Experiment No.: 1

Aim :

Define a class ‘product’ with data members pcode, pname and price. Create 3 objects of theclass and find the product having the lowest price.

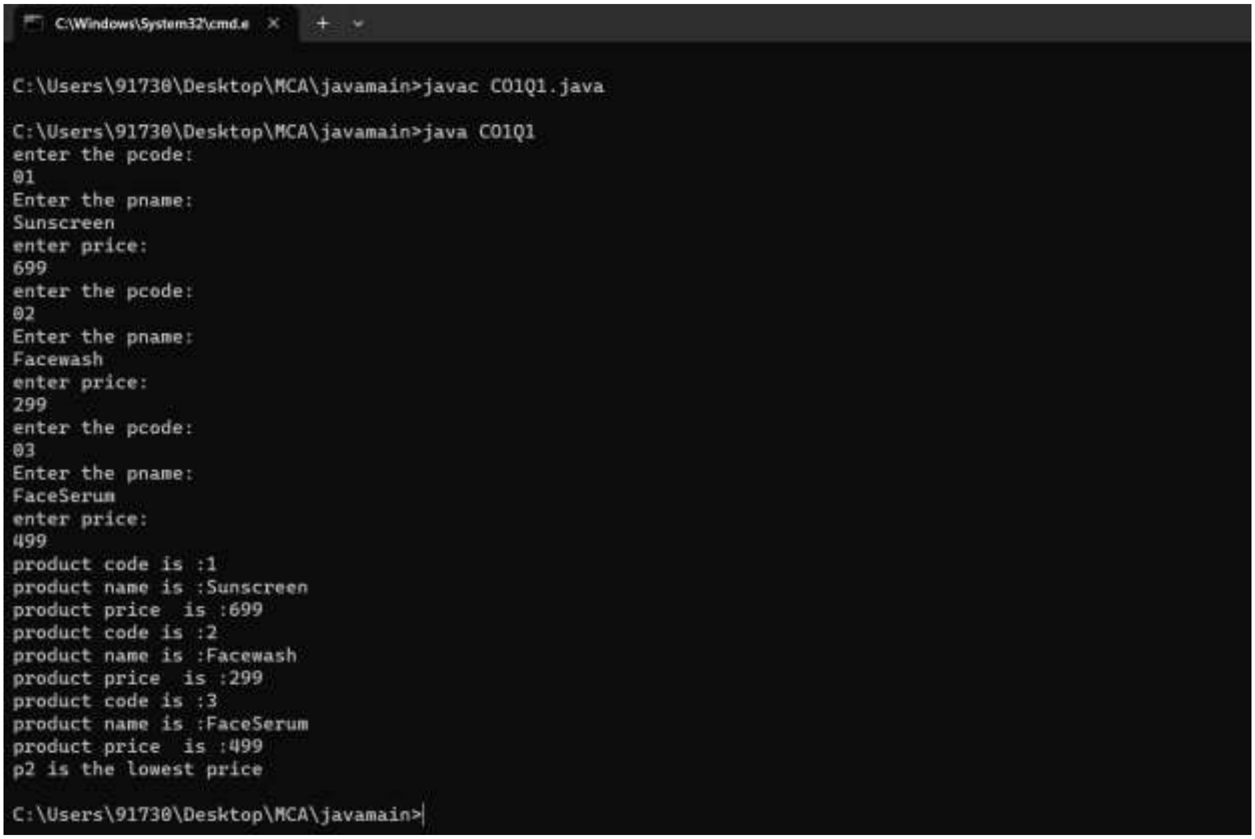
CO1 :

