and $(y_2-y_1)^2$= $(9-1)^2$ = 8^2 = 64 now 64 + 36 =100 and 100 is square root of 10 sp distance between (10,9) and (4,1) is 10 .

```
#Usimg operators
x1=int(input("enter x1 : "))
x2=int(input("enter x2 : "))
y1=int(input("enter y1 : "))
y2=int(input("enter y2 : "))
result= ((((x2 - x1 )**2) + ((y2-y1)**2) )**0.5)
print("distance between",(x1,x2),"and",(y1,y2),"is : ",result)
#Using Built in function
import math
a=input("enter first coordinate : ")
p1 = a.split(",")
b=input("enter second coordinate : ")
p2 = b.split(",")
distance = math.sqrt( ((int(p1[0])-int(p2[0]))**2)+((int(p1[1])-int(p2[1]))**2) )
print("distance between ", a,"and", b, "is",distance)
```

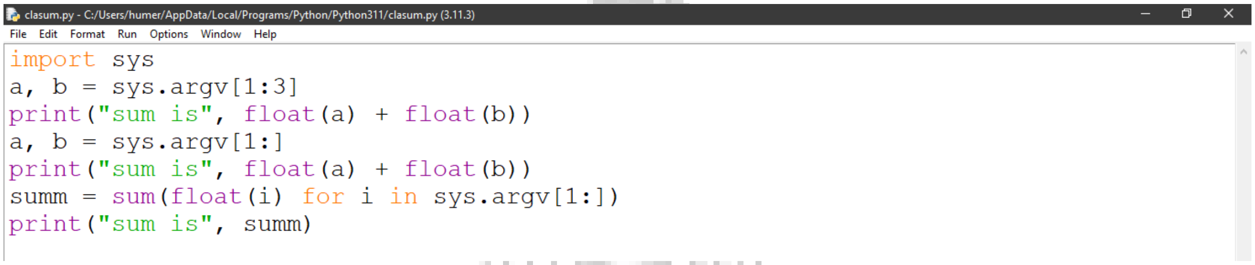
Output

```
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\venv\Scripts\python.exe
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\14cdistancebetweenpoints.py

enter x1 : 4

enter x2 : 3

enter y1 : 2

enter y2 : 5
```

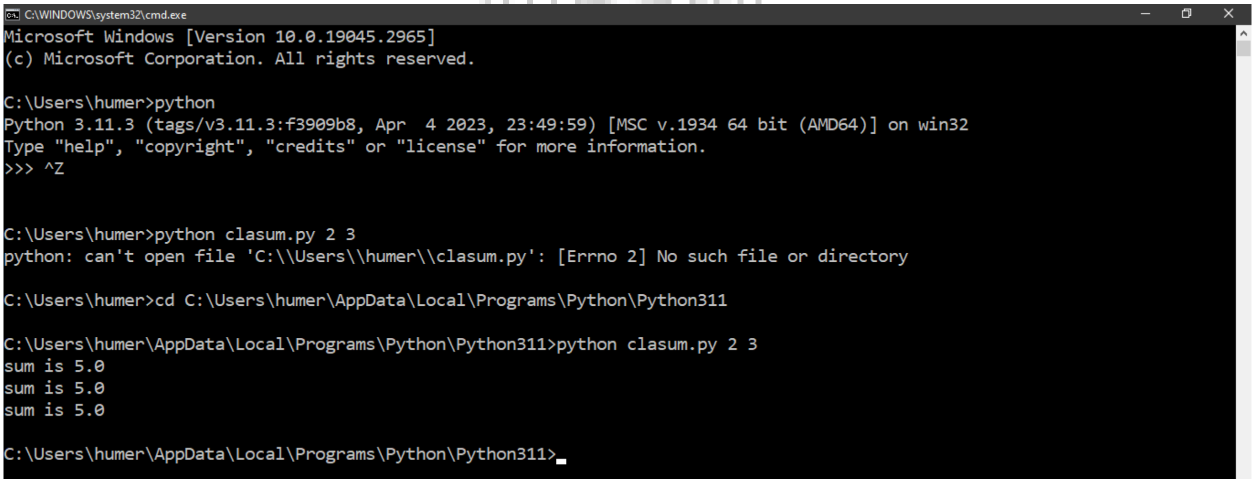
```
distance between (4, 3) and (2, 5) is :  3.1622776601683795

enter first coordinate : 4,2

enter second coordinate : 3,5

distance between  4,2 and 3,5 is 3.1622776601683795

Process finished with exit code 0
```

b) Write a program add python that takes 2 numbers as command line arguments and prints its sum.

```
import sys

a, b = sys.argv[1:3]

print("sum is", float(a) + float(b))

a, b = sys.argv[1:]

print("sum is", float(a) + float(b))

summ = sum(float(i) for i in sys.argv[1:])

print("sum is", summ)
```


clasum.py - C:/Users/humer/AppData/Local/Programs/Python/Python311/clasum.py (3.11.3)
File Edit Format Run Options Window Help

```
import sys
a, b = sys.argv[1:3]
print("sum is", float(a) + float(b))
a, b = sys.argv[1:]
print("sum is", float(a) + float(b))
summ = sum(float(i) for i in sys.argv[1:])
print("sum is", summ)
```



```
Microsoft Windows [Version 10.0.19045.2965]
(c) Microsoft Corporation. All rights reserved.

C:\Users\humer>python
Python 3.11.3 (tags/v3.11.3:f3909b8, Apr  4 2023, 23:49:59) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> ^Z


C:\Users\humer>python clasum.py 2 3
python: can't open file 'C:\\Users\\humer\\clasum.py': [Errno 2] No such file or directory

C:\Users\humer>cd C:\Users\humer\AppData\Local\Programs\Python\Python311

C:\Users\humer\AppData\Local\Programs\Python\Python311>python clasum.py 2 3
sum is 5.0
sum is 5.0
sum is 5.0

C:\Users\humer\AppData\Local\Programs\Python\Python311>
```

c) Program to display the following information: Your name, Full Address, Mobile Number, College Name, Course Subjects

```
#Simple implementation
n = input("Enter name of student: ")
c = input("Enter class of student: ")
a = int(input("Enter age of student: "))
print("Name:", n, "Class:", c, "Age:", a)
print()
print("Name:", n)
print("Class:", c)
print("Age:", a)
#Using Functions
def personal_details():
    name, age = "Simon", 19
    address = "Bangalore, Karnataka, India"
    print("Name: {}\nAge: {}\nAddress: {}".format(name, age, address))

personal_details()
```

Output

```
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\venv\Scripts\python.exe
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\14edetails.py

Enter name of student:

Arjun Enter class of student:

MCA Enter age of student:

20

Name: Arjun Class: MCA Age: 20


Name: Arjun

Class: MCA

Age: 20

Name: Simon
```

```
Age: 19

Address: Bangalore, Karnataka, India


Process finished with exit code 0
```

```
#Iimplementation using classes


class Student:
    marks = []

    def getData(self, rn, name, m1, m2, m3):
        Student.rn = rn
        Student.name = name
        Student.marks.append(m1)
        Student.marks.append(m2)
        Student.marks.append(m3)

    def displayData(self):
        print("Roll Number is: ", Student.rn)
        print("Name is: ", Student.name)
        # print ("Marks in subject 1: ", Student.marks[0])
        # print ("Marks in subject 2: ", Student.marks[1])
        # print ("Marks in subject 3: ", Student.marks[2])
        print("Marks are: ", Student.marks)
        print("Total    Marks    are:    ",    self.total())
        print("Average Marks are: ", self.average())

    def total(self):
        return (Student.marks[0] + Student.marks[1] + Student.marks[2])

    def average(self):
        return ((Student.marks[0] + Student.marks[1] + Student.marks[2]) / 3)


r = int(input("Enter the roll number: "))
name = input("Enter the name: ")
m1 = int(input("Enter the marks in the first subject: "))
m2 = int(input("Enter the marks in the second subject: "))
m3 = int(input("Enter the marks in the third subject: "))

s1 = Student()
s1.getData(r, name, m1, m2, m3)
s1.displayData()
```

Output

```
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\venv\Scripts\python.exe
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\14cdetailsclass.py

Enter the roll number: 01

Enter the name: Arjun

Enter the marks in the first subject: 90

Enter the marks in the second subject: 98

Enter the marks in the third subject: 92

Roll Number is:  1

Name is:  Arjun

Marks are:  [90, 98, 92]

Total Marks are:  280

Average Marks are:  93.33333333333333


Process finished with exit code 0
```

d) Write a Program for checking whether the given number is an even number or not.

```
# Python program to check if the input number is odd or even.
# A number is even if division by 2 gives a remainder of 0.
# If the remainder is 1, it is an odd number.
num = int(input("Enter a number: "))
if (num % 2) == 0:
   print("{0} is Even".format(num))
else:
   print("{0} is Odd".format(num))
```

Output

```
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\venv\Scripts\python.exe
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\14feven.py

Enter a number: 55

55 is Odd


Process finished with exit code 0


C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\venv\Scripts\python.exe
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\14feven.py

Enter a number: 12

12 is Even


Process finished with exit code 0
```

15. Control Structures, Lists

a) Program to find the largest three integers using if-else

```
a = int(input('Enter first number  : '))
b = int(input('Enter second number : '))
c = int(input('Enter third number  : '))

largest = 0

if a > b and a > c:
    largest = a
if b > a and b > c:
    largest = b
if c > a and c > b:
    largest = c
print("\nLargest using if statements")
print(largest, "is the largest of three numbers.")

largest = 0

if a > b and a > c :
    largest = a
elif b > c :
    largest = b
else :
    largest = c
print("\nLargest using if...elif statements")
print(largest, "is the largest of three numbers.")
```

Output

```
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\venv\Scripts\python.exe
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\15alargest.py
Enter first number  : 8
Enter second number : 12
Enter third number  : 19

Largest using if statements
19 is the largest of three numbers.
```

```
Largest using if...elif statements
19 is the largest of three numbers.

Process finished with exit code 0
```

b) Program that receives a series of positive numbers and display the numbers in order and their sum

```
mylist1=list(map(int,input("Enter elements of array\n"). split()))
mylist1.sort(reverse=False) #arr.sort() also be used.
print("Ascending order array")
print(mylist1)
mylist1.sort(reverse=True)
print("Descending order array")
print(mylist1)

# Using sum()
print("SUM Using sum(): ", sum(mylist1))

# Using sum() start from 50
print("SUM Using sum() after adding 50: ", sum(mylist1,50))

# Using List Comprehension with sum()
print("SUM Using List Comprehension with sum(): ", sum([i for i in mylist1]))

# Using for loop with range
total=0
for i in range(len(mylist1)):
    total=total+mylist1[i]
print("SUM Using for loop with range: ", total)

# Using for loop
total=0
for i in mylist1:
    total=total+i
print("SUM Using for loop: ", total)

# Using add() with for loop
from operator import add
total=0
for i in mylist1:
    total = add(i, total)
print("SUM Using add() with for loop: ", total)

# Using while loop
```

```
total=0
i=0
while (i < len(mylist1)):
    total=total+mylist1[i]
    i=i+1
print("SUM Using while loop: ", total)
```

Output

```
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\venv\Scripts\python.exe
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\15bordersum.py
Enter elements of array
8 9 5 4 6 7 2 1 3
Ascending order array
[1, 2, 3, 4, 5, 6, 7, 8, 9]
Descending order array
[9, 8, 7, 6, 5, 4, 3, 2, 1]
SUM Using sum():  45
SUM Using sum() after adding 50:  95
SUM Using List Comprehension with sum():  45
SUM Using for loop with range:  45
SUM Using for loop:  45
SUM Using add() with for loop:  45
SUM Using while loop:  45

Process finished with exit code 0
```

c) Program to find the product of two matrices

```
matOne = []
print("Enter 4 Elements for First Matrix: ")
for i in range(2):
    matOne.append([])
    for j in range(2):
        num = int(input())
        matOne[i].append(num)

matTwo = []
print("Enter 4 Elements for Second Matrix: ")
for i in range(2):
    matTwo.append([])
    for j in range(2):
        num = int(input())
        matTwo[i].append(num)

matThree = []
for i in range(2):
    matThree.append([])
    for j in range(2):
        sum = 0
        for k in range(2):
            sum = sum + (matOne[i][k] * matTwo[k][j])
        matThree[i].append(sum)

print("\nMultiplication Result of Two Given Matrix is:")
for i in range(2):
    for j in range(2):
        print(matThree[i][j], end=" ")
    print()
```

Output

```
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\venv\Scripts\python.exe
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\15cproductmatrices.py
Enter 4 Elements for First Matrix:
2
3
```

```
1
5
Enter 4 Elements for Second Matrix:
4
2
3
1
Multiplication Result of Two Given Matrix is:
17 7
19 7
Process finished with exit code 0
```

```python
# Multiply Two Matrices of Given Size
print("Enter the Row and Column Size of First Matrix: ", end="")
rOne = int(input())
cOne = int(input())
print("Enter " +str(rOne * cOne)+ " Elements: ", end="")
mOne = []
for i in range(rOne):
    mOne.append([])
    for j in range(cOne):
        num = int(input())
        mOne[i].append(num)

print("\nEnter Row and Column Size of Second Matrix: ", end="")
rTwo = int(input())
if cOne != rTwo:
    print("\nMultiplication Not Possible!")
    exit()
else:
    cTwo = int(input())
    print("Enter " +str(rTwo * cTwo)+ " Elements: ", end="")
    mTwo = []
    for i in range(rTwo):
        mTwo.append([])
        for j in range(cTwo):
            num = int(input())
            mTwo[i].append(num)

    # now multiplying two matrices
```

```
mThree = []
for i in range(rOne):
    mThree.append([])
    for j in range(cTwo):
        sum = 0
        for k in range(cOne):
            sum = sum + (mOne[i][k] * mTwo[k][j])
        mThree[i].append(sum)

# now printing the multiplication result
print("\nMultiplication Result of Two Given Matrix is:")
for i in range(rOne):
    for j in range(cTwo):
        print(mThree[i][j], end=" ")
    print()
```

Output

```
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\venv\Scripts\python.exe
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\15cproductmatrices2.py
Enter the Row and Column Size of First Matrix: 2
2
Enter 4 Elements: 2
2
3
4

Enter Row and Column Size of Second Matrix: 2
2
Enter 4 Elements: 3
5
6
2

Multiplication Result of Two Given Matrix is:
18 14
33 23

Process finished with exit code 0
```

d) Program to display two random numbers that are to be added, the program should allow the student to enter the answer.
   If the answer is correct, a message of congratulations should be displayed.
   If the answer is incorrect, the correct answer should be displayed.

```
import random
number = random.randint(1, 100)
attempts = 0  # count no of attempts to guess the number
guess = 0
attempts = 0
while attempts<3:
    guess = int(input('Guess a number between 1 and 100: '))
    attempts += 1
    if guess == number:
        print('Congratulations! You used', attempts, 'attempts!')
        break
    elif guess < number:
        print('Go higher!')
    else:
        print('Go lower!')
else:
    print('The number is ',number)
```

Output

```
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\venv\Scripts\python.exe
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\15drandom.py
Guess a number between 1 and 100: 56
Go higher!
Guess a number between 1 and 100: 95
Go lower!
Guess a number between 1 and 100: 80
Go higher!
The number is  89

Process finished with exit code 0
```

e) Using a for loop, write a program that prints out the decimal equivalents of 1/2,1/3,1/4, .1/10.

```
# initializing a variable with 1
p = 1

# iterating the for loop in a range from 2 to 10
for q in range(2, 11):
  # printing 1 (here p) divided by the current number and formatting it
    print(f"{p}/{q} :", p/q)
```

Ouput

```
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\venv\Scripts\python.exe
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\15edecimal.py
1/2 : 0.5
1/3 : 0.3333333333333333
1/4 : 0.25
1/5 : 0.2
1/6 : 0.16666666666666666
1/7 : 0.14285714285714285
1/8 : 0.125
1/9 : 0.1111111111111111
1/10 : 0.1

Process finished with exit code 0
```

f) Write a program using a while loop that asks the user for a number, and prints a countdown from that number to zero.

```
num=int(input("Enter your number "))
while(num>=0):
    print(num)
    num=num-1
```

Output

```
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\venv\Scripts\python.exe
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\15fcountdown.py
Enter your number 5
5
4
3
2
1
0

Process finished with exit code 0
```

16. Functions and Recursion - Write recursive and non-recursive functions for the following

a) To find GCD of two integers

#Using Recursion

```
# write a program to understand the GCD of two number in python using the recursion.
def gcd_fun (x, y):
    if (y == 0): # it divide every number
        return x  # return x
    else:
        return gcd_fun (y, x % y)
x =int (input ("Enter the first number: ")) # take first no.
y =int (input ("Enter the second number: ")) # take second no.
num = gcd_fun(x, y) # call the gcd_fun() to find the result
print("GCD of two number is: ", num)
```

Output

```
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\venv\Scripts\python.exe
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\16aGCDRecursion.py
Enter the first number: 5
Enter the second number: 20
GCD of two number is:  5

Process finished with exit code 0
```

```
#Using Iteration
def GCD_Loop( a, b):
    if a > b:  # define the if condition
        temp = b
    else:
        temp = a
    for i in range(1, temp + 1):
        if (( a % i == 0) and (b % i == 0 )):
            gcd = i
    return gcd
x = int(input (" Enter the first number: ") ) # take first no.
y =int (input (" Enter the second number: ")) # take second no.
num = GCD_Loop(x, y) # call the gcd_fun() to find the result
print("GCD of two number is: ",num)
```

Output

```
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\venv\Scripts\python.exe
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\16aGCDLoop.py
 Enter the first number: 5
Enter the second number: 20
GCD of two number is:  5

Process finished with exit code 0
```

b) To find the factorial of positive integer

#Using Recursion

```
# Python program to find the factorial of a number provided by the user using recursion
def factorial(x):
    """This is a recursive function
    to find the factorial of an integer"""
    if x == 1:
        return 1
    else:
        # recursive call to the function
        return (x * factorial(x-1))
# change the value for a different result
num = 7
# to take input from the user
# num = int(input("Enter a number: "))
# call the factorial function
result = factorial(num)
print("The factorial of", num, "is", result)
```

Output

```
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\venv\Scripts\python.exe
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\16afactorialrecursion.py
The factorial of 7 is 5040

Process finished with exit code 0
```

#Using Iteration

```
# Python program to find the factorial of a number provided by the user.
# change the value for a different result
#num = 7
# To take input from the user
num = int(input("Enter a number: "))
factorial = 1
# check if the number is negative, positive or zero
if num < 0:
   print("Sorry, factorial does not exist for negative numbers")
elif num == 0:
   print("The factorial of 0 is 1")
else:
   for i in range(1,num + 1):
       factorial = factorial*i
   print("The factorial of",num,"is",factorial)
```

Output

```
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\venv\Scripts\python.exe
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\16afactorialloops.py
Enter a number: 7
The factorial of 7 is 5040

Process finished with exit code 0
```

c) To print Fibonacci Sequence up to given number n

#Using Recursion

```
#Python program to generate Fibonacci series Program using Recursion
def Fibonacci_series(Number):
    if(Number == 0):
        return 0
    elif(Number == 1):
        return 1
    else:
        return (Fibonacci_series(Number - 2) + Fibonacci_series(Number - 1))
n = int(input("Enter the value of n: "))
print("Fibonacci Series:", end = ' ')
for n in range(0, n):
  print(Fibonacci_series(n), end = ' ')
```

Output

```
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\venv\Scripts\python.exe
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\16cfibrecuresion.py
Enter the value of n: 7
Fibonacci Series: 0 1 1 2 3 5 8
Process finished with exit code 0
```

#Using Iteration

```
#Python program to generate Fibonacci series until 'n' value
n = int(input("Enter the value of n: "))
a = 0
b = 1
sum = 0
count = 1
print("Fibonacci Series: ", end = " ")
while(count <= n):
  print(sum, end = " ")
  count += 1
  a = b
  b = sum
  sum = a + b
```

Otuput

```
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\venv\Scripts\python.exe
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\16cfibloops.py
Enter the value of n: 7
Fibonacci Series:  0 1 1 2 3 5 8
Process finished with exit code 0
```

d) To display prime number from 2 to n.

```
# Take input from user
upto = int(input("Find prime numbers upto : "))
print("\nAll prime numbers upto", upto, "are : ")
for num in range(2, upto + 1):
    i = 2
    for i in range(2, num):
        if(num % i == 0):
            i = num
            break;
    # If the number is prime then print it.
    if(i != num):
        print(num, end=" ")
```

Output

```
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\venv\Scripts\python.exe
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\16dprime.py
Find prime numbers upto : 50

All prime numbers upto 50 are :
3 5 7 11 13 17 19 23 29 31 37 41 43 47
Process finished with exit code 0
```

e) Function that accepts two arguments: a list and a number n. It displays all of the numbers in the list that are greater than n

```
#call the main function
def main():
    #declares local variables
    number = 5
    number_list = [8,9,2,1,4,5,6,7,10,3]
    #displays the number
    print('Number ', number)
    #displays the list of numbers
    print('List of numbers: ', number_list, sep='')
    #Display the list of numbers that are larger
    #than the number
    print('List of numbers that are larger than ',number)
    #Call the larger_than_n_list function,
    #passing a number and number list as arguments.
    display_larger_than_n_list(number, number_list)
# The display_largger_than_n_list function acceps two arguments:
#a list, and a number. The function displays all of the numbers
#in the list that are greater than that number.
def display_larger_than_n_list(n, n_list):
    #Declare an empty list
    larger_than_n_list = []
    #Loop through the values in the list.
    for valie in n_list:
        #Determins if a value is greatter than n.
        #if so, append the value to the list
        if valie > n:
            larger_than_n_list.append(valie)
    #Display the list.
    print("larger_than_n_list",larger_than_n_list)
#Call the main function
main()
```

Output

```
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\venv\Scripts\python.exe
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\16efunclistnum.py
Number  5
```

```
List of numbers: [8, 9, 2, 1, 4, 5, 6, 7, 10, 3]
List of numbers that are larger than  5
larger_than_n_list [8, 9, 6, 7, 10]

Process finished with exit code 0
```

f) Functions that accept a string as an argument and return the number of vowels and consonants that the string contains

```
# Python3 code to count vowel in a string using set
# Function to count vowel
def vowel_count(str):
    # Initializing count variable to 0
    count = 0
    consonents = 0
    #misc = 0
    # Creating a set of vowels
    vowel = set("aeiouAEIOU")
    consonentslist = set("bcdfghjklmnpqrstvwxyzBCDFGHJKLMNPQRSTVWXYZ")
    # Loop to traverse the alphabet
    # in the given string
    for alphabet in str:
        # If alphabet is present
        # in set vowel
        if alphabet in vowel:
            count = count + 1
        elif alphabet in consonentslist:
            consonents = consonents + 1
        #else:
        #    misc = misc + 1
    print("No. of vowels :", count)
    print("No. of consonents :", consonents)
    #print("No. of other symbols :", misc)
# Driver code
str = "Aurora College"
# Function Call
vowel_count(str)
```

Output

```
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\venv\Scripts\python.exe
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\16ffunstrvowcon.py
No. of vowels : 5
No. of consonents : 7

Process finished with exit code 0
```

17. Files, Exceptions, Lists, Sets, Random Numbers

a) Program to write a series of random numbers in a file from 1 to n and display.

```
# import the required modules
import random
# define the main function
def main():
    # prompt the user to enter the numbers
    n = int(input("Please enter the number of random numbers to be hold in a file: "))
    # create a new text file
    outfile = open("randomNumbers.txt", "w")
    # write numbers in the text file
    for i in range(n):
        outfile.write(str(random.randint(1, 500)))
        outfile.write(' ')
    outfile.close()
    infile = open("randomNumbers.txt", "r")
    for i in range(n):
        ran = infile.read()
        print(ran,end=' ')
    # close the file
    infile.close()
# call the main function
main()
```

Output

```
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\venv\Scripts\python.exe
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\17afilerandom.py
Please enter the number of random numbers to be hold in a file: 10
489 473 199 411 318 398 173 495 68 388
Process finished with exit code 0
```

b) Program to write the content in a file and display it with a line number followed by a colon

```
# define the main function
def main():
    # prompt the user to enter the numbers
    n = int(input("Please enter the number of lines to be entered in a file: "))
    # create a new text file
    outfile = open("linenos.txt", "w")
    # write numbers in the text file
    for i in range(n):
        line=input("Enter the line ")
        outfile.write(line)
        outfile.write('\n')
    outfile.close()
    infile = open("linenos.txt", "r")
    for i in range(n):
        line = infile.readline()
        print(i+1,': ',line,end=")
    # close the file
    infile.close()
# call the main function
main()
```

Output

```
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\venv\Scripts\python.exe
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\17bfilelineno.py
Please enter the number of lines to be entered in a file: 3
Enter the line hi
Enter the line hru
Enter the line bye
1 :  hi
2 :  hru
3 :  bye

Process finished with exit code 0
```

c) Program to display a list of all unique words in a text file

```
outfile = open("unique.txt", "w")
# write in the text file
text = "This is a text file containing words. Few words are repeated while few others are unique"
outfile.write(text)
outfile.close()
text_file = open('unique.txt', 'r')
text = text_file.read()
#cleaning
text = text.lower()
words = text.split()
words = [word.strip('.,!;()[]') for word in words]
words = [word.replace("'s", '') for word in words]
#finding unique
unique = []
for word in words:
    if word not in unique:
        unique.append(word)
#sort
unique.sort()
#print
print(unique)
```

Output

```
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\venv\Scripts\python.exe
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\17cfileuniquewords.py
['a', 'are', 'containing', 'few', 'file', 'is', 'others', 'repeated', 'text', 'this', 'unique', 'while', 'words']

Process finished with exit code 0
```

d) Program to analyze the two text files using set operations

```
file1 = open("fileset1.txt", "w")
# write in the text file
text1 = "This is a text file containing words which will be analyzed using sets"
file1.write(text1)
file1.close()
file2 = open("fileset2.txt", "w")
# write in the text file
text2 = "This is a text file containing words that will be analyzed using sets"
file2.write(text2)
file2.close()
file3 = open("fileset3.txt", "w")
# write in the text file
text3 = "This is a text file containing words that will be analyzed using sets"
file3.write(text2)
file3.close()

set1 =  set(open('fileset1.txt').read().split())
set2 =  set(open('fileset2.txt').read().split())
set3 =  set(open('fileset3.txt').read().split())

if set1 == set2:
    print("Contents of File1 and File2 are same")
elif set1 == set3:
    print("Contents of File1 and File3 are same")
elif set2 == set3:
    print("Contents of File2 and File3 are same")
```

Output

```
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\venv\Scripts\python.exe
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\17dfileset.py

Contents of File2 and File3 are same

Process finished with exit code 0
```

e) Write a program to print each line of a file in reverse order.

```
file = open("filereverse.txt", "w")
# write numbers in the text file
contents1 = "This is first line\nThis is second line\nThis is last line"
file.write(contents1)
file.close()
# Open the input file again and get # the content as list to a variable data
with open("filereverse.txt", "r") as myfile:
    contents2 = myfile.readlines()
print("Before Reverse", contents2)
# We will just reverse the # array using following code
contents2 = contents2[::-1]
print("After Reverse", contents2)
```

Output

```
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\venv\Scripts\python.exe
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\17efilereverse.py

Before Reverse ['This is first line\n', 'This is second line\n', 'This is last line']
After Reverse ['This is last line', 'This is second line\n', 'This is first line\n']

Process finished with exit code 0
```

f) Write a program to count frequency of characters in a given file. Can you use character frequency total whether the given file is a Python program file, C program file or a text file?

```
file1 = open("freqcount.txt", "w")
# write numbers in the text file
contents1 = "This is text file"
file1.write(contents1)
file1.close()
file2 = open("freqcount.c", "w")
# write numbers in the c file
contents2 = "int main"
file2.write(contents2)
file2.close()
file3 = open("freqcount.py", "w")
# write numbers in the python file
contents3 = "print(hello)"
file3.write(contents3)
file3.close()
file1 = open("freqcount.txt", "r")
a=[]
b={}
for i in file1:
    for j in range(0,len(i)):
        a.append(i[j])
for i in a:
    if i in b:
        b[i]+=1
    else:
        b[i]=1
print(b)
c=file1.name.split(".")
if c[1]=="txt":
    print("It is a text file")
elif c[1]=="c":
    print("It is a c file")
else:
    print("It is a python file")
file1.close()
file3 = open("freqcount.py", "r")
```

```
a=[]
b={}
for i in file3:
    for j in range(0,len(i)):
        a.append(i[j])
for i in a:
    if i in b:
        b[i]+=1
    else:
        b[i]=1
print(b)
c=file3.name.split(".")
if c[1]=="txt":
    print("It is a text file")
elif c[1]=="c":
    print("It is a c file")
else:
    print("It is a python file")
file3.close()
```

Output

```
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\venv\Scripts\python.exe
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\17ffilefreqcount.py

{'T': 1, 'h': 1, 'i': 3, 's': 2, ' ': 3, 't': 2, 'e': 2, 'x': 1, 'f': 1, 'l': 1}
It is a text file
{'p': 1, 'r': 1, 'i': 1, 'n': 1, 't': 1, '(': 1, 'h': 1, 'e': 1, 'l': 2, 'o': 1, ')': 1}
It is a python file

Process finished with exit code 0
```

g) Write a program combine lists that combines these lists in to a dictionary.

```
def test(keys, values):
  return dict(zip(keys, values))

l1 = ['a', 'b', 'c', 'd', 'e', 'f']
l2 = [1, 2, 3, 4, 5]
print("Original lists:")
print(l1)
print(l2)
print("\nCombine the values of the said two lists into a dictionary:")
print(test(l1, l2))
```

Output

```
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\venv\Scripts\python.exe
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\17gliststodict.py
Original lists:
['a', 'b', 'c', 'd', 'e', 'f']
[1, 2, 3, 4, 5]

Combine the values of the said two lists into a dictionary:
{'a': 1, 'b': 2, 'c': 3, 'd': 4, 'e': 5}

Process finished with exit code 0
```

18. Object Oriented Programming

a) Program to implement the inheritance

```
#Inheritance
class Person:
    def __init__(self, firstname='Arjun', lastname='Paul', age=25, country='India',
city='Hyderabad'):
        self.firstname = firstname
        self.lastname = lastname
        self.age = age
        self.country = country
        self.city = city
        self.skills = []

    def person_info(self):
      return f'{self.firstname} {self.lastname} is {self.age} years old. He lives in {self.city},
{self.country}.'
    def add_skill(self, skill):
        self.skills.append(skill)

p1 = Person()
print(p1.person_info())
p1.add_skill('HTML')
p1.add_skill('CSS')
p1.add_skill('JavaScript')
p2 = Person('Tom', 'Jerry', 30, 'India', 'Banglore')
print(p2.person_info())
print(p1.skills)
print(p2.skills)

class Student(Person):
   pass
s1 = Student('Emm', 'Wha', 30, 'India', 'Pune')
s2 = Student('Bob', 'Trudy', 28, 'India', 'Mumbai')
print(s1.person_info())
s1.add_skill('JavaScript')
s1.add_skill('React')
s1.add_skill('Python')
print(s1.skills)
print(s2.person_info())
s2.add_skill('Organizing')
s2.add_skill('Marketing')
s2.add_skill('Digital Marketing')
print(s2.skills)
#Overriding parent method

class Student(Person):
    def __init__ (self, firstname='Arjun', lastname='Paul', age=25, country='India',
city='Hyderabad', gender='male'):
        self.gender = gender
        super().__init__(firstname, lastname,age, country, city)

    def person_info(self):
        gender = 'He' if self.gender =='male' else 'She'
        return f'{self.firstname} {self.lastname} is {self.age} years old. {gender} lives in
{self.city}, {self.country}.'

s1 = Student('Alice', 'Smiley', 30, 'India', 'Pune','female')
s2 = Student('Bob', 'Trudy', 28, 'India', 'Mumbai','male')
print(s1.person_info())
s1.add_skill('JavaScript')
s1.add_skill('React')
s1.add_skill('Python')
print(s1.skills)

print(s2.person_info())
s2.add_skill('Organizing')
s2.add_skill('Marketing')
s2.add_skill('Digital Marketing')
print(s2.skills)
```

Output

```
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\venv\Scripts\python.exe
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\18ainheritance.py
Arjun Paul is 25 years old. He lives in Hyderabad, India.
Tom Jerry is 30 years old. He lives in Banglore, India.
['HTML', 'CSS', 'JavaScript']
[]
```

```
Emm Wha is 30 years old. He lives in Pune, India.
['JavaScript', 'React', 'Python']
Bob Trudy is 28 years old. He lives in Mumbai, India.
['Organizing', 'Marketing', 'Digital Marketing']
Alice Smiley is 30 years old. She lives in Pune, India.
['JavaScript', 'React', 'Python']
Bob Trudy is 28 years old. He lives in Mumbai, India.
['Organizing', 'Marketing', 'Digital Marketing']

Process finished with exit code 0
```

b) Program to implement the polymorphism

```
class Vehicle:
  def __init__(self, brand, model):
    self.brand = brand
    self.model = model

  def move(self):
    print("Move!")

class Car(Vehicle):
  pass

class Boat(Vehicle):
  def move(self):
    print("Sail!")

class Plane(Vehicle):
  def move(self):
    print("Fly!")

car1 = Car("Ford", "Mustang") #Create a Car object
boat1 = Boat("Ibiza", "Touring 20") #Create a Boat object
plane1 = Plane("Boeing", "747") #Create a Plane object

for x in (car1, boat1, plane1):
  print(x.brand)
  print(x.model)
  x.move()
```

Output

```
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\venv\Scripts\python.exe
C:\Users\humer\PycharmProjects\Python-MCA-DEP-2025\18bpolymorphism.py
Ford
Mustang
Move!
Ibiza
Touring 20
Sail!
```

```
Boeing
747
Fly!

Process finished with exit code 0
```

19. Demonstrate data analysis using NumPy
    a) Create an array of 10 zeros
    b) Create an array of even integers upto 50
    c) Create a 3x3 matrix
    d) Generate an array of 25 random numbers sampled from a standard normal distribution.
    e) Create an array of 20 linearly spaced points between 0 and 1
    f) Demonstrate slicing and indexing operations
    g) Get the sum of all columns in matrix
20. Write a Program in Python to create and combine student and subject data frames in Pandas.
21. Create a data frame ‘Book’ that contains three vectors [Name, Price, Author]. Convert this data frame into a matrix and list the object using the operator ‘as’.
22. Performing Exploratory data analysis on web scraped data of 2021-22 NBA player stats (http://www.basketball-reference.com/)
    a. Perform data cleaning
    b. Handle missing values by replacing with 0
    c. Write to CSV file
    d. Which player scored the most points per game?
    e. Which player had the highest 3-point field goals per game?
    f. Demonstrate Group By() function
23. Data visualization through Sea born for the above program 9.
    a. Box plot of points scored grouped by position
    b. Compute the correlation matrix
24. To determine the mean of a set of numbers. To plot the numbers in a bar plot and have a straight line run through the plot at the mean.
25. To determine the median of a set of numbers. To plot the numbers in a barplot and have a straight line runs through the plot at the median.

To determine the standard deviation. To plot the numbers in a bar plot and have a straight line run through the plot at the mean and another straight line run through the plot at mean + standard deviation.

Lab Manual
Database Management Systems

1. OBJECTIVE AND RELEVANCE DATABASE MANAGEMENT SYSTEMS:

Students will have the ability to:

- Keep abreast of current developments to continue their own professional development.
- To engage themselves in lifelong learning of Database management systems theories and technologies this enables them to purse higher studies.
- To interact professionally with colleagues or clients located abroad and the ability to overcome challenges that arises from geographic distance, cultural differences, and multiple languages in the context of computing.
- Develop team spirit, effective work habits, and professional attitude in written and oral forms, towards the development of database applications

2. OUTCOMES

DATABASE MANAGEMENT SYSTEMS:

Students will be able to demonstrate their skills

- In drawing the ER, EER, and UML Diagrams.
- In analyzing the business requirements and producing a viable model for the implementation of the database.
- In converting the entity-relationship diagrams into relational tables.
- To develop appropriate Databases to a given problem that integrates ethical, social, legal, and economic concerns.

3. EQUIPMENT REQUIRED

Hardware

No. of System	:	60(IBM)
Processor	:	PIV™ 1.67 GH$_z$
RAM	:	512 MB
Hard Disk	:	40 GB
Mouse	:	Optical
MouseNetwork Interface card	:	Present

Software

Operating System	:	Window XP
Software	:	Oracle

4. CODE OF CONDUCT

- Students should report to the concerned lab as per the time table.

- Students who turn up late to the labs will in no case be permitted to do the program schedule for the day.

- After completion of the program, certification of the concerned staff in-charge in the observation book is necessary.

- Student should bring a notebook of 100 pages and should enter the readings/observations into the notebook while performing the experiment.

- The record of observations along with the detailed experimental procedure of the EXPERIMENT in the immediate last session should be submitted and certified staff member in-charge.