Implement object-oriented concepts like inheritance, overloading and interfaces

Procedure :

```
import java.util.Scanner;

public class CO1Q1 {

        int pcode;

        String pname;

        int price;

public void get() {

        Scanner sc = new Scanner(System.in);

        System.out.println("enter the pcode:");

        pcode = sc.nextInt();

        System.out.println("Enter the pname:");

        pname = sc.nextInt();

        System.out.println("enter price:");

        price = sc.nextInt();                         }

public void put() {

        System.out.println("product code is :"+ pcode);

        System.out.println("product name is :"+ pname);

        System.out.println("product price  is :"+price);

                                                        }
public static void main(String args[]) {

   CO1Q1 p1 = new CO1Q1();
   CO1Q1 p2 = new CO1Q1();
   CO1Q1 p3 = new CO1Q1();

   p1.get();
   p2.get();
   p3.get();

   p1.put();
   p2.put();
   p3.put();

    if(p1.price <p2.price&&p1.price<p3.price)
   {
   System.out.println("p1 is the lowest price");
}
else if(p2.price <p1.price&&p2.price<p3.price)
   {
System.out.println("p2 is the lowest price");
}
  else
   {
 System.out.println("p3 is the lowest price")
 }
}
}
```

Output Screenshot :



```
C:\Users\91730\Desktop\MCA\javamain>javac CO1Q1.java

C:\Users\91730\Desktop\MCA\javamain>java CO1Q1
enter the pcode:
01
Enter the pname:
Sunscreen
enter price:
699
enter the pcode:
02
Enter the pname:
Facewash
enter price:
299
enter the pcode:
03
Enter the pname:
FaceSerum
enter price:
499
product code is :1
product name is :Sunscreen
product price  is :699
product code is :2
product name is :Facewash
product price  is :299
product code is :3
product name is :FaceSerum
product price  is :499
p2 is the lowest price

C:\Users\91730\Desktop\MCA\javamain>
```

Result :

The program was executed and the result was successfully obtained. Thus CO1 was obtained.

Experiment No.: 2

Aim :

Read 2 matrices from the console and performmatrix addition.

CO1 :

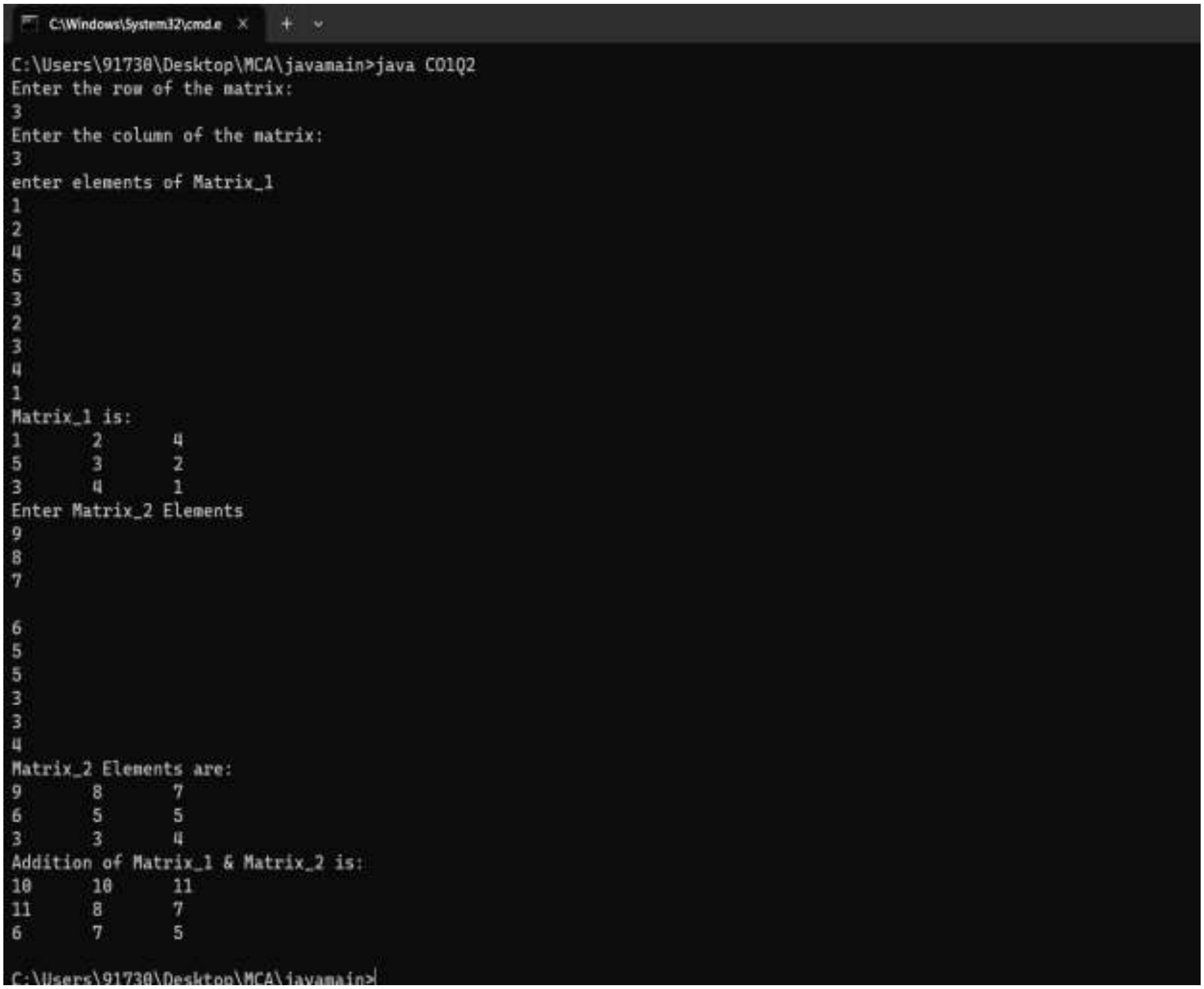
Implement object-oriented concepts like inheritance, overloading and interfaces