- Not more than 2-students in a group are permitted to perform the EXPERIMENT on the set.

- Any damage of the systems will be viewed seriously either by putting penalty or by dismissing the total group of students from the lab for the semester/year.

- Students should be present in the labs for total scheduled duration

5. SYLLABUS ANALYSIS

S.No.	Name of the Experiment	Unit No.	Text / Reference Book	Page Nos
Database Management Systems				
1	Implementation of DDL commands of SQL CREATE ALTER TRUNCATE RENAME DROP Implementation of Constraints on Tables NOT NULL UNIQUE PRIMARY KEY CHECK FOREIGN KEY ●	Unit 1	T1, T2, R1	
2	Implementation of DML commands INSERT UPDATE DELETE Implementation of DRL commands SELECT Implementation of Joins Equi Join Non-equi Join Outer JoinSelf Join Implementation of Functions Numeric	Unit-2	T1, T2, R1	

	Date String			
3	PL/SQL Programming SIMPLE PROGRAMS PROCEDURES TRIGGERS CURSORS	Unit-2	T1, T2, R1	
4	Implementation of DTL Commands COMMIT ROLLBACK SAVEPOINT	Unit-4	T1, T2, R1	
5	Implemenation of DCL Commands Creating a USER GRANT REVOKE Implementation of FORMS Implementation of REPORTS	Unit-5	T1, T2, R1	

6. List of text book (OU Prescribed Text Books) DATABASE MANAGEMENT SYSTEMS

TEXT BOOKS

T1: Raghu Ramakrishnan, Johannes Gehrke, "*Database Management Systems*", 3rd Edition,McGraw Hill, 2003.

T2: Abraham Silberschatz, Henry F Korth, S Sudharshan, "*Database System Concepts*", 6thEdition, McGraw-Hill International Edition, 2011

REFERENCE BOOKS

R1: Ramez Elmasri, Durvasul VLN Somayajulu, Shamkant B Navathe,. Shyam K Gupta, "*Fundamentals of Database Systems*", 6th Edition, Addison Wesley, 2011.

R2: Peter Rob, Carlos Coronel, "*Database System Concepts*", Cengage Learning,2008

7. SESSION PLAN

S. No.	Veek No.	Unit as per Syllabus	Activity	Remarks
DATABASE MANAGEMENT SYSTEM				
1	1	UNIT-1	Introduction to Oracle	PREREQUISITE
2	1	UNIT-1	Implementation of DDL Commands.	
3	2	UNIT-1	Implementation of constraints on table.	
4	3	UNIT-2	Implementation of DML Commands	
5	4	UNIT-2	Implementation of DRL Commands	
6	5	UNIT-2	Implementation of Joins	
7	6	UNIT-2	Implementation of Functions-Date, Numeric and String.	
8	6	UNIT-2	Implementation of Inner Queries	
10	7	UNIT-2	Implementation of Group Functions & Group By Clause	
11	8	UNIT-4	Implementation of DTL Commands	
12	8	UNIT-4	Implementation of Locking Mechanism	
13	9	UNIT-2	Implementation of Basic PL/SQL Programs	
14	10	UNIT-2	Implementation of Procedures and Functions	
15	11	UNIT-2	Implementation of VIEWS	
16	12	UNIT-2	Implementation of Cursors	
17	13	UNIT-2	Implementation of Triggers	
18	14	UNIT-5	Implementation of DCL Commands	
19	15	UNIT-5	Implementation of Forms	
20	15	UNIT-5	Implementation of Reports	

1. IMPLEMENTATION OF DDL COMMANDS

AIM: To create a DDL to perform creation of table, alter, modify, drop column.

DDL COMMANDS

1. **CREATION OF TABLE:** Create table command is used to create a new table and definethe data types and the size.

Syntax:

```
create table<tablename>(
column1 datatype,
column2 datatype...
);
```

Example:

```
SQL>create table std(sno number(5),sname varchar(20),age number(5),sdob
date,sm1 number(4,2),sm2 number(4,2),sm3 number(4,4));

Table created.

SQL>insert into std values(101,‟AAA‟,16,‟03-jul-
88‟,80,90,98);1 row created.

SQL>insert into std values(102,‟BBB‟,18,‟04-aug-
89‟,88,98,90);1 row created.
```

OUTPUT:

```
Select * from std;

SNO   SNAME   AGE   SDOBSM1        SM2   SM3

101   AAA     16    03-jul-88 80   90    98

102   BBB     18    04-aug-89 88   98    90
```

2. **ALTER TABLE :** Alter table command is used to do the following

a. Add a new column to the existing table
b. Modify the data type
c. Modify the size
d. Enable or disable the constraints
e. Add new constraints to the existing table
f. Drop a column

a. ALTER TABLE
WITH ADDSyntax:
Alter table<tablename>add(col1 datatype,col2 datatype..);

```
SQL>create table student(id number(5),name varchar(10),game
varchar(20));Table created.

SQL>insert into student
values(1,‟mercy‟,‟cricket‟);1 row created.
```

OUTPUT:

```
SQL>select * from

student; ID NAME

GAME

1       Mercy Cricket

SQL>alter table student add(age
number(4));Table Altered

SQL>insert into student
values(2,‟sharmi‟,‟tennis‟,19);1 row created
```

OUTPUT after Altering the table :

```
SQL>select * from student;

 ID    NAME       GAME        AGE

 1     Mercy      cricket

 2     Sharmi     Tennis      19
```

b. ALTER TABLE WITH MODIFY:

Syntax:
Alter table<tablename>modify(col1 datatype,col2 datatype..);

Example:

```
      SQL>alter table student modify(id number(6),game

varchar(25));SQL>desc student;
```

NAME NULL? TYPE

Id Number(6)

Name

Varchar(20)

Game

Varchar(25)

Age

Number(4)

3. **DROP TABLE :** This command is used to remove the table permanently from the table.

   Syntax: drop table<tablename>;

   Example:

   SQL>drop table
   student; Table
   dropped.

4. **TRUNCATE TABLE :** Truncate table deletes the records from the table but the structureof the table will remain in the database.

   Syntax: TRUNCATE TABLE <TABLE NAME>;

   Example:
   SQL>Truncate table
   stud;Table trucated

5. **RENAME TABLE:** Rename table is used to change the name of the table.

   Syntax:
   Rename table <old table name> to <new table name>;

   Example:
   SQL>Rename table stud to
   student;Table renamed

VIVA QUESTIONS

1. Defin e DDL
2. Categories of SQL Command.
3. Difference between truncate and drop.
4. Uses of alter command.

2 - IMPLEMENTATION OF CONSTRAINTS ON TABLE

AIM: To apply constraints to the table.

Create table tablename (column_name1 data_ type constraints, column_name2 data_type constraints …)

Example:

Create table Emp (EmpNo number(5), EName VarChar(15), Job Char(10) constraint un unique, DeptNo number(3) CONSTRAINT FKey2 REFERENCES DEPT(DeptNo));

Create table stud (sname varchar2(20) not null, rollno number(10) not null,dob date notnull);

DOMAIN INTEGRITY

Example: Create table cust(custid number(6) not null, name

char(10));Alter table cust modify (name not null);

CHECK CONSTRAINT

Example: Create table student (regno number (6), mark number (3) constraint b check(mark >=0 and mark <=100)); Alter table student add constraint b2 check (length(regno<=4));

ENTITY INTEGRITY

a) **Unique key constraint**

SQL> Create table cust(custid number(6) constraint unique, name char(10)); Alter tablecust add(constraint c unique(custid));

b) **Primary Key Constraint**

SQL> Create table stud(regno number(6) constraint primary key, name char(20));

Practice Questions

Q1. Create a table called EMP with the following

structure.NameType
---------- ---------------

------- EMPNO
NU
MBER(6)
ENAME VARCHAR2(20)

JOB
VARCH
AR2(10) DEPTNO
NUMBE
R(3) SAL
NUMBE
R(7,2)

Allow NULL for all columns except ename

and job.Solution:

1. Understand create table Syntax.

2. Use the create table Syntax to create the said tables.

3. Create primary key constraint for each table as understand from logical tablestructure. Ans:

SQL> create table emp(empno number(6),ename varchar2(20)not null,job varchar2(10)not null, deptno number(3),sal number(7,2));

Table created.

Q2: Add a column experience to the emp table.Experience numeric null

allowed.Solution:

1. Learn alter table Syntax.

2. Define the new column and its data type.

3. Use the alter table Syntax.

Ans: SQL> alter table emp add(experience number(2));Table altered.

Q3: Modify the column width of the job field of emp table.

Solution:

1. Use the alter table Syntax.

2. Modify the column width and its data type.

Ans: SQL> alter table emp modify(job varchar2(12));Table altered.

SQL> alter table emp modify(job varchar(13));Table altered.

Q4: Create dept table with the following structure.

Name Type
----------- ------------------
-- DEPTNO
NUMBER(2)
DNAME
VARCHAR2(1
0) LOC
VARCHAR2(1
0)

Deptno as the

primarykeySolution:

1. Understand create table Syntax.

2. Decide the name of the table.

3. Decide the name of each column and its data type.

4. Use the create table Syntax to create the said tables.

5. Create primary key constraint for each table as understand from logical tablestructure.

SQL> create table dept(deptno number(2) primary key,dnamevarchar2(10),loc varchar2(10));

Table created.

Q5: create the emp1 table with ename and empno, add constraints to

check theempno value while entering (i.e) empno > 100.

Solution:

1. Learn alter table Syntax.
2. Define the new constraint [columns name type]
3. Use the alter table Syntax for adding constraints.

```
SQL> create table emp1(ename varchar2(10),empno number(6)

constraintcheck(empno>100));

Table created.
```

Q6: drop a column experience to the emp table.

Solution:

1. Learn alter table Syntax. Use the alter table Syntax to drop the column.

```
SQL> alter table emp drop column experience; Table

altered.
```

Q7: Truncate the emp table and drop the dept table Solution:

1. Learn drop, truncate table Syntax.

```
SQL> truncate table
emp;            Table
truncated.

SQL> drop table
dept;
Table dropped.
```

VIVA QUESTIONS

1. What are constraints?
2. What is Primary key?
3. What is candidate key?

4. What is super key?
5. What is foreign key?
6. Difference between unique and not null.

3 - IMPLEMENTATION OF DML COMMANDS

AIM:To study the various DML commands and implement them on the database.

DML commands are the most frequently used SQL commands and is used to query and manipulate the existing database objects. Some of the commands are Insert, Update, Delete.

Insert Command: This is used to add one or more rows to a table. The values are separated by commas and the data types char and date are enclosed in apostrophes. Thevalues must be entered in the same order as they are defined.

Syntax:
Insert into <tablename> values((&numberval1,'&stringval','&stringval2',&numericval2);

Update Command: It is used to alter the column values in a table. A single column maybe updated or more than one column could be updated.

Syntax:
Update <tablename> set columnname=new_column_value where <condition>;

Delete command: After inserting row in a table we can also delete them if required. The delete command consists of a from clause followed by an optional where clause.

Syntax:
Delete from tablename where <condition>;

Q1: Insert a single record into dept table.

Ans: SQL> insert into dept values

(1,'IT','Tholudur');1 row created.

Q2: Insert more than a record into emp table using a single insert command.

Ans: SQL> insert into emp

values(&empno,'&ename','&job',&deptno,&sal);Enter value for empno: 1

Enter value for ename:

Mathi Enter value for

```
job: AP

Enter value for

deptno: 1 Enter

value for sal: 10000

old 1: insert into emp
values(&empno,'&ename','&job',&deptno,&sal) new 1: insert
into emp values(1,'Mathi','AP',1,10000)

1 row created.

SQL> / Enter value for

empno: 2 Enter value for

ename: Arjun Enter value

for job: ASP

Enter value for

deptno: 2 Enter

value for sal: 12000

old 1: insert into emp

values(&empno,'&ename','&job',&deptno,&sal) new 1: insert

into emp values(2,'Arjun','ASP',2,12000)

1 row created.

SQL> / Enter value for

empno: 3 Enter value for

ename: Gugan Enter value

for job: ASP

Enter value for

deptno: 1 Enter

value for sal: 12000
```

```
old 1: insert into emp
values(&empno,'&ename','&job',&deptno,&sal) new 1: insert
into emp values(3,'Gugan','ASP',1,12000)
1 row created.
```

Q3: Update the emp table to set the salary of all employees to Rs15000/- who areworking as ASP

```
SQL> select * from emp;

EMPNO ENAME JOB DEPTNO SAL

---------- ------------------- ------------- ---------- ----------

  1 Mathi AP 1    10000

  2 Arjun ASP     2 12000
  3 Gugan ASP 1 12000

SQL> update emp set sal=15000 where job='ASP'; 2 rows

updated.SQL> select * from emp;

EMPNO ENAME JOB DEPTNO SAL

 ---------- ------------------- ------------- ---------- ---

-------1 Mathi AP 1 10000

2 Arjun ASP 2 15000

3 Gugan ASP 1 15000
```

Q4: Create a pseudo table employee with the same structure as the table emp andinsert rows into the table using select clauses.

```
SQL> create table employee as select * from

emp;Table created.

SQL>         desc

employee;

Name       Null?
```

```
Type

-------------------------------------- -------- --------------------

-------EMPNO NUMBER(6)

ENAME          NOT          NULL

VARCHAR2(20)     JOB      NOT

NULL              VARCHAR2(13)

DEPTNO NUMBER(3)

SAL NUMBER(7,2)
```

Q5: Delete only those who are working as lecturer

```
SQL> select * from emp;
EMPNO ENAME JOB DEPTNO SAL
---------- ------------------- ------------ ---------- ----------

1 Mathi AP 1 10000
2 Arjun ASP 2 15000
3 Gugan ASP 1 15000
4 Karthik Prof 2 30000
5 Akalya AP 1 10000
6 suresh lect 1 8000
6 rows selected.

SQL>   delete   from   emp   where
job='lect';1 row deleted.

SQL> select * from emp;
EMPNO ENAME JOB DEPTNO SAL
-                    --------- ------------------- ------------ ---------- ----------
1 Mathi AP 1 10000
2 Arjun ASP 2 15000
3 Gugan ASP 1 15000
4 Karthik Prof 2 30000
5 Akalya AP 1 10000
```

VIVA QUESTIONS

1. What is update command in sql?
2. Difference between delete and drop
3. Difference between delete , drop and truncate.
4. Insert in sql.

4 - IMPLEMENTATION OF DRL COMMAND

AIM: To study the DRL command and implement it on the database

Date Retrievel Command – SELECT Command: It is used to retrieve information from the table. It is generally referred to as querying the table. We can either display all columns in a table or only specify column from the table.

Syntax:
Select */Column_list from table_name where <condition> order by column_name group by column_name having <group condition>; Queries for practice:

1. **Display all the tables in the database**
   SQL>select * from tab;

2. **Display employee name, job from the emp table**

   SQL> select ename, job from emp;

3. **Display unique department in the emp table.**

   SQL> select distinct dname from emp;

4. **Display the employee names who are working in dept number 10**

   SQL>select ename from emp where dno=10;

5. **Display the employee names who are working in dept number 10 and in salesdepartment.**

   SQL>select ename from emp where dno=10 and dname='sales';

6. **Display the employee names who are working in dept number 10 or in salesdepartment.**

   SQL>select ename from emp where dno=10 and dname='sales';

7. **Display the names of the employees whose salary is between 2000 to 10000**

   SQL>select ename from emp where sal between 2000 and 10000;

8. **Display the names of the employess who are working in department number 10,20 or 30.**

   SQL> select ename from emp where dno in (10,20,30);

9. **Display the names of the employees whose name start with "S".**

   SQL>select ename from emp where ename like "S%";

10. **Display the names of the employees whose name has "LL".**

    SQL>select ename from emp where ename like "%LL%";

11. **Display the names of the employess who are not working in department number10, 20 or 30.**

    SQL> select ename from emp where dno not in (10,20,30);

12. Display ename, sal, doj of the employees with sal in ascending order. SQL>select ename, sal , doj from emp order by sal asc;

13. Display ename, sal, doj of the employees with doj in descending order. SQL>select ename, sal , doj from emp order by doj desc;

14. **Display employee name, job from the emp table**

    SQL> select ename, job from emp;

15. **List the records in the emp table orderby salary in ascending order.**

    SQL> select * from emp order by sal;

VIVA QUESTIONS

1. What is order by clause?
2. IN & NOT IN operator

3. What is LIKE predicate
4. What is Between – AND operator
5. What are SQL Aliases?

5 - IMPLEMENTATION OF JOINS

AIM : To display the date from more than one table using joins.

A JOIN clause is used to combine rows from two or more tables, based on a related columnbetween them.

Guidelines to use join:

- Write the join condition in the where clause
- Prefix the column name with the table name when the same column names appear in more than one table.
- To join 'n' tables we need a minimum of (n-1) join conditions.
- Use a join to query data from more than one table.

Syntax:

Select table1. Column, table2.column
From table1, table2
Where
table1.column1=table2.column2; Types of joins:

1. **Equi Join :** A relationship between two tables is an equi join, when both the tables have common columns and are equal. Equi joins are also called as simple joins or inner joins.

Display the empno, ename , deptno and location from emp and dept tables.

```
SQL> select emp.empno, emp.ename, emp.deptno, dept.location
     from emp, dept
     where emp.empno=dept.deptno;
```

Display the customer name, ordered , item id , item total, order total from the belowtables.
Customer(name,custid)
Order(custid,ordid,total)
Item(ordid,itemid,itemtot)

```
SQL>  select  c.name,o.ordid,  i.itemid,  i.itemtot,  o.total
     From customer c, order o, item i
     Where c.custid=o.custid and o.ordid=i.ordid;
```

2. **Non-equi join:** The relationship between the two tables is a non-equi join, when wefind no common columns in both the tables.

Display employee name , salary and salary grade.

```
SQL> select e.ename, e.sal,
e.gradeFrom emp e, salgrade
s
Where e.sal between s.losal and s.hisal;
```

3. **Outer Join:** If a row does not satisfy a join condition, the row will not appear in the query result. We use outer join to see rows that do not usually meet the join condition.Outer join operator is the plus sign(+)

   Syntax:
   Select table1.column, table2.column

From table1, table2
Where
table1.column(+)=table2.column;
Here table1 is deficient of information Select table1.column, table2.column From table1, table2
Where
table1.column=table2.column(+);
Here table2 is deficient of information

Example:

```
SQL>select e.ename, d.deptno,
d.dnamefrom emp e, dept d
where
e.deptno(+)=d.deptno
order by e.deptno;
```

4. **Self Join :** Joining a table to itself

Display the output in the following format
“Worker works for Manager”

```
SQL> select worker.ename || ‘works for’ || manager.ename
From emp worker, emp manager
Where worker.mgr=manager.empno;
```

VIVA QUESTIONS

1. What is join?
2. Difference between Cartesian product and Join
3. What is self join?
4. Difference between left join and right join.
5. What is natural join?

6 - IMPLEMENTATION OF FUNCTIONS

AIM: To write queries using single row functions and group functions.

1. **Display the names and dob of all programmers who were born in january.**

SQL>select pname , dob from programmer where to_char (dob,'MON')='JAN';

2. Calculate the experience in years of each programmer and display along withprogrammer name in descending order.

SQL> select pname, round (months_between(sysdate, doj)/12, 2) "EXPERIENCE" fromprogrammer order by months_between (sysdate, doj) desc;

3. List out the programmer names who will celebrate their birthdays during currentmonth.

SQL>select pname from programmer where to_char(dob,'MON') like to_char (sysdate, 'MON');

4. **Display the least experienced programmer's details.**

SQL>select * from programmer where doj = (select max (doj) from programmer);

5. **Who is the most experienced programmer knowing pascal.**

SQL>select pname from programmer where doj = (select min (doj) from programmer);

6. **Who is the youngest programmer born in 1965.**

SQL> select pname , dob from programmer where dob = (select max (dob) fromprogrammer where to_char (dob,'yy') = 65);

7. **In which year, most of the programmers are born.**

SQL>select to_char (dob , 'YY') from programmer group by to_char (dob, 'YY') having count(*) = (select max (count(*)) from programmer group by to_char(dob,'YY');

8. **In which month most number of programmers are joined.**

INPUT SQL>select to_char (doj,'YY') from programmer group by to_char (doj,'YY') having count (*) = (select max (count(*)) from programmer group by to_char (doj,'YY');

9. **What is the length of the shortest name in programmer table ?**

INPUT SQL>select length (pname) from programmer where length (pname) = select min (length (pname) from programmer);

10. Display the names of the programmers whose name contains up to 5 characters.

INPUT SQL>select pname from programmer where length (pname) <=5;

11. Display all packages names in small letters and corresponding programmernames in uppercase letters.

INPUT SQL>select lower (title), upper (pname) from software;

VIVA QUESTIONS

1. Which function is used to display date and time?
2. String functions
3. Date functions
4. Difference between single row functions and group functions.
5. Which function is used to display length of a string?

7 - IMPLEMENTATION OF NESTED QUERIES

AIM :To write queries using Set operations and to write nested queries.

Set Operations:

UNION	-	OR
INTERSECT	-	AND
EXCEPT -	-	NOT

NESTED QUERY:- A nested query makes use of another sub-query to compute or retrievethe information.

1. Find the name of the institute in which the person studied and developed thecostliest package.

SQL>select splace, pname from study where pname = (select pname from software wherescost = (select max (scost) from software);

RESULT

```
SPLACE                                  PNAME
------------                            -------------
SAHBHARI                                MARY
```

2. Find the salary and institute of a person who developed the highest sellingpackage.

SQL> select study.pname, sal, splace from study, programmer where study.pname = programmer.pname and study.pname = (select pname from software where scost = (select max (scost) from software));

RESULT

```
PNAME                           SAL    SPLACE
```

MARY 4500 SABHARI

3. How many packages were developed by the person who developed the cheapestpackage.

SQL>select pname, count (title) from software where dcost = (select min(dcost) from software) group by pname;

RESULT

```
PNAME                                        COUNT(TITLE)

------------- VIJAY                         1
```

Calculate the amount to be recovered for those packages whose development costhas not yet recovered.

SQL>select title , (dcost-scost) from software where dcost > scost;

4. Display the title, scost, dcost, difference of scost and dcost in the descending orderof difference.

SQL> select title, scost, dcost, (scost - dcost) from software descending order by (scost-dcost);

5. Display the details of those who draw the same salary.

SQL> select p.pname, p.sal from programmer p, programmer t where p.pname <> t.pnameand p.sal = t.sal;(or)

SQL>select pname,sal from programmer t where pname<>t.pname and sal= t.sal;

VIVA QUESTIONS

1. What is inner query or sub query?
2. What is the sequence of execution of queries in a query with sub query?
3. What are set operations?
4. Correlated queries
5. What is outer query?

8 - IMPLEMENTATION OF GROUP FUNCTIONS & GROUP BY CLAUSE

AIM: To write queries using group functions and group by clause.

Group functions are built-in SQL functions that operate on groups of rows and return onevalue for the entire group.

Types of group functions are:

COUNT – COUNT()
MAXIMUM- MAX()
MINIMUM – MIN()
AVERAGE – AVG()
SUM – SUM()

Examples:
Display sum of the salary of all the employees in the emp table

```
SQL> select sum(sal) from emp;
```

Display maximum of the salary of all the employees in the emp table

```
SQL> select max(sal) from emp;
```

Display minimum of the salary of all the employees in the emp table

```
SQL> select min(sal) from emp;
```

Display average of the salary of all the employees in the emp table

```
SQL> select avg(sal) from emp;
```

Display count of employees in the emp table

```
SQL> select count(*) from emp;
```

GROUP BY Clause is used along with the group functions to retrieve data groupedaccording to one or more columns.
All columns in the select list that are not in group functions must be in group by clause.

Example:

```
SQL> selet deptno, avg(sal) from emp group by deptno;
```

Group by column does not have to be in the select list SQL> selet avg(sal) from emp group by deptno; **Group by on multiple columns:**

SQL>select deptno, job, sum(sal) from emp group by deptno, job;

Group by-Having Clause:
If you have to restrict the rows based on a result of a group function, you must have to usegroup-by clause along with Having clause.

Display the average salaries of the departments that have an average salary greaterthan 2000.
SQL>select deptno, avg(sal) from emp group by deptno having avg(sal)>2000;

Display the department numbers and average salary for those departments who maximum salary is greater than 2900
SQL> select deptno, avg(sal) from emp group by deptno having max(sal)>2900

Nesting Group Functions:

Display Maximum average salary.
SQL> select max(avg(sal) from emp group by deptno;

AIM: To write queries using clauses such as GROUP BY, ORDER BY, etc. and retrievinginformation by joining tables.

Source tables: emp, dept, programmer, software, study.

Order by : The order by clause is used to display the results in sorted order.
Group by **: The attribute or attributes given in the clauses are used to form groups. Tuples with the same value on all attributes in the group by clause are placed in one group.**
Having**: SQL applies predicates (conditions) in the having clause after groups have been formed, so aggregate function be used.**

1. Display total salary spent for each job

category. INPUT SQL>select job,sum (sal) from

emp group by job;
RESULT

```
JOB                                         SUM(SAL)
---------                                  ----------
```

```
    6 rows selected.
ANALYST        6000
CLERK          23050
MANAGER  8275
PRESIDENT  5000
SALESMAN  5600
assistant           2200
clerk               2003
```

2. Display lowest paid employee details under each manager.

INPUT SQL>select ename, sal from emp where sal in (select min(sal) from emp group bymgr);

RESULT

```
                         ENAME      SAL
                         ---------- ----------
                         chai       3
                         JAMES      950
                         MILLER     1000
                         ADAMS      1100
                         russel     2200
5 rows selected.
```

3. Display number of employees working in each department and their departmentname.

INPUT SQL> select dname, count (ename) from emp, dept where emp.deptno=dept.deptnogroup by dname;

RESULT

```
DNAME                          COUNT(ENAME)
--------------                         ------------
ACCOUNTING                             3
RESEARCH                               5
SALES                                  9
```

4. Display the sales cost of package developed by each

programmer. INPUT SQL>select pname, sum(scost) from

software group by pname;

RESULT

```
PNAME                                  SUM(SCOST)
--------------------                   ----------
john                                   12000
kamala                                 12000
raju                                   12333
```

3 rows selected.

5. Display the number of packages sold by each programmer. INPUT SQL>select pname, count(title) from software group by pname;

RESULT

```
PNAME                               COUNT(TITLE)
--------------------                ------------
john                                1
kamala                              1
raju                                1
```

```
ramana                                1
rani                                  1
5 rows selected.
```

6. Display the number of packages in each language for which the development costis less than thousand.

INPUT SQL>select devin, count(title) from software where dcost < 1000 group by devin;

RESULT

```
DEVIN                        COUNT(TITLE)
---------- ------------
cobol                            1
```

7. Display each institute name with number of students.

INPUT SQL>select splace, count(pname) from study group by splace;

RESULT

```
                     SPLACE              COUNT(PNAME)
                     --------------------  ------------
                     BDPS                2
                     BITS             1
                     BNRILLIANI       1
                     COIT                1
                     HYD                 1
  5 rows selected.
```

8. How many copies of package have the least difference between development andselling cost, were sold?

SQL>select sold from software where scost – dcost=(select min(scost – dcost) fromsoftware);

RESULT

SOLD

11

9. Which is the costliest package developed in Pascal.

SQL>select title from software where devin = 'PASCAL' and dcost = (select max(dcost)fromsoftware where devin = 'PASCAL');

no rows selected

10. Which language was used to develop most no .of packages.

SQL>select devin, count (*) from software group by devin having count(*) = (selectmax(count(*)) from software group by devin);

```
DEVIN                          COUNT(*)
```

```
----------                                    ----------

jsp                                            2
```

11. Who are the male programmers earning below the average salary of femaleprogrammers?

```
SQL>select pname from programmer where sal < (select avg(sal) from programmer wheresex = ‘F’) and sex = ‘M’;
PNAME
--------------------
vijay
```

12. Display the details of software developed by the male programmers earningmore than 3000/-.

```
SQL>select programmer.pname, title, devin from programmer, software where sal > 3000and sex = ‘M’ and programmer.pname = software.pname;
no rows selected
```

VIVA QUESTIONS

1. What is aggregate functions
2. List the group functions
3. Difference between group by and order by clause.
4. What is use of Having clause in Group by clause?
5. Which function is used to display the number of records in a table?

9 - IMPLEMENTATION OF DCL COMMANDS

AIM: To practice basic SQL DCL Commands such as COMMIT, ROLLBACK etc.

1. **COMMIT**

Save changes

(transactional).Syntax:

COMMIT [WORK] [COMMENT
'*comment_text*'] COMMIT [WORK] [FORCE
'*force_text*' [,*int*]]

FORCE - will manually commit an in-doubt *distributed* transaction
force_text - transaction
identifierint - sets a specific
SCN.
If a network or machine failure prevents a distributed transaction from committing properly, Oracle will store any commit comment in the data dictionary along with thetransaction ID.

SQL>commit;

RESULT: Commit complete.

2. **ROLLBACK**

Undo work done

(transactional).Syntax:

ROLLBACK [WORK] [TO

[SAVEPOINT]'*savepoint_text_identifier*']; ROLLBACK [WORK]

[FORCE '*force_text*'];

FORCE - will manually rollback an in-doubt *distributed* transaction

INPUT:

SQL>rollback;

RESULT: Rollback complete.

3. **SAVEPOINT**

Save changes to a point

(transactional).Syntax:

SAVEPOINT *text_identifier*

Example:

```
UPDATE
employees   SET
salary = 95000
WHERE last_name = 'Smith';

SAVEPOINT justsmith;

UPDATE
employees        SET
salary = 1000000;

SAVEPOINT everyone;

SELECT       SUM(salary)        FROM

employees;        ROLLBACK         TO

SAVEPOINT justsmith; COMMIT;
```

VIVA QUESTIONS

1. What is Commit?
2. What is Savepoint?
3. What is Rollback?
4. What the situations under which rollback will not work?
5. Difference between Rollback and Commit.

10: IMPLEMENTATION OF LOCKING TECHNIQUES

AIM: To learn commands related to Table Locking

LOCK TABLE Statement Manually lock one or more tables.

Syntax:
LOCK TABLE [*schema.*] *table* [*options*] IN *lockmode* MODE

[NOWAIT] LOCK TABLE [*schema.*] *view* [*options*] IN *lockmode*

MODE [NOWAIT]

Options:
PARTITION (*partition*)
SUBPARTITION
(*subpartition*)@dblink

lockmodes:
EXCLUSI
VE
SHARE
ROW EXCLUSIVE
SHARE ROW
EXCLUSIVE
ROW SHARE* | SHARE UPDATE*

If NOWAIT is omitted Oracle will wait until the table is available.

Several tables can be locked with a single command - separate with commas

e.g. LOCK TABLE table1,table2,table3 IN ROW EXCLUSIVE

MODE;Default Locking Behaviour :

A pure SELECT will not lock any rows.

INSERT, UPDATE or DELETE's - will place a ROW EXCLUSIVE lock.

SELECT...FROM...FOR UPDATE NOWAIT - will place a ROW EXCLUSIVE lock.

Multiple Locks on the same rows with LOCK TABLE

Even when a row is locked you can always perform a SELECT (because SELECT does not lock any rows) in addition to this, each type of lock will allow additional locks to be granted as follows.

ROW SHARE = Allow ROW EXCLUSIVE or ROW SHARE or SHARE locks to be granted to thelocked rows.

ROW EXCLUSIVE = Allow ROW EXCLUSIVE or ROW SHARE locks to be granted to the

locked rows.

SHARE ROW EXCLUSIVE = Allow ROW SHARE locks to be granted to the locked

rows. SHARE = Allow ROW SHARE or SHARE locks to be granted to the locked

rows.

EXCLUSIVE = Allow SELECT queries only

Although it is valid to place more than one lock on a row, UPDATES and DELETE's may stillcause a *wait* if a conflicting row lock is held by another transaction.

VIVA QUESTIONS

1. What is Locking?
2. What is shared lock?
3. What is exclusive lock
4. Difference between shared lock and exclusive lock.
5. What is binary lock?

11 - IMPLEMENTATION OF SIMPLE PL/SQL PROGRAMS

AIM: To implement simple PL/SQL Programs

1. PL/SQL block for inserting rows into EMPDET table with the followingCalculations:
HRA=50% OF BASIC
DA=20% OF BASIC
PF=7% OF BASIC
NETPAY=BASIC+DA+HRA-PF

```
DECLARE
ENO1            empdet.eno%type;
ENAME1       empdet.name%type;
DEPTNO1    empdet.deptno%type;
BASIC1          empdet.basic%type;
HRA1 empdet.HRA%type;
DA1    empdet.DA%type;
PF1 empdet.pf%type;
NETPAY1 empdet.netpay%type;
BEGIN
ENO1:=&ENO1;
ENAME1:='&ENAME1';
DEPTNO1:=&DEPTNO1;
BASIC1:=&BASIC1;
HRA1:=(BASIC1*50)/100;
DA1:=(BASIC1*20)/100;
PF1:=(BASIC1*7)/100;
NETPAY1:=BASIC1+HRA1+DA1-PF1;
```

```
        INSERT INTO EMPDET VALUES (ENO1, ENAME1, DEPTNO1, BASIC1, HRA1,
DA1,PF1, NETPAY1);
END;
```

RESULT:

```
SQL> @BASIC
Enter value for eno1:
104      old      11:
ENO1:=&ENO1;
new  11: ENO1:=104;
Enter   value   for   ename1:   SRINIVAS
REDDYold  12: ENAME1:='&ENAME1';
new  12: ENAME1:='SRINIVAS REDDY';
Enter value for deptno1: 10
old  13: DEPTNO1:=&DEPTNO1;
new  13: DEPTNO1:=10;

Enter value for basic1:
6000      old      14:
BASIC1:=&BASIC1;
new  14: BASIC1:=6000;

PL/SQL procedure successfully completed.


SQL>/
Enter value for eno1:
105      old      11:
ENO1:=&ENO1;
new  11: ENO1:=105;
Enter   value   for   ename1:
CIRAJ        old        12:
ENAME1:='&ENAME1';
new  12: ENAME1:='CIRAJ';
Enter value for deptno1: 10
old  13: DEPTNO1:=&DEPTNO1;
new  13: DEPTNO1:=10;
Enter value for basic1:
6000      old      14:
BASIC1:=&BASIC1;
new  14: BASIC1:=6000;

PL/SQL      procedure      successfully
```

```
completed.    SQL>     SELECT     *     FROM

EMPDET;
```

RESULT

```
ENO NAME                          DEPTNO     BASIC      HRA       DA       PF    NETPAY
--------- ---------------------------- --------- --------- --------- --------- --------- --------------------
---101 SANTOSH                         10        5000        2500       1000      350
                                       8150
     102 SHANKAR                    20        5000        2500        1000       350     8150
     103 SURESH                     20        5500        2750        1100       385     8965
     104 SRINIVASA REDDY            10        6000        3000        1200       420     9780
     105 CIRAJ                      10        6000        3000        1200       420     9780
```

2. PL/SQL block to check whether given number is Armstrong or not.

```
DECLARE
num      number(5);
rem    number(5); s
number(5):=0;
num1 number(5);
```

```
BEGIN
num:=&num;
num1:=num;
while(num>0)
loop

rem:=mod(num,10);


s:=s+power(rem,3);
num:=trunc(num/10);
End loop;
if (s=num1)then
dbms_RESULT.put_line(num1||' IS ARMSTRONG NUMBER ');



END;
/
else End if;

dbms_RESULT.put_line(num1||' IS  NOT ARMSTRONG NUMBER ');
```

RESULT:

```
SQL>@arm
Enter value for
num: 153 old 7:
num:=&num;
new 7: num:=153;
153 IS ARMSTRONG NUMBER

PL/SQL procedure successfully

completed.SQL> /
Enter value for
num: 123 old 7:
num:=&num;
new 7: num:=123;
123 IS  NOT ARMSTRONG NUMBER

PL/SQL procedure successfully completed.
```

3. PL/SQL block to find Sum of Digits of a given

Number.INPUT

```
DECLARE
 num      number(5);
rem number(5); sm
number(5):=0;
num1 number(5);
```

```
BEGIN
num:=&num;  num1:=num;
while(num>0)          loop
rem:=mod(num,10);
sm:=sm+rem;
num:=trunc(num/10);
end loop;
dbms_RESULT.put_line('SUM OF DIGITS OF  '||num1||' IS: '||sm);

end;
/
```

RESULT:

```
SQL> @sum
INPUT    truncated    to    2
characters Enter  value  for
num: 123
old  7:  num:=&num;
new  7:  num:=123;
SUM OF DIGITS OF  123 IS: 6
PL/SQL procedure successfully completed.

SQL> @sum
INPUT    truncated    to    2
characters Enter  value  for
num: 456
old  7:  num:=&num;
new  7:  num:=456;
SUM OF DIGITS OF  456 IS: 15
PL/SQL procedure successfully completed.
```

4. PL/SQL block to Generate Fibonacci Series

```
DECLARE
 num    number(5);
f1    number(5):=0;
f2    number(5):=1;
f3 number(5);
i number(5):=3;
BEGIN
END;
/
num:=&num;
dbms_RESULT.put_line('THE FIBONACCI SERIES IS:');
dbms_RESULT.put_line(f1);
dbms_RESULT.put_line(f2);
while(i<=num)          loop
```

```
 f3:=f1+f2;
dbms_RESULT.put_line(f3);f1:=f2;
f2:=f3;i:=i+1;
end loop;
```

RESULT:

```
SQL> start fib
Enter    value    for
num:    10   old    8:
num:=&num;
new  8:  num:=10;
THE FIBONACCI SERIES IS:
```

```
0
1
1
2
3
5
8
13
21
34
PL/SQL procedure successfully completed.
```

VIVA QUESTIONS

1. What is PL/SQL?
2. Conditional statements in PL/SQL.
3. Loops in PL/SQL
4. Data types in PL/SQL
5. How do you execute an PL/SQL block?