Procedure :

```
import java.util.Scanner;
 public class CO1Q2 {
             public static void main(String[] args) {
                          Scanner sc = new Scanner(System.in);
                          int i,j;
                         System.out.println("Enter the row of the matrix:");
                         int row = sc.nextInt();
                         System.out.println("Enter the column of the matrix:");
                         int col = sc.nextInt();
                          int arr1[][]=new int[row][col];
                         System.out.println("enter elements of Matrix_1");
                          for(i=0;i<row;i++) {
                            for(j=0;j<col;j++){
                                                 arr1[i][j] = sc.nextInt(); } }
                         System.out.println("Matrix_1 is:");
                          for (i=0;i<row;i++) {
                        for(j=0;j<col;j++) {
                                                System.out.print(arr1[i][j]+"\t"); }
                                              System.out.println(); }
                        int arr2[][] = new int[row][col];
```

```
                System.out.println("Enter Matrix_2 Elements ");
                for (i=0;i<row;i++)
                { for(j=0;j<col;j++) {
                                arr2[i][j] = sc.nextInt();            }}
                System.out.println("Matrix_2 Elements are: ");
                for (i=0;i<row;i++) {
                for(j=0;j<col;j++){
                                System.out.print(arr2[i][j]+"\t");    }
                                System.out.println();                  }
                int arr3[][] = new int[row][col];
                System.out.println("Addition of Matrix_1 & Matrix_2 is: ");
                for (i=0;i<row;i++) {
                        for(j=0;j<col;j++) {
                                arr3[i][j] = arr1[i][j] + arr2[i][j];
 }
}
                for (i=0;i<row;i++)
 {
                 for(j=0;j<col;j++)
  {
                                System.out.print(arr3[i][j]+"\t");
  }
                                System.out.println();
}
}
}
```

Output Screenshot :



```
C:\Users\91730\Desktop\MCA\javamain>java CO1Q2
Enter the row of the matrix:
3
Enter the column of the matrix:
3
enter elements of Matrix_1
1
2
4
5
3
2
3
4
1
Matrix_1 is:
1       2       4
5       3       2
3       4       1
Enter Matrix_2 Elements
9
8
7

6
5
5
3
3
4
Matrix_2 Elements are:
9       8       7
6       5       5
3       3       4
Addition of Matrix_1 & Matrix_2 is:
10      10      11
11      8       7
6       7       5

C:\Users\91730\Desktop\MCA\javamain>
```

Result :

The program was executed and the result was successfully obtained. Thus CO1 was obtained.

Experiment No.: 3

Aim :

Add complex numbers.