12 - IMPLEMENTATION OF PROCEDURE AND

FUNCTIONSAIM: TO WRITE A PROCEDURE TO INSERT NUMBER

PL/SQL procedure is a named block that does a specific task. PL/SQL procedure allows you
to encapsulate complex business logic and reuse it in both database layer and applicationlayer.

Syntax:

```
PROCEDURE [schema.]name[( parameter[,
parameter...] ) ][AUTHID DEFINER | CURRENT_USER]
IS
[--declarations
statements]BEGIN
[--executable
statements]
EXCEPTIOM
[--exception
handlers]
END[name];
```

```
SQL> create table emp1(id number(3),First_name

varchar2(20));Table created.

SQL> insert into emp1
values(101,'Nithya');1 row created.

SQL> insert into emp1
values(102,'Maya');1 row created.

SQL> select * from

emp1;          ID

FIRST_NAME

--------- --------------------

101 Nithya

102 Maya
```

```
SQL>                  set
serveroutput        on;
SQL>      create      or
replace
  2 procedure insert_num(p_num number)is
  3 begin
4         insert           into           emp1(id,First_name)
values(p_num,user);5 end insert_num;
6 /
Procedure created.
SQL> exec insert_num(3);
PL/SQL         procedure         successfully
completed.SQL> select * from emp1;


ID FIRST_NAME
--------- --------------------
101  Nithya
     102       M
aya
SCOTT
```

Function: A PL/SQL function is same as a procedure except that it returns a value.

AIM: To write a function to find factorial

Syntax:

CREATE [OR REPLACE] FUNCTION function_name [(parameter_name [IN | OUT | IN OUT] type [, ...])] RETURN

```
return_datatype
{IS
|
AS
}
BE
GI
N
<function_body
>
END[function_n
ame];
```

```
SQL> create or replace function fact(n number)

2   return number is

3   i number(10);

4   f number:=1;

5   begin

6   for i in 1..N loop

    7           f:=f*i;

8   end loop;

      9        r
    eturn f;

10  end;

    11 /


    Function created.

  SQL>  select  fact(2)  from

    dual;FACT(2)

      ----------

    2
```

RESULT:

Thus procedures and functions were implemented successfully

VIVA QUESTIONS

1. What is Procedure?
2. What is stored procedure?
3. What is function?
4. How do you execute a function?
5. How do you execute a procedure?

13 - IMPLEMENTATION OF VIEWS

AIM: To create views and to write queries on views

A view is actually a composition of a table in the form of a predefined SQL query. A view can contain all rows of a table or select rows from a table. A view can be created from one or many tables which depends on the written SQL query to create a view.

Syntax:

Create view view_name as

Select column1, column2….

From table_name

Where [condition];

1. **Create a view from single table containing all columns from the base table.**

```
SQL>create view view1 as (select * from programmer);
```

2. **Create a view from single table with selected columns.**

```
SQL>create a view view2 as (select pname,dob,doj,sex,sal from programmer);
```

3. **Create a view from two tables with all columns.**

```
SQL>create view xyz as select * from programmer full natural join software;
```

4. **Create a view from two tables with selected columns.**

```
SQL> create view lmn as (select programmer, pname, title, devin from programmer, software where sal < 3000 and programmer.pname = software.pname);
```

5. **Check all DML commands with above 4 views.**

```
SQL> insert into view1 values (‘ramu’,’12-sep-03’,’28-jan-85’,’f’,’dbase’,’oracle’,74000);

1 row created;

SQL>update view1 set salary =50000 where pname like ‘raju’;

1 row updated.
```

Note: update command does not works for all queries on views.

SQL>delete from view1 where pname like ‘raju’;

1 row deleted.

6. Drop views which you generated.

SQL>drop view

view1; View

dropped;

SQL>drop view

view2; View

dropped;

VIVA QUESTIONS

1. What is View?
2. What is simple view?
3. What is complex view?
4. How do you drop a view?
5. Can we update views?

14 - IMPLEMENTATION OF CURSORS

AIM: To implement cursors on the existing database

A **cursor** is a temporary work area created in system memory when an **SQL**statement is executed. A cursor is a set of rows together with a pointer that identifies a current row. It isa database object to retrieve data from a result set one row at a time. It is useful when we want to manipulate the record of a table in a single method.

Each cursor contains the followings 5 parts:

1. Declare Cursor: In this part we declare variables and return a set of values.
2. Open: This is the entering part of the cursor.
3. Fetch: Used to retrieve the data row by row from a cursor.
4. Close: This is an exit part of the cursor and used to close a cursor.
5. Deallocate: In this part we delete the cursor definition and release all the system resources associated with the cursor.

Syntax:
CURSOR cursor_name
IS
SELECT_statement;

1. **Cursor to display the list of Employees and Total Salary Department wise.**

```
DECLARE
 cursor c1 is select * from dept;
cursor c2 is select * from emp; s
emp.sal%type;

BEGIN
```

```
for i in c1         loop
s:=0;
dbms_RESULT.put_line('----------------------------------------------');
dbms_RESULT.put_line('Department is :' || i.deptno ||' Department name is:' ||
i.dname);

dbms_RESULT.put_line('---------------------------------------------- ');
for j in c2         loop
 if ( i.deptno=j.deptno) thens:=s+j.sal;
dbms_RESULT.put_line(j.empno|| '                              '|| j.ename || '    '|| j.sal );
```

```
        end   if;   end
loop;
dbms_RESULT.put_line('---------------------------------------------------------------');
dbms_RESULT.put_line('Total salary is: '|| s); dbms_RESULT.put_line(' ---------');

end loop;
END;
```

RESULT:

```
SQL> @abc
-----------------------------------------------------------------------
Department is :10         Department name is: ACCOUNTING
----------------------------------------------------------------
-------7782          CLARK   2450
7839       KING   5000
7934       MILLER   1300
----------------------------------------------------------------------
Total salary is:  8750
----------------------------------------------------------------------
----------------------------------------------------------------------
Department is :20  Department name is: RESEARCH
----------------------------------------------------------------
-------7369          SMITH   800
7566       JONES   2975
7788       SCOTT   3000
7876       ADAMS   1100
7902       FORD   3000
----------------------------------------------------------------------
Total salary is:  10875
-----------------------------------------------------------------------
-----------------------------------------------------------------------
Department is :30  Department name is: SALES
----------------------------------------------------------------
-------7499          ALLEN   1600
7521       WARD   1250
7654       MARTIN   1250
7698       BLAKE   2850
7844       TURNER   1500
7900       JAMES   950
-----------------------------------------------------------------------
Total salary is:  9400
-----------------------------------------------------------------------
-----------------------------------------------------------------------
Department is  :40  Department name is: OPERATIONS
```

```
-----------------------------------------------------------------------------
-----------------------------------------------------------------------------
Total salary is:  0
-----------------------------------------------------------------------------
PL/SQL procedure successfully completed.
```

2. PL/SQL CURSOR to write a Cursor to display the list of employees who areWorking as a Managers or Analyst.

```
DECLARE
cursor c(jb varchar2) is select ename from emp where job=jb;
em emp.job%type;
BEGIN
open c('MANAGER');
dbms_RESULT.put_line(' EMPLOYEES WORKING AS MANAGERS ARE:');
loop
fetch c into em;
exit        when        c%notfound;
dbms_RESULT.put_line(em);
end   loop;
close c;

open c('ANALYST');
dbms_RESULT.put_line(' EMPLOYEES WORKING AS ANALYST ARE:');
loop
END;
fetch c into em;
exit        when        c%notfound;
dbms_RESULT.put_line(em);
end   loop;
close c;
```

RESULT:

```
EMPLOYEES WORKING AS MANAGERS ARE:
J
O
N
E
S
B
L
A
K
E
C
L
A
R
K
EMPLOYEES WORKING AS ANALYST ARE:
S
C
O
T
T
F
O
R
D

PL/SQL procedure successfully completed.
```

3. Writing PL/SQL CURSOR to write a Cursor to display List of Employees from EmpTable in PL/SQL block

```
DECLARE
cursor c is select empno, ename, deptno, sal from emp ;i
emp.empno%type;
j  emp.ename%type;  k
emp.deptno%type;


BEGIN
```

```
l emp.sal%type;

open c;
dbms_RESULT.put_line('Empno, name, deptno, salary of employees are:= ');loop

END;
fetch c into i, j, k, l;
exit when c%notfound; dbms_RESULT.put_line(i||' '||j||'
'||k||' '||l);
end loop;
close c;
```

RESULT:

```
SQL> @EMP
Empno,name,deptno,salary of employees are:=
  7369    SMITH    20   800
  7499    ALLEN    30   1600
  7521    WARD     30   1250
  7566    JONES    20   2975
  7654    MARTIN   30   1250
  7698    BLAKE    30   2850
  7782    CLARK    10   2450
  7788    SCOTT    20   3000
  7839    KING     10   5000
  7844    TURNER   30   1500
  7876    ADAMS    20   1100
  7900    JAMES    30   950
  7902    FORD     20   3000
  7934    MILLER   10   1300

PL/SQL procedure successfully completed.
```

4. Writing PL/SQL CURSOR to write a Cursor to find employee with given job anddeptno.

```
DECLARE
        cursor c1(j varchar2, dn number) is select empno, ename from emp where
job=j anddeptno=dn;
row1    emp%rowtype;
jb emp.job%type;
d emp.deptno%type;
```

```
BEGIN
```

```
jb:='&jb';
d:=&d;
open c1(jb,d);
fetch c1 into row1.empno,row1.ename;

if c1%notfound then
dbms_RESULT.put_line('Employee does not exist');
else
dbms_RESULT.put_line('empno is:'||row1.empno||'  '||'employee name
is:'||row1.ename);
end if;
END;
```

RESULT:

```
SQL> @CUR
Enter    value    for    jb:
MANAGER      old       7:
jb:='&jb';
new  7: jb:='MANAGER';
Enter  value  for
d:    20    old    8:
d:=&d;
new  8: d:=20;
empno is:7566 employee name is:JONES

PL/SQL         procedure          successfully

completed.SQL> /
Enter  value  for  jb:
CLERK      old       7:
jb:='&jb';
new  7: jb:='CLERK';
Enter  value  for
d:    40    old    8:
d:=&d;
new       8:   d:=40;
Employee does not
exist

PL/SQL procedure successfully completed.
```

VIVA QUESTIONS

1. What is Cursor?
2. What is explicit cursor?

3. What is implicit cursor?
4. How do you fetch in cursor?
5. How do you deallocate in cursor?

15 - IMPLEMENTATION OF TRIGGERS

AIM : To implement triggers on the database.
Triggers are stored programs, which are automatically executed or fired when someevents occur.

Syntax:

```
CREATE [OR REPLACE ] TRIGGER trigger_name
{BEFORE | AFTER | INSTEAD OF }
{INSERT [OR] | UPDATE [OR] | DELETE}
[OF col_name]
ON table_name
[REFERENCING OLD AS o NEW
AS n][FOR EACH ROW]
WHEN
(condition)
DECLARE
Declaration-statements
BEGIN
Executable-statements
EXCEPTION
Exception-handling-statements
END
```

TRIGGER to ensure that DEPT TABLE does not contain duplicate of null values in DEPTNO column.

```
CREATE OR RELPLACE TRIGGER trig1 before insert on dept for each
rowDECLARE
a number;
BEGIN
if(:new.deptno is Null) then
raise_application_error(-20001,'error::deptno cannot be null');
else

deptno');

select count(*) into a from dept where deptno=:new.deptno;if(a=1) then
raise_application_error(-20002,'error:: cannot have duplicate

END;

end if;
end if;
```

RESULT:

```
SQL> @trigger

Trigger created.

SQL> select * from dept;
 DEPTNO DNAME          LOC
 --------- ------------- -------------
 10 ACCOUNTING         NEW YORK
 20 RESEARCH           DALLAS
 30 SALES              CHICAGO
 40 OPERATIONS         BOSTON

 SQL> insert into dept
 values(&deptnp,'&dname','&loc'); Enter value for
 deptnp: null
 Enter value for dname:
 marketing Enter value for
 loc: hyd
 old 1: insert into dept
 values(&deptnp,'&dname','&loc') new 1: insert
 into dept values(null,'marketing','hyd') insert into
 dept values(null,'marketing','hyd')
 *
 ERROR at line 1:
 ORA-20001: error::deptno cannot
 be null ORA-06512: at
 "SCOTT.TRIG1", line 5
 ORA-04088: error during execution of trigger 'SCOTT.TRIG1'

 SQL> /
 Enter value for deptnp: 10
 Enter value for dname:
 manager Enter value for
 loc: hyd
 old 1: insert into dept
 values(&deptnp,'&dname','&loc') new 1: insert
 into dept values(10,'manager','hyd') insert into
 dept values(10,'manager','hyd')
 *
 ERROR at line 1:
 ORA-20002: error:: cannot have duplicate
 deptno ORA-06512: at "SCOTT.TRIG1", line
 9
 ORA-04088: error during execution of trigger 'SCOTT.TRIG1'
```

```
SQL> /
Enter value for deptnp: 50
Enter value for dname:
MARKETING Enter value for
loc: HYDERABAD
old  1: insert into dept values(&deptnp,'&dname','&loc')
new  1: insert into dept values(50,'MARKETING','HYDERABAD')

1 row created.
SQL> select * from dept;

DEPTNO DNAME          LOC
```

```
--------- -------------- -------------
 10                 ACCOUNTING
                            NE
 W YORK
 20 RESEARCH            DALLAS
 30 SALES            CHICAGO
 40 OPERATIONS           BOSTON
 50 MARKETING            HYDERABAD
```

VIVA QUESTIONS

1. What is Trigger?
2. How many types of triggers are there?
3. What is row level trigger?
4. How do table level trigger?
5. How is a trigger executed?

16 - IMPLEMENTATION OF DCL COMMANDS

AIM: To create a user and implement GRANT and REVOKE Commands

- Discretionary access control is based on the concept of access rights, or privileges, andmechanisms for giving users such privileges.
- A privilege allows a user to access some data object in a certain manner (e.g., to read orto modify).
- Privileges are of two types:
  - System Privileges
  - Object privileges
- **System Privileges** are normally granted by a DBA to users. Examples of systemprivileges are CREATE SESSION, CREATE TABLE, CREATE USER etc.
- **Object privileges** means privileges to perform a particular action on objects such astables, views, synonyms, procedure. These are granted by owner of the object.
- A user who creates a database object such as a table or a view automatically gets allapplicable privileges on that object.
- The DBMS subsequently keeps track of how these privileges are granted to other users, and possibly revoked, and ensures that at all times only users with the necessary privileges can access an object.
- SQL-92 supports discretionary access control through the GRANT and REVOKEcommands.
- The GRANT command gives privileges to users,

  Syntax: - GRANT **privileges** ON **object** TO **users** [WITH GRANT OPTION]

- The REVOKE command takes away privileges.

  Syntax: - REVOKE [GRANT OPTION FOR] **privileges**
  ON **object** FROM **users {***RESTRICT* / CASCADE}

<u>Examples</u>

To Create a User:
Sytax: Create user <username> identified by <password>;

Create user aurora identified by aurora123;

Now the user has to get the privilege **Create Session** to login into his session.
Grant create session to <username>;

Getting privilege to create a table:
Grant Create table to username;

Getting privilege to create a view:
Grant create view to username;

We can also grant privileges in one statement.

Grant create table, create view to username;

To revoke the privileges from the user:
Revoke create table, create view from username;

Suppose you own emp table. Now you want to grant select, update, insert privilege on thistable to another user "aurora".
Grant select, update, insert on emp to aurora;

Suppose you want to grant all privileges on emp table to aurora. Then

Grant all on emp to aurora;

For example, to revoke select, update, insert privilege you have granted to Sami thengivethe following statement.
Revoke select, update, insert on emp from aurora;

To revoke select statement on emp gran ted to public give the following command.

Revoke select on emp from

public;<u>ROLES</u>

A role is a group of Privileges. A role is very handy in managing privileges, particularly in such situation when number of users should have the same set of privileges.

For example, you have four users: Sami, Scott, Ashi, Tanya in the database. To these usersyou want to grant select, update privilege on EMP table, select, delete privilege on dept table. To do this first create a role by giving the following statement

Create role clerks

Then grant privileges to this role.

Grant select, update on emp to clerks;
Grant select, delete on dept to clerks;

Now grant this clerks role to users like this

Grant clerks to sami, scott, ashi,

tanya ;

VIVA QUESTIONS

1. What is Grant?
2. What is Revoke?
3. What are system privileges?
4. What are object privileges
5. How do you create a user?

17 - IMPLEMENTATION OF FORMS

AIM: To design a form using Oracle Developer

2000 Introduction

Use Form Builder to simplify for the creation of data-entry screens, also known as Forms. Forms are the applications that connect to a database, retrieve information requested bythe user, present it in a layout specified by Form designer, and allow the user to modify or add information. Form Builder allows you to build forms quickly and easily.

In this Hands-On, you learn how to: Create a Data block for the “Customer” table, create a layout, Use “content” canvas, Use “execute query”, Navigate a table, use next, previous record, Enter query, Manipulate table’s record, Insert, Update, Delete and Save record.

Form Builder Tool

Open the "Form Builder" tool.

Welcome window

You will get the ‘Welcome to the Form Builder’ window. If you don’t want to get this window anymore uncheck the ‘Display at startup’ box. You can start your work with any of the following options:

- Use the data Block Wizard
- Build a new form manually
- Open an existing form
- Build a form based on a template

The default is ‘Use the data Block Wizard.’ If you want to build a new form manually, click on "Cancel” or check ‘Build a new form manually’ and click ‘OK.’

Connect to database

In the ‘Object Navigator’ window, highlight "Database Objects." Go to the Main menu and
choose "File," then "Connect."

In the ‘Connect’ window, login in as “Scott” password “tiger,” then click “CONNECT.”

Notice that the box next to ‘Database Objects’ is not empty anymore and it has a ‘+’ sign in
it. That will indicate that this item is expandable and you are able to see its entire

objects. Click on the '+' sign next to the 'Database Objects' to expand all database

schemas.

Create a Module

In the 'Object Navigator' window, highlight module1. This is a default name. Go to the Main menu and choose "File," select "Save as" to store the new object in the "itself" folder andsave it as customer data entry. "c: _de." In this example the 'DE' abbreviation stands for DataEntry.

Create a Data Block

In the 'Object Navigator' window, highlight "Data Blocks," and click on the "create" icon.The 'Create' icon is in the vertical tool bar in the 'Object Navigator' window. It is a green '+' sign. If you drag your cursor on the icon a tooltip will show 'Create.'

New Data Block

In the 'New Data Block' window, choose the default option "Data Block Wizard" and click
"OK."

Welcome Data Block

In the 'Welcome Data Block Wizard' window click on the "NEXT" icon.

Type of Data Block

Select the type of data block you would like to create by clicking on a radio button. Select the default option 'Table or View' and then click "NEXT" again.

Selecting Tables

Click on "browse." In the 'Tables' window, highlight the "cust11" table; then click "OK."

Selecting columns for the Data Block Wizard

To choose all columns, click on the two arrow signs in the 'Data Block Wizard' window. To choose selected columns, click on the one arrow sign. And then select all columns, and click "next."

Layout Wizard

End of the Data Block Wizard and beginning of the Layout Wizard

In the 'Congratulations' screen, use the default checkmark radio button (Create the data block, then call the Layout Wizard), and click "Finish." You can also use the Data Block Wizard to modify your existing data block. Simply select the data block in the Object Navigator and click the Data Block Wizard toolbar button, or choose 'Data Block wizard' from the 'Tools' menu.

Welcome screen

In the 'Welcome to the Layout Wizard' window, click" Next."

Selecting canvas

In the 'Layout Wizard' window, select the "new canvas" option. Canvas is a place that you will have your objects such as columns, titles, pictures, etc. If you have already had your canvas, select the canvas and then click on the next. The following are different types of canvases: Content, Stacked, Vertical Toolbar, Horizontal Toolbar, and Tab.

Think of the 'Content' canvas as one flat place to have all your objects. In the stacked canvas, you can have multiple layers of objects and it is the same as the tab canvas. You use the vertical or horizontal toolbar canvases for your push buttons. Check the different types of canvases by clicking on the 'down arrow' box next to the 'Type' field. Select "content," then click "Next."

Selecting Columns for the Layout Wizard

In the 'Layout Wizard' window, select all the columns. These are the columns that you want
to be displayed on the canvas. Then click "Next."

Change your objects appearances

Change size or prompt if needed. In this window, you can enter a prompt, width, and height for each item on the canvas. You can change the measurement units. As a default the defaultunits for item width and height are points. You can change it to inch or centimeter. When you change size, click "Next."

Selecting a layout style

Select a layout style for your frame by clicking a radio button. Select "Form," if you want one record at a time to be displayed. Select "Tabular," if you want more than one record at atime to be displayed. Select "Forms," and then click "next."

Record layout

Type the "Frame Title" and click "next." Checkmark the 'Display Scrollbar' box when you use multiple records or the 'Tabular' option.

Congratulation Screen

In the ‘Congratulations’ window, click "Finish." You will see the output layout screen.

Make some window adjustments and then run the form. To run the form, click on the 'Run' icon. The 'Run' icon is on the horizontal toolbar in the 'CUSTOMER_DE' canvas.

The object module should be compiled successfully before executing the Form.

Execute Query

Click on the "Execute Query" icon below the main menu. If you drag the cursor on the toolbar in the 'Forms Runtime' window, a tooltip will be displayed and you see 'Execute Query.'

So to know all your option, drag your cursor to view all the icon descriptions.

Next Record

Click on the "Next Record" icon to navigate to the next record.

Previous Record

Click on the "Previous Record" icon to navigate to the previous

record. This is an easy way to navigate through the "Customer"

table.

Enter Query

Click on the "Enter Query" icon to query selected records.

Insert Record

Click "Insert Record" to add new customer. All items on the forms will be blanked. You can either type all the customer information or duplicate it from previous record.

Duplicate Record

To duplicate the previous record, go to the main menu and select the 'Record' sub-menu. A drop down menu will be displayed. Select the 'Duplicate' option in the sub-menu.

Apply the changes. Remember in this stage, your record was inserted but not committed yet.

Next and Previous Record

Click "next record" and "previous record" to navigate through the records and the one was added.

Save transactions

Click "Save" to commit the insert statement.

Delete Record

Click "Remove Record" to delete the record.

Lock a Record

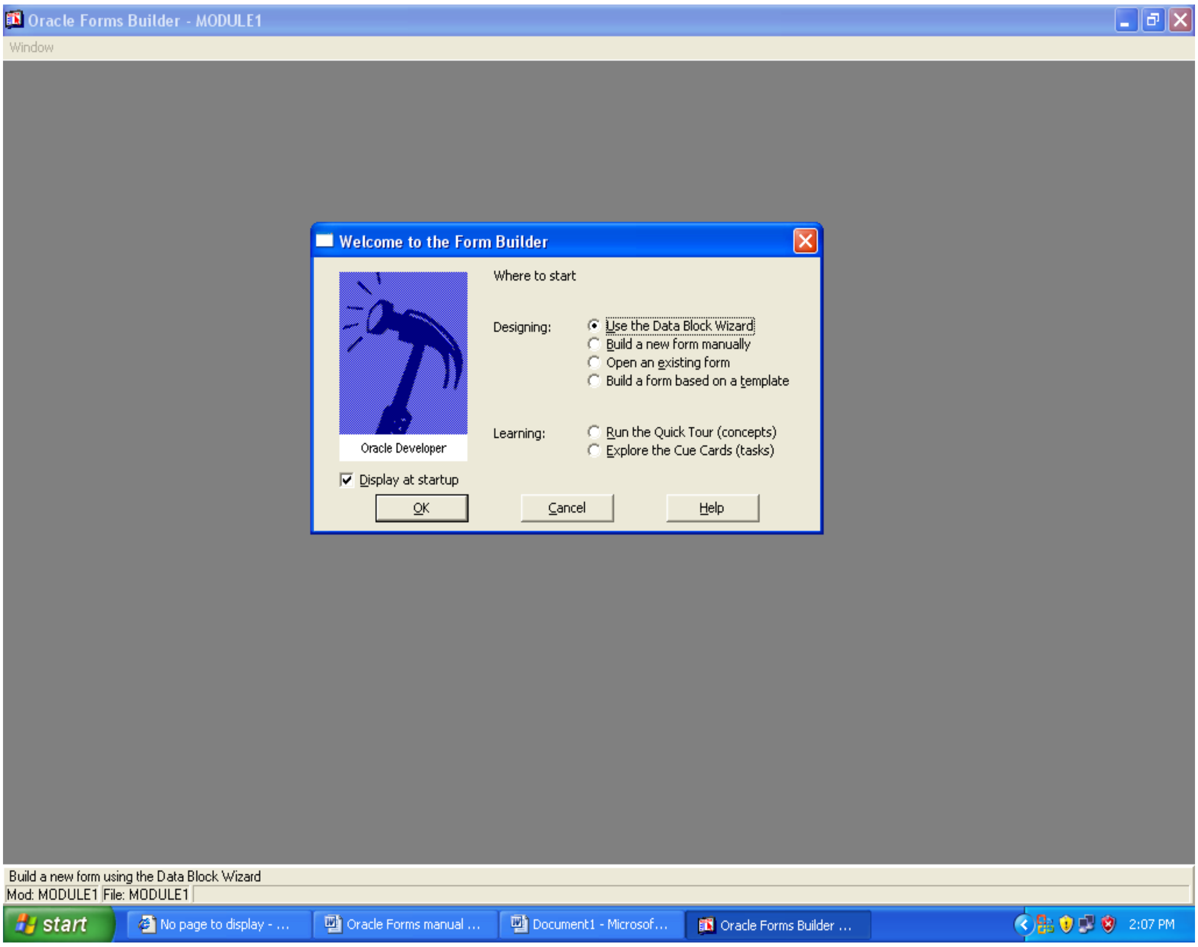
You can also lock the record.

Exit from Form Runtime

Exit the FORM Runtime. If you have not committed any transaction, you will be promptedto save changes. Click “YES” to save changes.

Click “OK” for acknowledgement.

Don’t forget to save the Form.



Welcome to the Form Builder

Where to start

Designing:
● Use the Data Block Wizard
○ Build a new form manually
○ Open an existing form
○ Build a form based on a template

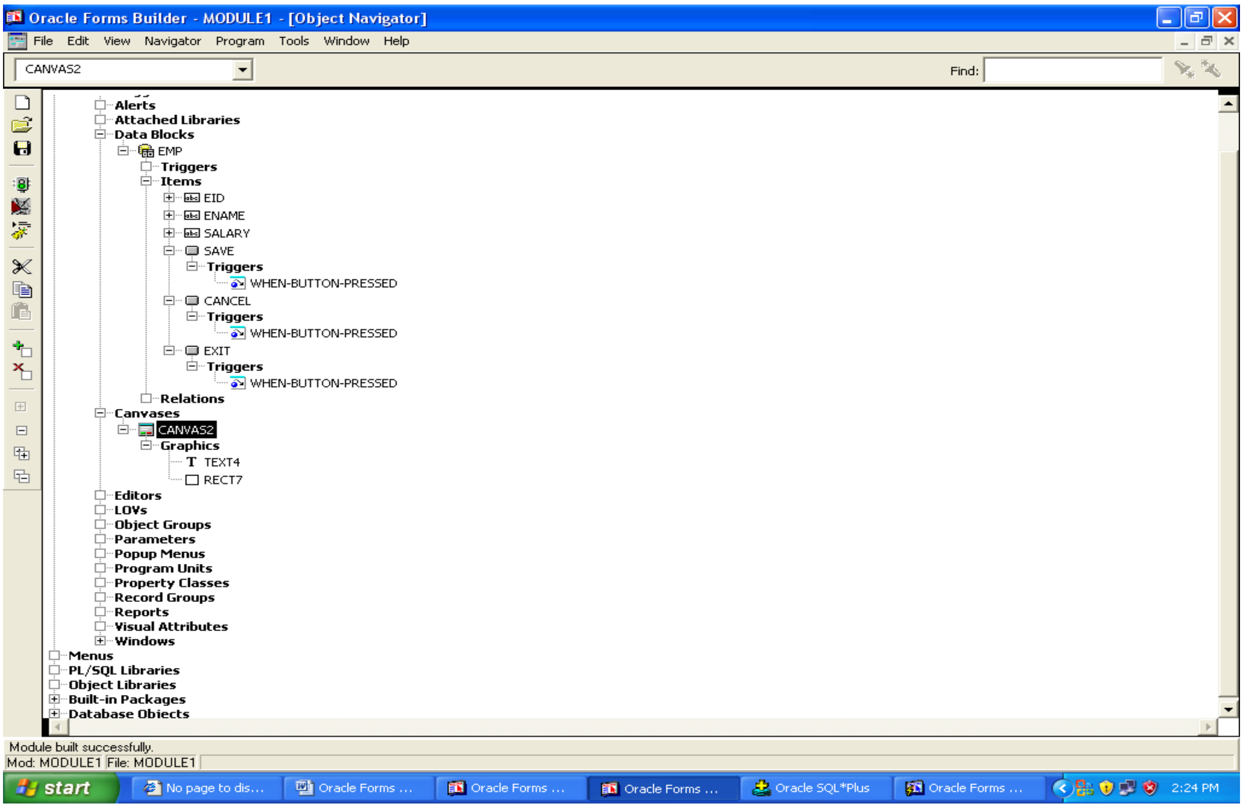
Learning:
○ Run the Quick Tour (concepts)
○ Explore the Cue Cards (tasks)

☑ Display at startup

OK | Cancel | Help

Build a new form using the Data Block Wizard
Mod: MODULE1 File: MODULE1

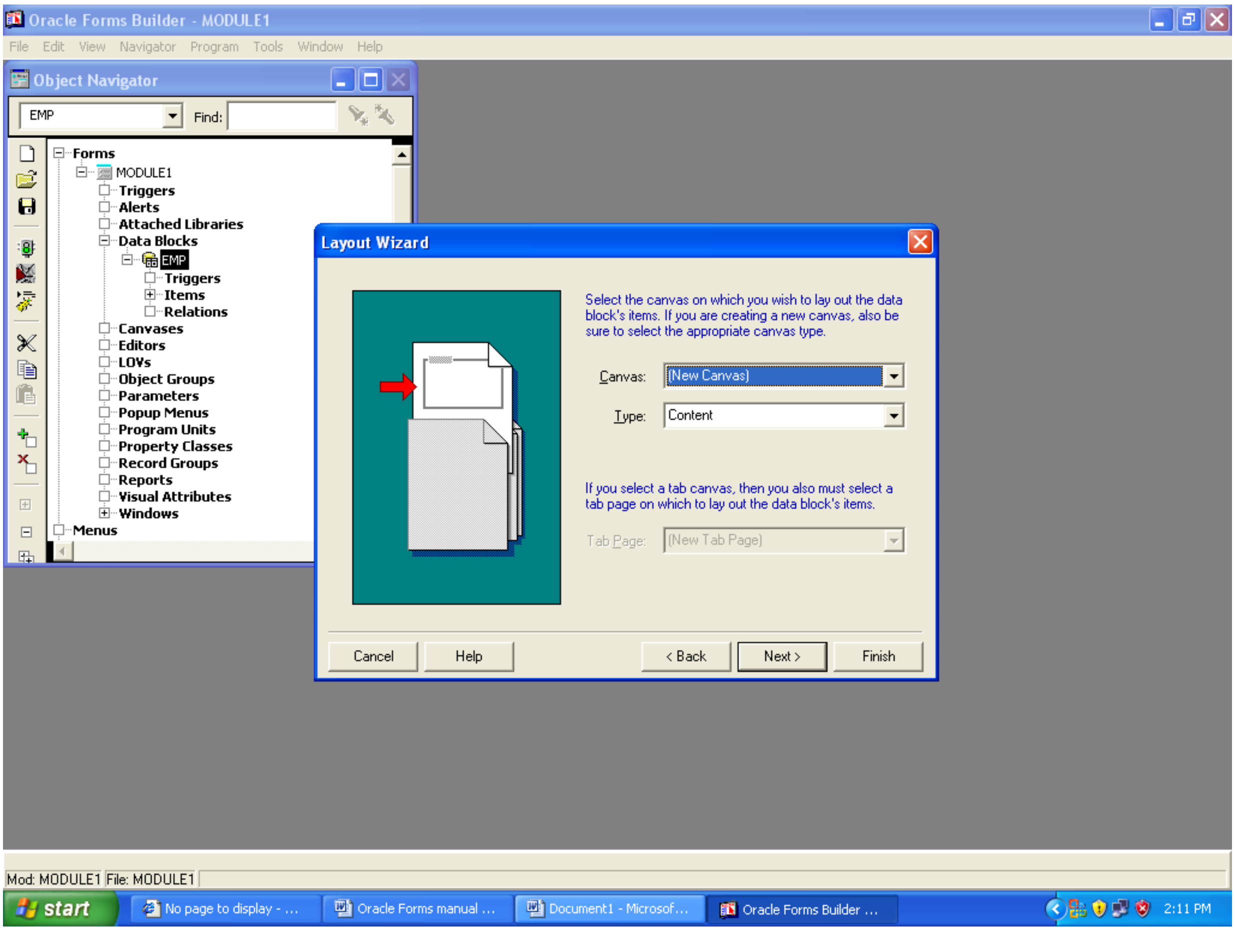
Selecting the type of form to create



```
    Alerts
    Attached Libraries
    Data Blocks
        EMP
            Triggers
            Items
                EID
                ENAME
                SALARY
                SAVE
                    Triggers
                        WHEN-BUTTON-PRESSED
                CANCEL
                    Triggers
                        WHEN-BUTTON-PRESSED
                EXIT
                    Triggers
                        WHEN-BUTTON-PRESSED
            Relations
    Canvases
        CANVAS2
            Graphics
                T TEXT4
                RECT7
    Editors
    LOVs
    Object Groups
    Parameters
    Popup Menus
    Program Units
    Property Classes
    Record Groups
    Reports
    Visual Attributes
    Windows
Menus
PL/SQL Libraries
Object Libraries
Built-in Packages
Database Objects
```

Module built successfully.
Mod: MODULE1 File: MODULE1

Object wizard



Select the canvas on which you wish to lay out the data block's items. If you are creating a new canvas, also be sure to select the appropriate canvas type.

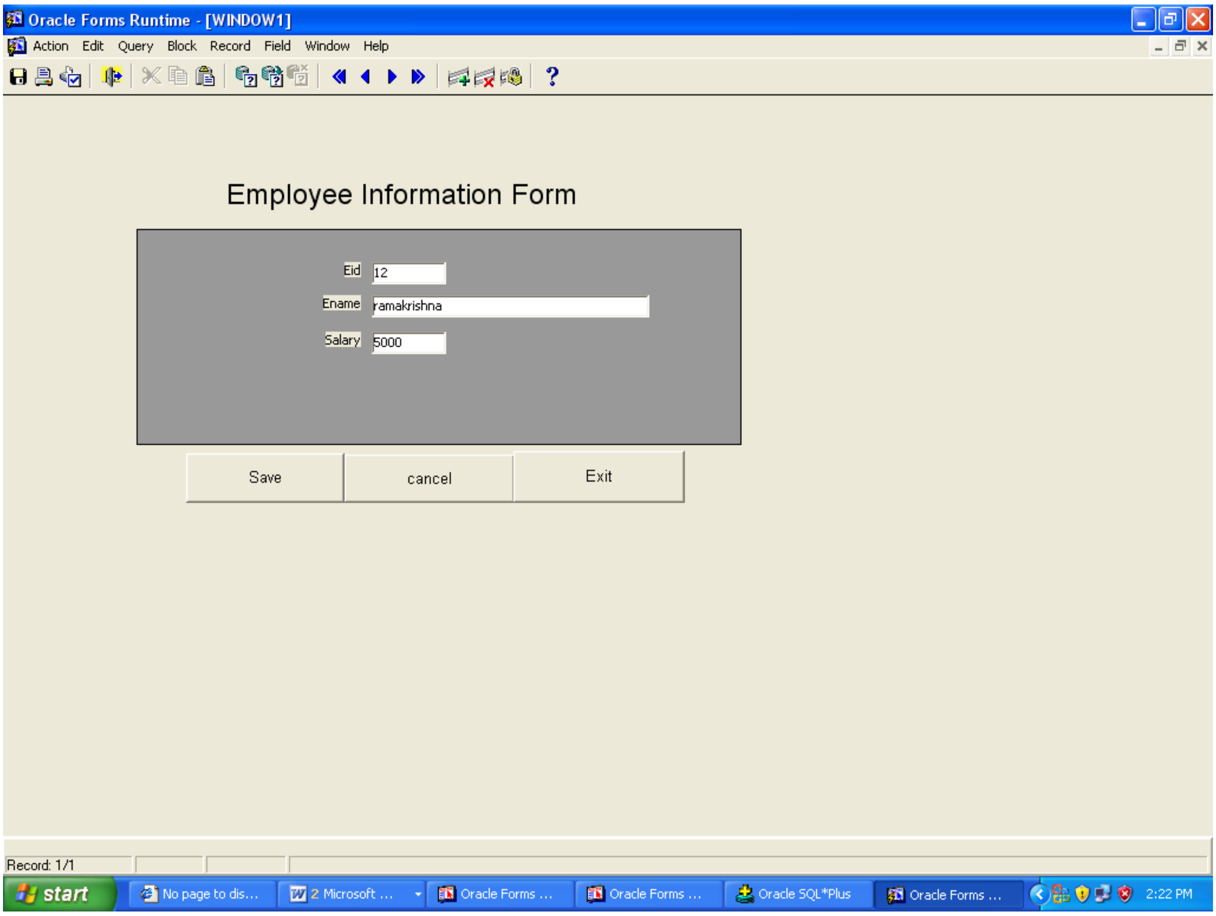
Canvas: (New Canvas)

Type: Content

If you select a tab canvas, then you also must select a tab page on which to lay out the data block's items.