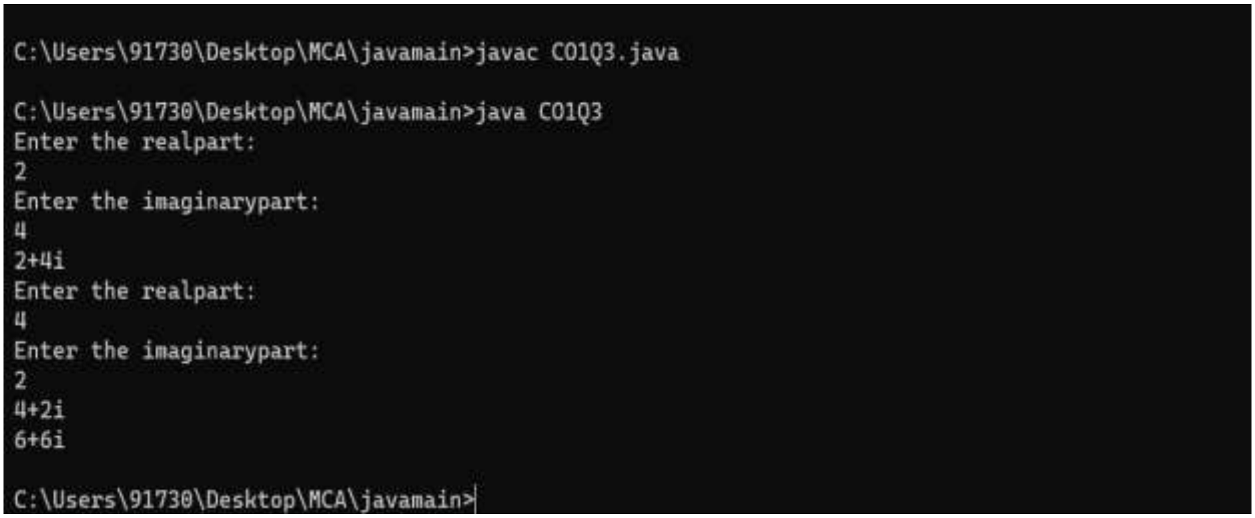
CO1 :

Implement object-oriented concepts like inheritance, overloading and interfaces

Procedure :

```
import java.util.Scanner;
public class CO1Q3 {
    public static void main(String args[]) {
                    Scanner sc =  new Scanner(System.in);
                    System.out.println("Enter the realpart:");
                    int num1 = sc.nextInt();
                    System.out.println("Enter the imaginarypart:");
                    int num2 = sc.nextInt();
                    System.out.println(num1+"+"+num2+"i");
                    System.out.println("Enter the realpart:");
                    int number1 = sc.nextInt();
                   System.out.println("Enter the imaginarypart:");
                    int number2 = sc.nextInt();
                    System.out.println(number1+"+"+number2+"i");
                     System.out.println(num1+number1 + "+" + (num2+number2) + "i");


          }
}
```

Output Screenshot :



```
C:\Users\91730\Desktop\MCA\javamain>javac CO1Q3.java

C:\Users\91730\Desktop\MCA\javamain>java CO1Q3
Enter the realpart:
2
Enter the imaginarypart:
4
2+4i
Enter the realpart:
4
Enter the imaginarypart:
2
4+2i
6+6i

C:\Users\91730\Desktop\MCA\javamain>
```

Result :

The program was executed and the result was successfully obtained. Thus CO1 was obtained.

Experiment No.: 4

Aim :

Read a matrix from the console and check whether it is symmetric or not.

CO1 :