Tab Page: (New Tab Page)

Selecting the canvas on which data block can be displayed



Employee Information Form

Eid: 12

Ename: ramakrishna

Salary: 5000

Record: 1/1

Form showing the Employee details

18 - IMPLEMENTATION OF REPORTS

AIM: To design reports using Oracle Developer 2000

Introduction

Tabular report shows data in a table format. It is similar in concept to the idea of an Oracle table. Oracle, by default, returns output from your select statement in tabular format.

Hands-on

In this Hands-On, your client is a stock broker that keeps track of its customer stock transactions. You have been assigned to write the reports based on their reports layout requirements.

Your client wants you to create a simple listing report to show list of the stock trades by using stocks table for their brokerage company

Your tasks are:

1- Write a tabular report.

2- Apply user layout

Format mask. 3- Run the

report.

4- Test the repot.

You will learn how to: use report wizard, object navigator, report builder, “date model”, property palette, work on query and group box, see report style, use tabular style, navigating through report’s record, change the format mask for dollar, numeric and dateitems.

Open Report Builder

Tool Open the "Report

Builder" tool. **Connect to**

database

In the Object Navigator, highlight "Database Objects,” choose "File," then select the"Connect" option.

In the ‘Connect’ window, login as “itself” password schooling, then click “CONNECT.”

Save a report

In the Object Navigator, highlight the "untitled" report, choose “File,” and select the “Saveas” option.

In the ‘Save as’ window, make sure to save the report in the ISELF folder and name it "rpt01_stock_history,” report number 1 stock history.

Data Model

In the Object Navigator, double click on the "Data Model" icon.

Create SQL box

In the Data Model window, click on the "SQL Query" icon. Then drag the plus sign cursorand click it anywhere in the “Data Model” screen where you wish your object to be.

In the ‘SQL Query Statement’ window, write a query to read all the stocks record sorted bytheir symbol.

(SQL Query

Statement)

SELECT * FROM

stocks ORDER BY

symbol Click

“OK.”

Change SQL box’s name

In the Data Model window, in the “SQL” box, right click on the ‘Q_1’ and open its property palette.

In its property palette, change the name to Q_STOCKS. Then close the window.

Change GROUP box’s name

In the Data Model, right click on the group box (G_SYMBOL) and open its property palette.In the Group property palette, change the name to 'G_STOCKS,' and close the

window.

Open Report Wizard

In the Data Model, click on the 'Report Wizard' icon on the horizontal tool bar.

In the Style tab, on the Report Wizard window, type 'Stock History' in the Title box andchoose the report style as 'Tabular.'

Notice that when you change the report style a layout of that report will be displayed onthe screen.

Choose a different style to display its layout of its report style.

Data, Fields, Totals, Labels and Template tabs

Click "NEXT" to go to the Data tab. In the 'SQL Query Statement' verify your query.

Click "NEXT" to navigate to the Fields tab, select the fields that you would like to be displayin your report. Select all the columns to be display.

Click "NEXT" to navigate to Totals tab, select the fields for which you would like to calculate totals. We have none in this hands-on exercise.

Click "NEXT" to open the Labels tab, modify the labels and widths for your fields and totalsas desired.

Click "NEXT" again to go to the Template tab, and choose a template for your report. Your report will inherit the template's colors, fonts, line widths, and structure.

Use the default template and click "finish."

Running a report

Now, you should have your output report on the screen.

Resize an object

Maximize the output report and format the report layout. To resize an object , select it anddrag its handler to the preferred size.

Move an object

To move an object, select and drag it while the cursor is on the object.This is a simple report.

Navigate through the output

To navigate through the output report in the Report Editor - Live Pre-viewer, click on the"next page" or "previous page" icon on the horizontal toolbar.

Do the same with the "first page" or "last page" icon.

Use the “zoom in” and “zoom out” icon to preview the report.

Know report’s functions

To know each icon functionalities, drag your cursor on it and a tooltip will display itsfunction.

Change Format Mask

To change the "format mask" of a column, the column should be selected. Then go to the toolbar and click on the “$” icon, "add decimal place," and the “right justify” format to the all currency columns (Todays Low, Todays High, and current price)

Select the “traded today” column, and click on the ‘,0’ icon (apply commas), and make itright justify.

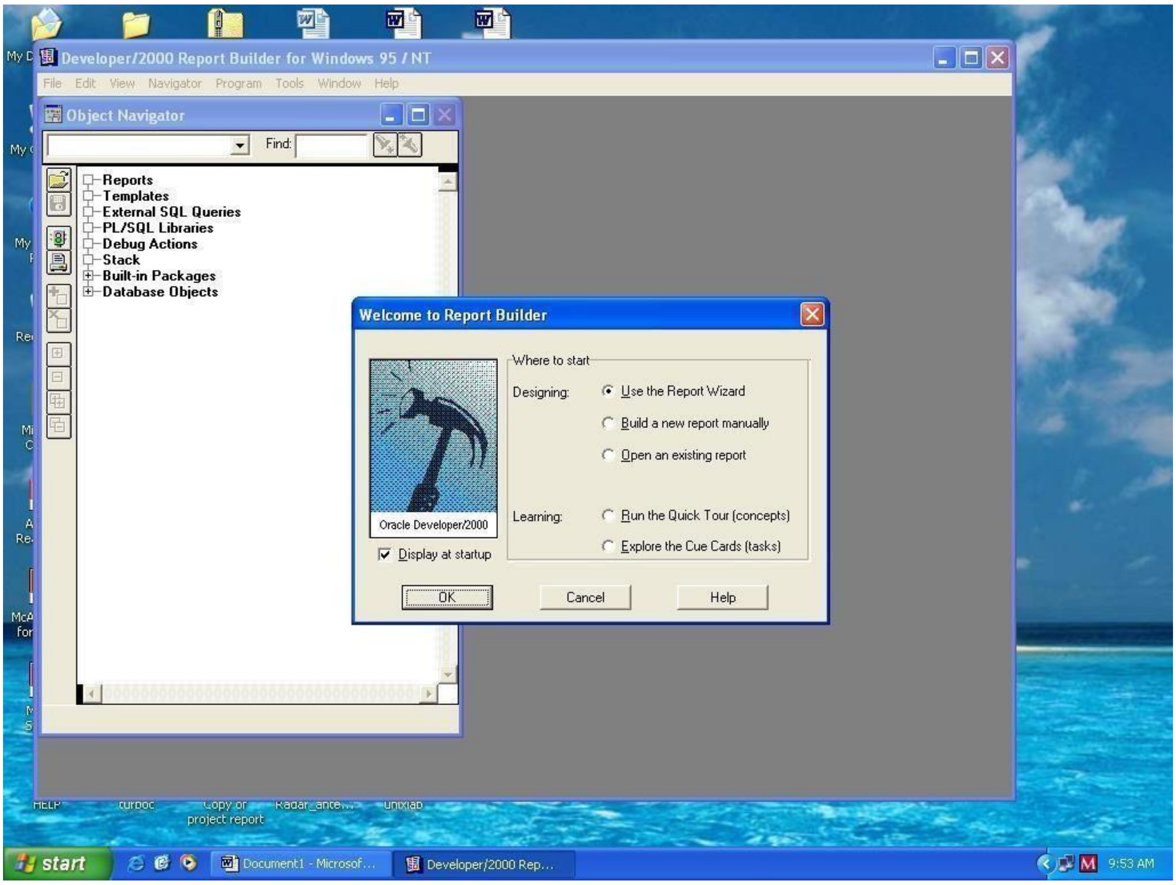
Also, you can change any attributes of field by opening its property palette. To open an

object’s property palette, right click on it and select the Property Palette

option. Right click on the "trade date" column and open its "property

palette."

Change the date "Format Mask" property and make it “year 2000 complaint (MM-DD-RR).”



Welcome to Report Builder

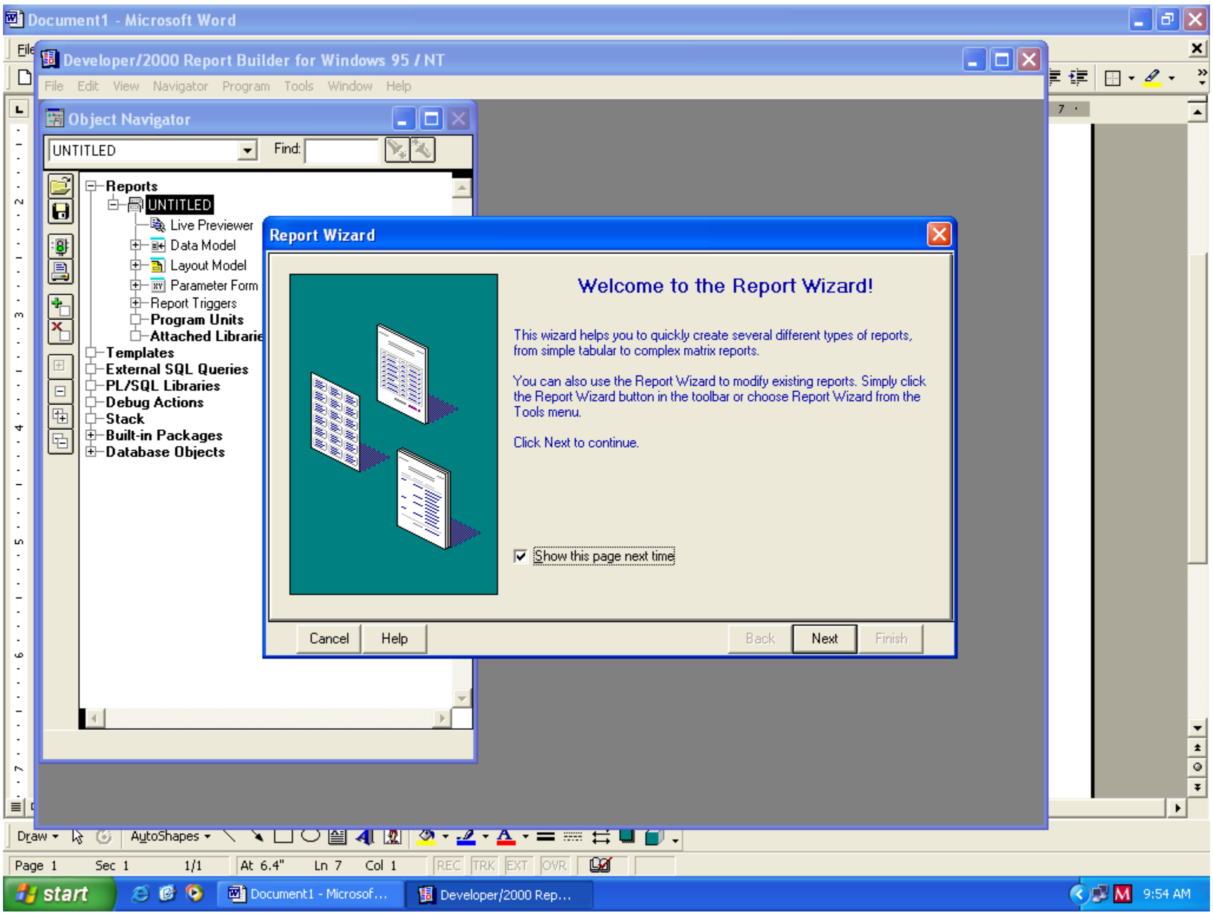
Where to start

Designing:
● Use the Report Wizard
○ Build a new report manually
○ Open an existing report

Learning:
○ Run the Quick Tour (concepts)
○ Explore the Cue Cards (tasks)

☑ Display at startup

Selecting type of report



Welcome to the Report Wizard!

This wizard helps you to quickly create several different types of reports, from simple tabular to complex matrix reports.

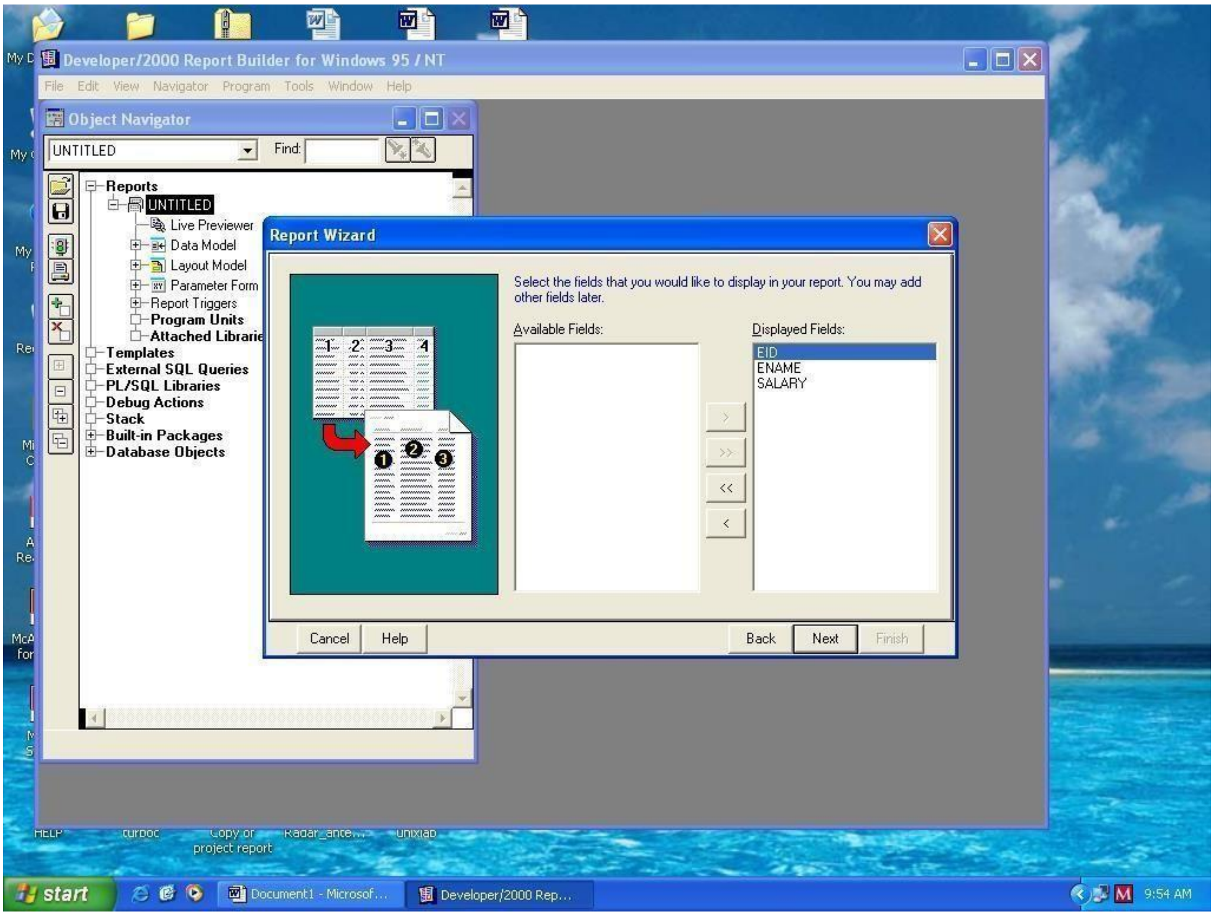
You can also use the Report Wizard to modify existing reports. Simply click the Report Wizard button in the toolbar or choose Report Wizard from the Tools menu.

Click Next to continue.