Implement object-oriented concepts like inheritance, overloading and interfaces

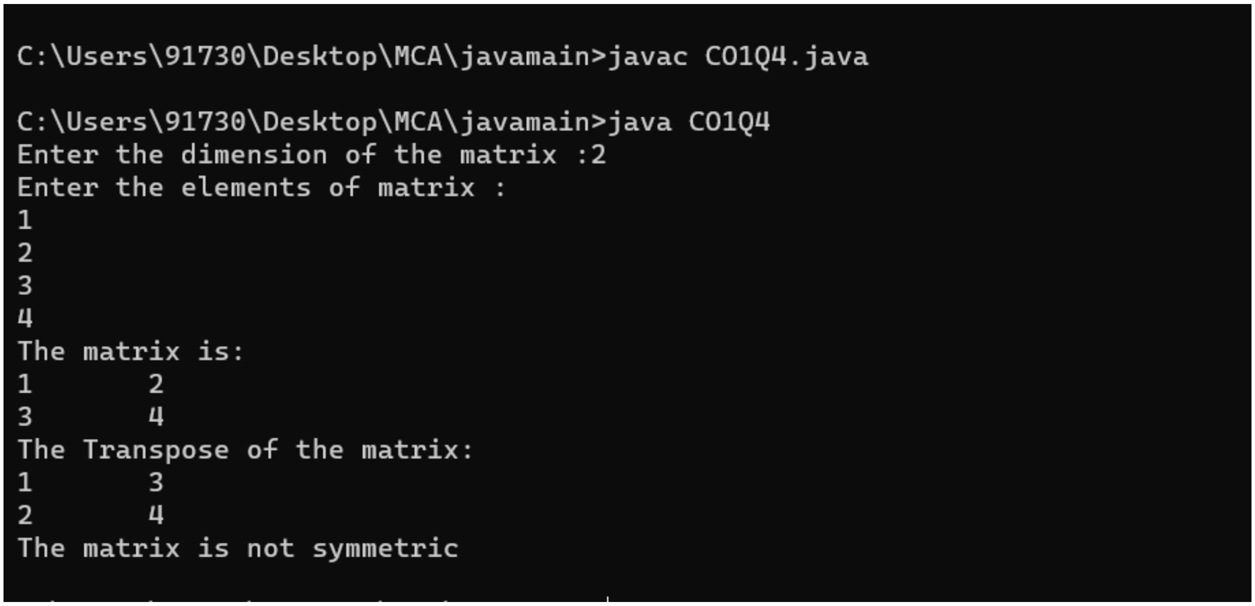
Procedure :

```
import java.util.*;
public class CO1Q4{
                                        //SYMMETRIC MATRIX

        public static void main(String[] args){


                Scanner sc = new Scanner(System.in);
                System.out.print("Enter the dimension of the matrix :");
                int dimension = sc.nextInt();



                int array1[][] = new int[dimension][dimension];
                int array2[][] = new int[dimension][dimension];


                System.out.println("Enter the elements of matrix :");
                for(int i=0;i<dimension;i++){
                        for(int j=0;j<dimension;j++){
                                array1[i][j] = sc.nextInt();
                        }
                }
```

```
System.out.println("The matrix is: ");
for(int i=0;i<dimension;i++){
        for(int j=0;j<dimension;j++){
                System.out.print(array1[i][j] + "         ");
        }
        System.out.println(" ");
}



System.out.println("The Transpose of the matrix: ");
for(int i=0;i<dimension;i++){
        for(int j=0;j<dimension;j++){
                array2[i][j]=array1[j][i];
        }
}
                for(int i=0;i<dimension;i++)
                {
        for(int j=0;j<dimension;j++){
                System.out.print(array2[i][j] + "         ");
        }
        System.out.println(" ");
}
int flag=0;
for(int i=0;i<dimension;i++){
        for(int j=0;j<dimension;j++){
                if(array1[i][j] != array2[i][j]){
                        flag=1;
                        break;
        }
```

```
                    }
            }
            if(flag==1){
                    System.out.println("The matrix is not symmetric");
            }
            else{
                    System.out.println("The matrix is symmetric");
            }

}
}
```

Output Screenshot :



```
C:\Users\91730\Desktop\MCA\javamain>javac CO1Q4.java

C:\Users\91730\Desktop\MCA\javamain>java CO1Q4
Enter the dimension of the matrix :2
Enter the elements of matrix :
1
2
3
4
The matrix is:
1       2
3       4
The Transpose of the matrix:
1       3
2       4
The matrix is not symmetric
```

Result :

The program was executed and the result was successfully obtained. Thus CO1 was obtained.