☑ Show this page next time

Creating reports

Selecting the format of reports



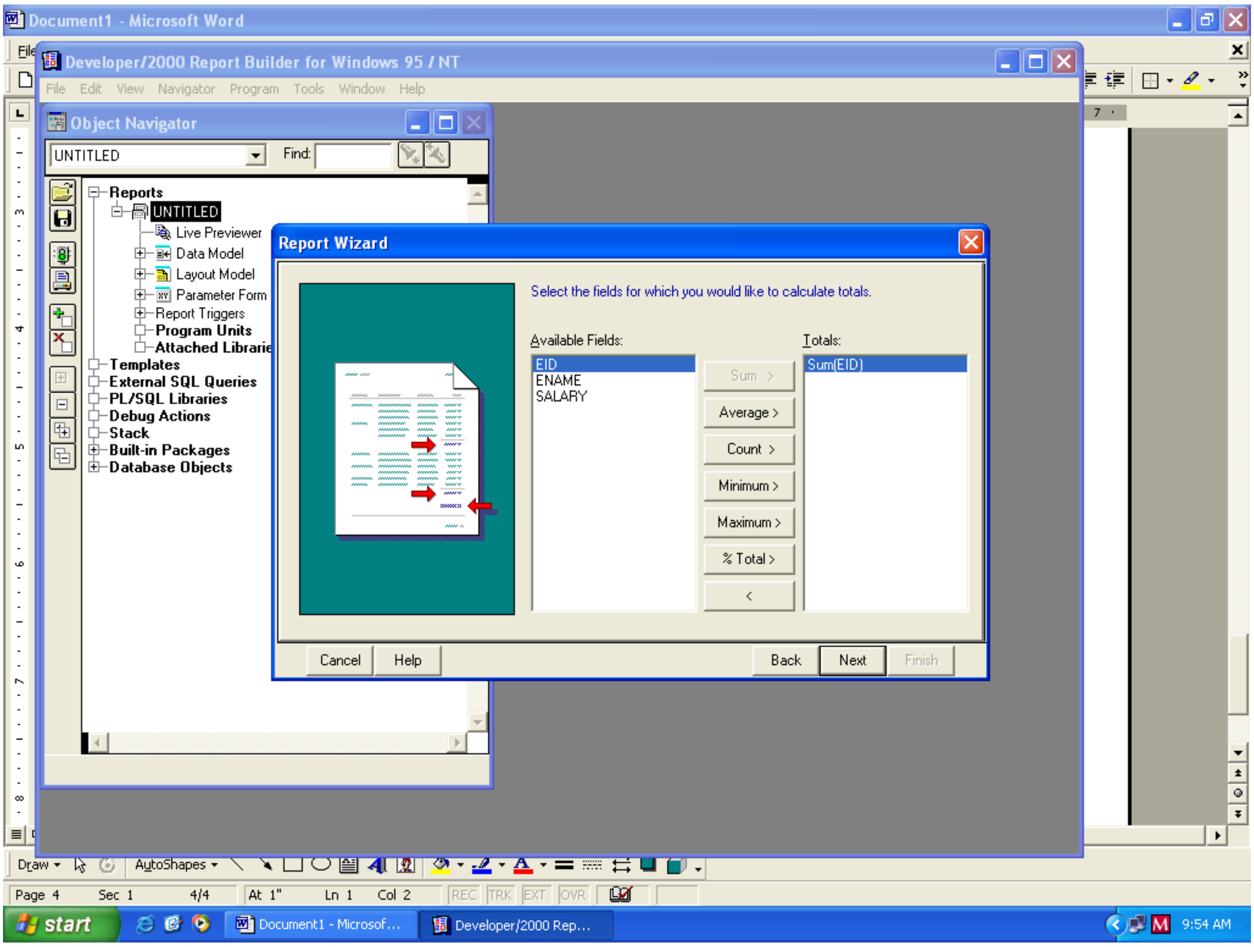
Select the fields that you would like to display in your report. You may add other fields later.

Available Fields:

Displayed Fields:

EID
ENAME
SALARY

Selecting the Table in database

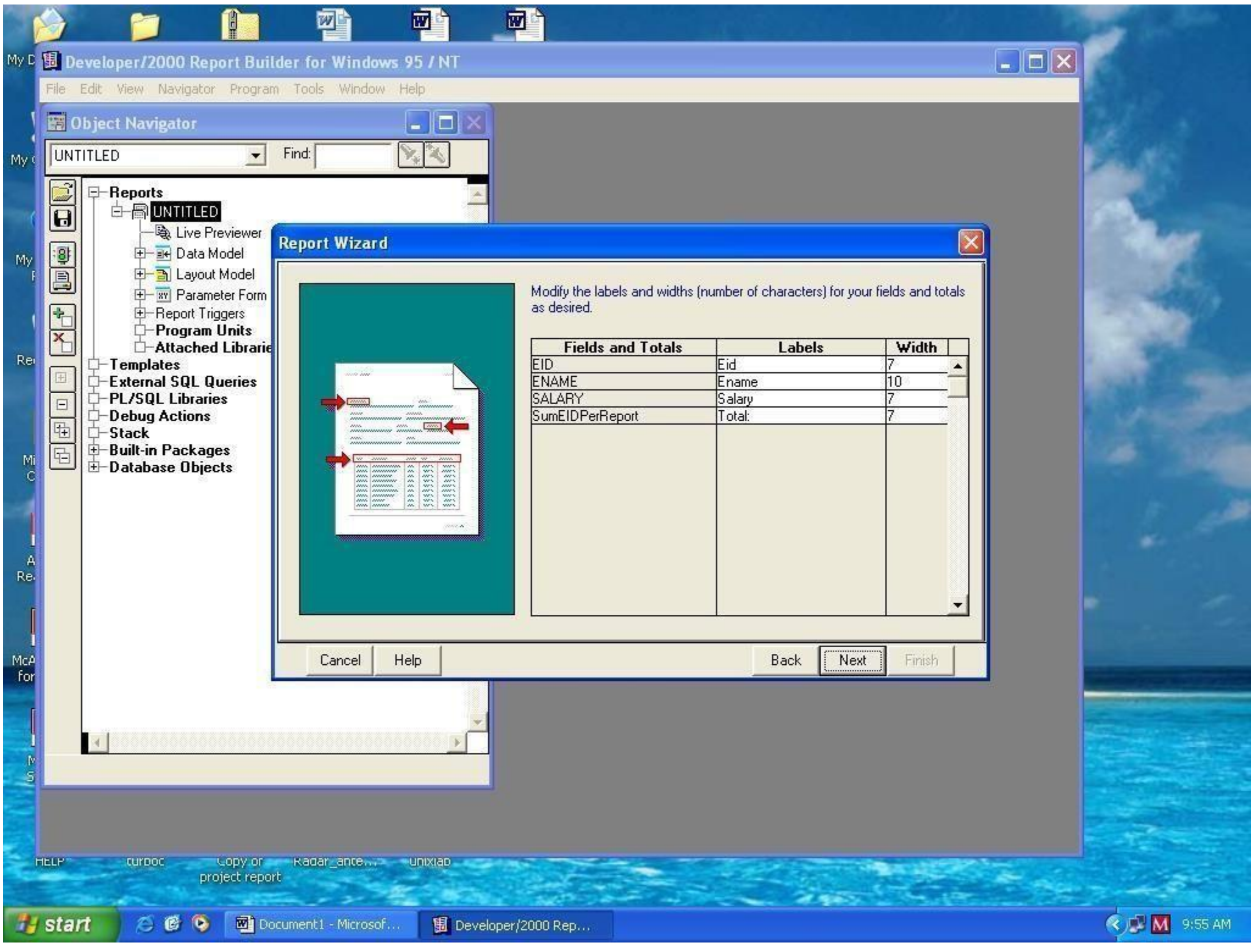


Select the fields for which you would like to calculate totals.

Available Fields:
EID
ENAME
SALARY

Totals:
Sum(EID)

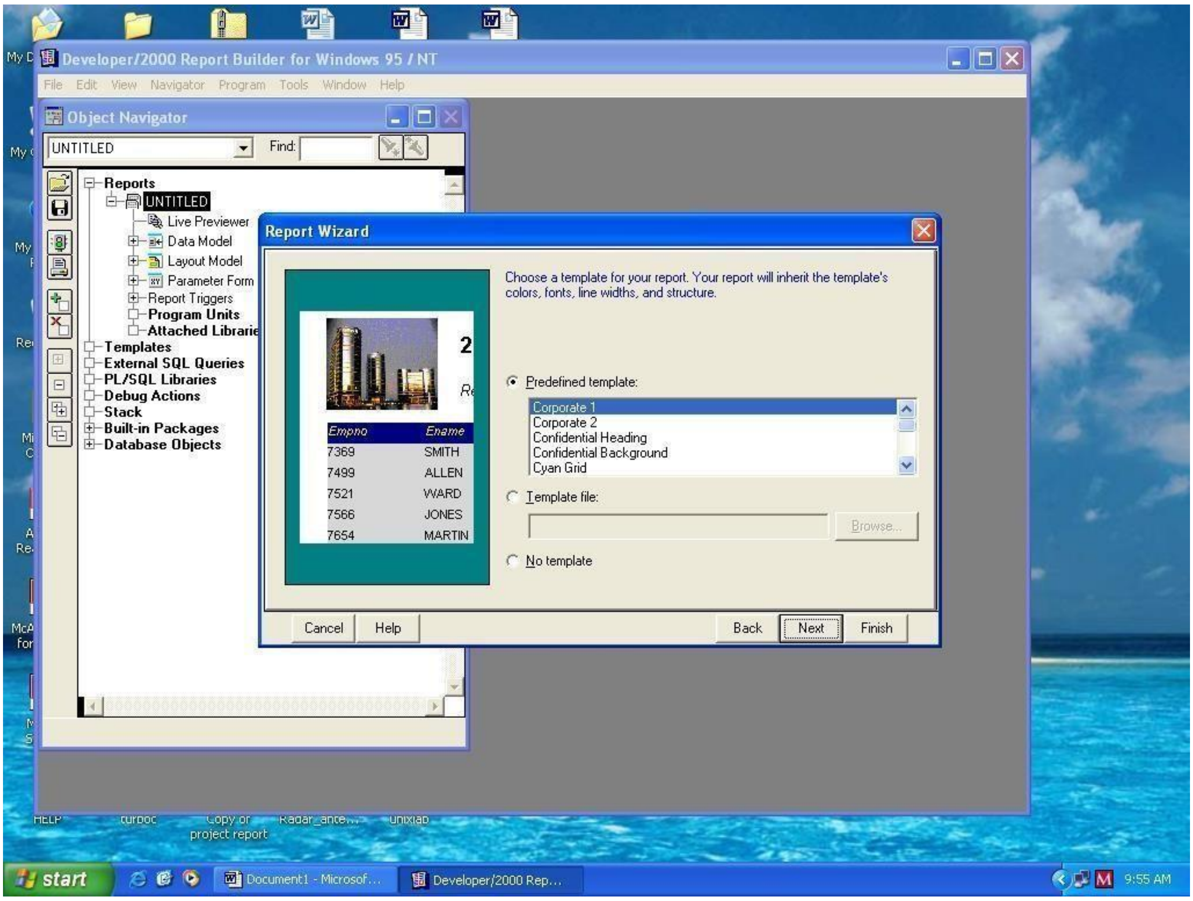
Selecting the columns in the report



Modify the labels and widths (number of characters) for your fields and totals as desired.

Fields and Totals	Labels	Width
EID	Eid	7
ENAME	Ename	10
SALARY	Salary	7
SumEIDPerReport	Total:	7

Modify the labels in a table



Choose a template for your report. Your report will inherit the template's colors, fonts, line widths, and structure.

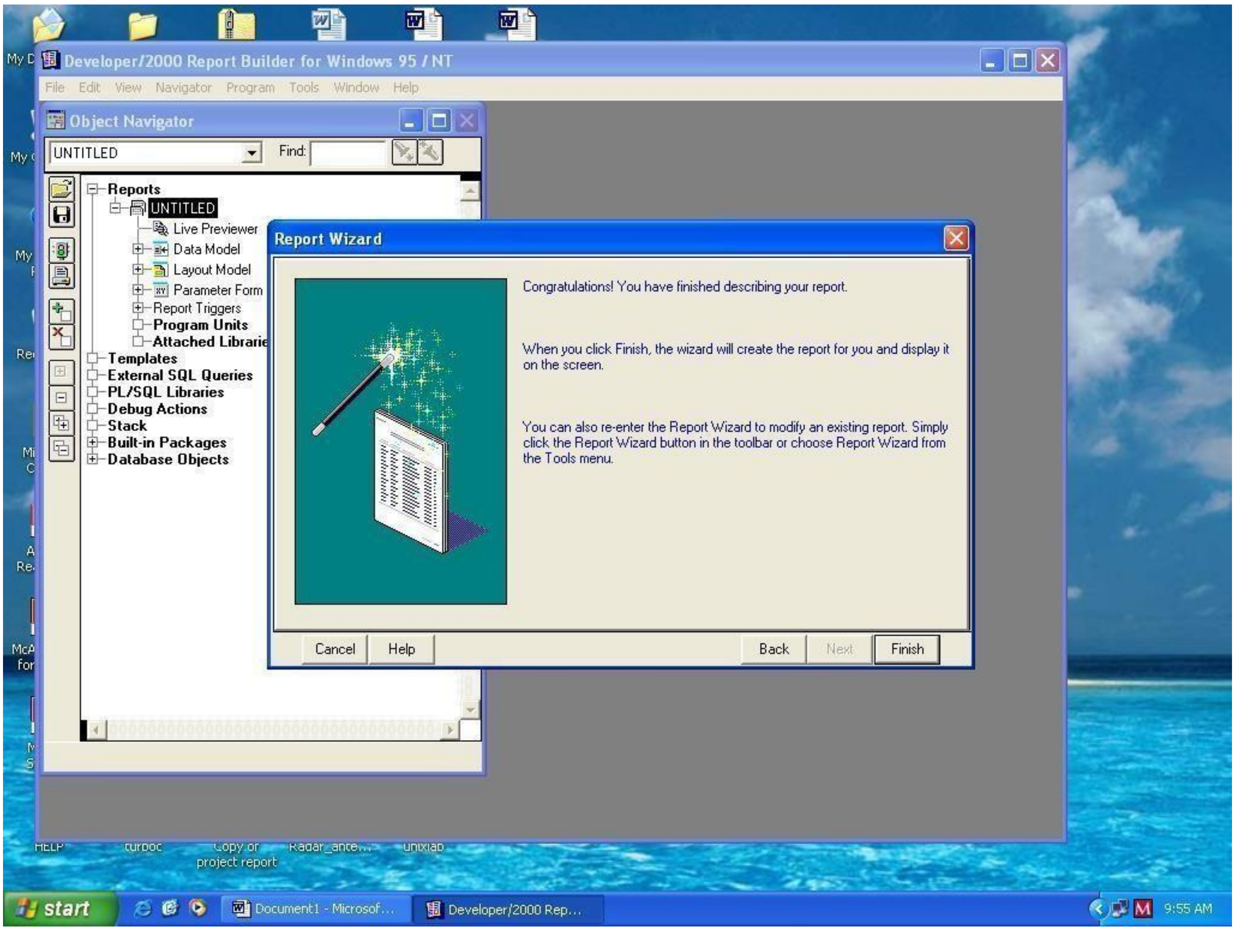
● Predefined template:

Corporate 1
Corporate 2
Confidential Heading
Confidential Background
Cyan Grid

○ Template file:

○ No template

Choose a template to represent the report

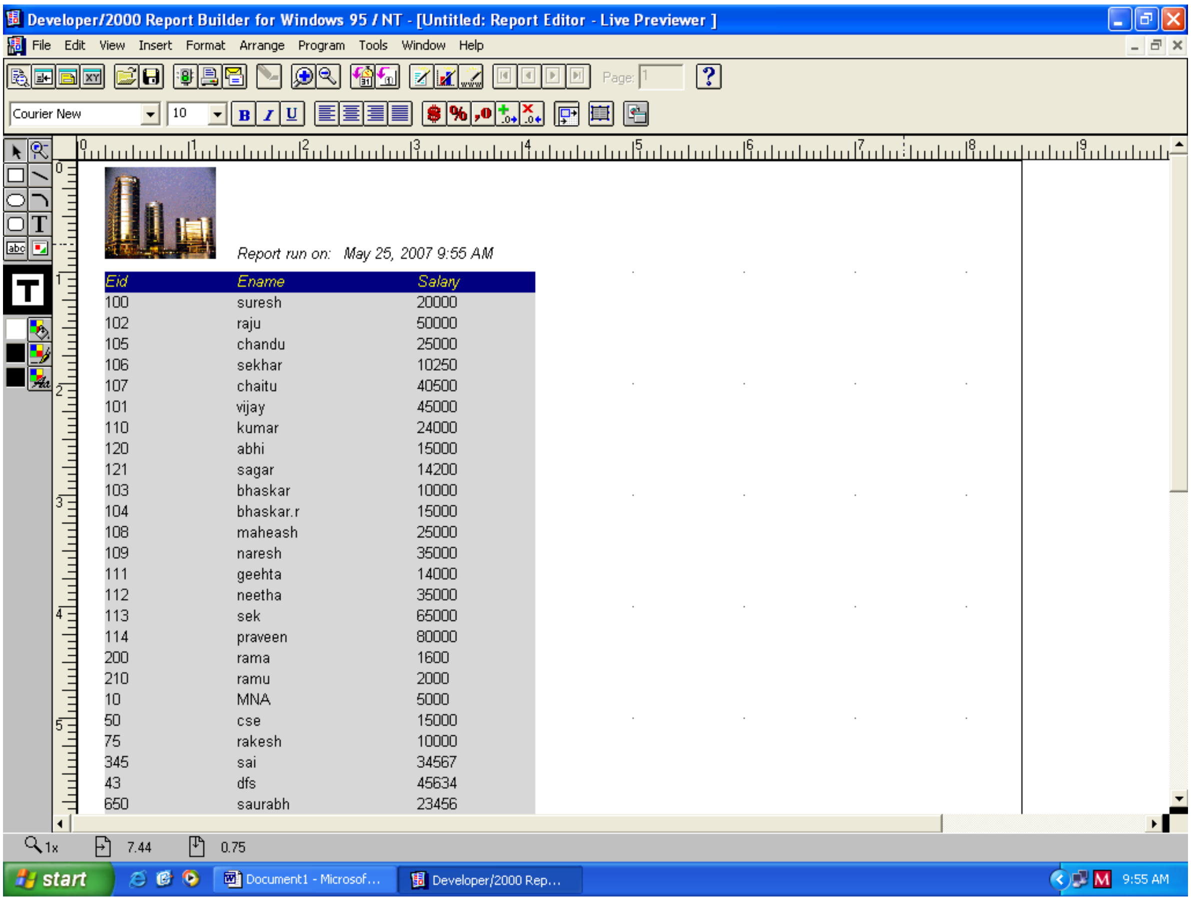


Congratulations! You have finished describing your report.

When you click Finish, the wizard will create the report for you and display it on the screen.

You can also re-enter the Report Wizard to modify an existing report. Simply click the Report Wizard button in the toolbar or choose Report Wizard from the Tools menu.

To specify the completion of report generation


Developer/2000 Report Builder for Windows 95 / NT - [Untitled: Report Editor - Live Previewer]
File Edit View Insert Format Arrange Program Tools Window Help
Courier New
10
Page: 1
1x
7.44
0.75
start
Document1 - Microsof...
Developer/2000 Rep...
9:55 AM

Report run on: May 25, 2007 9:55 AM

Eid	Ename	Salary
100	suresh	20000
102	raju	50000
105	chandu	25000
106	sekhar	10250
107	chaitu	40500
101	vijay	45000
110	kumar	24000
120	abhi	15000
121	sagar	14200
103	bhaskar	10000
104	bhaskar.r	15000
108	maheash	25000
109	naresh	35000
111	geehta	14000
112	neetha	35000
113	sek	65000
114	praveen	80000
200	rama	1600
210	ramu	2000
10	MNA	5000
50	cse	15000
75	rakesh	10000
345	sai	34567
43	dfs	45634
650	saurabh	23456