Experiment No.: 5

Aim :

Program to Sort strings

CO2 :

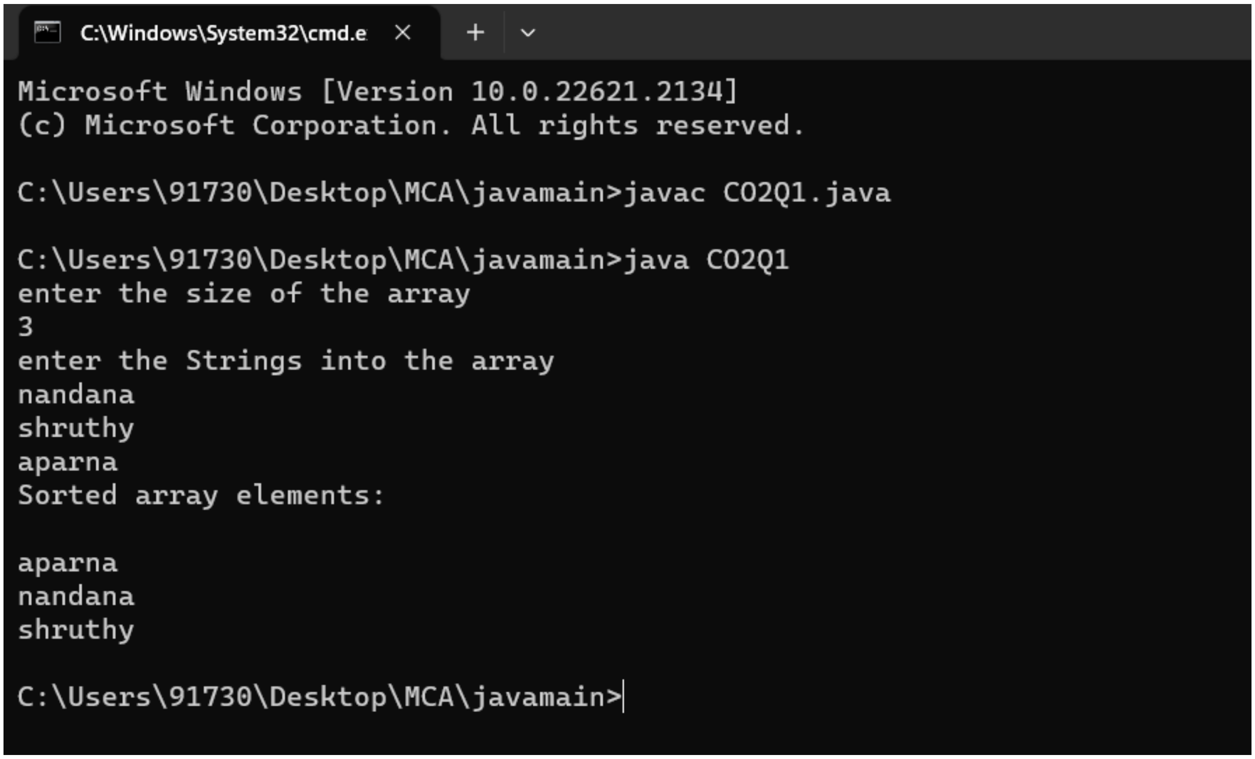
Familiarization and understanding of arrays and strings.

Procedure :

```
import java.util.*;
public class CO2Q1
{
//static Scanner s=new Scanner(System.in);
public static void main(String args[])
{
        Scanner s=new Scanner(System.in);
String temp;
String[] A=new String[8];
int a;
System.out.println("enter the size of the array");
a=s.nextInt();
System.out.println("enter the Strings into the array");
for(int i=0;i<=a;i++)
{
A[i]=s.nextLine();
}
System.out.println("Sorted array elements:");
for(int i=0;i<=a;i++)
{
for(int j=i+1;j<=a;j++){
```

```
if(A[i].compareTo(A[j])>0)
{
temp=A[i];
A[i]=A[j];
A[j]=temp;
}}}
for(int i=0;i<=a;i++)
{
System.out.println(A[i]); } } }
```

Output Screenshot :



```
Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

C:\Users\91730\Desktop\MCA\javamain>javac CO2Q1.java

C:\Users\91730\Desktop\MCA\javamain>java CO2Q1
enter the size of the array
3
enter the Strings into the array
nandana
shruthy
aparna
Sorted array elements:

aparna
nandana
shruthy

C:\Users\91730\Desktop\MCA\javamain>
```

Result :

The program was executed and the result was successfully obtained. Thus CO2 was obtained.

Experiment No.: 6

Aim :

Search an element in an array.

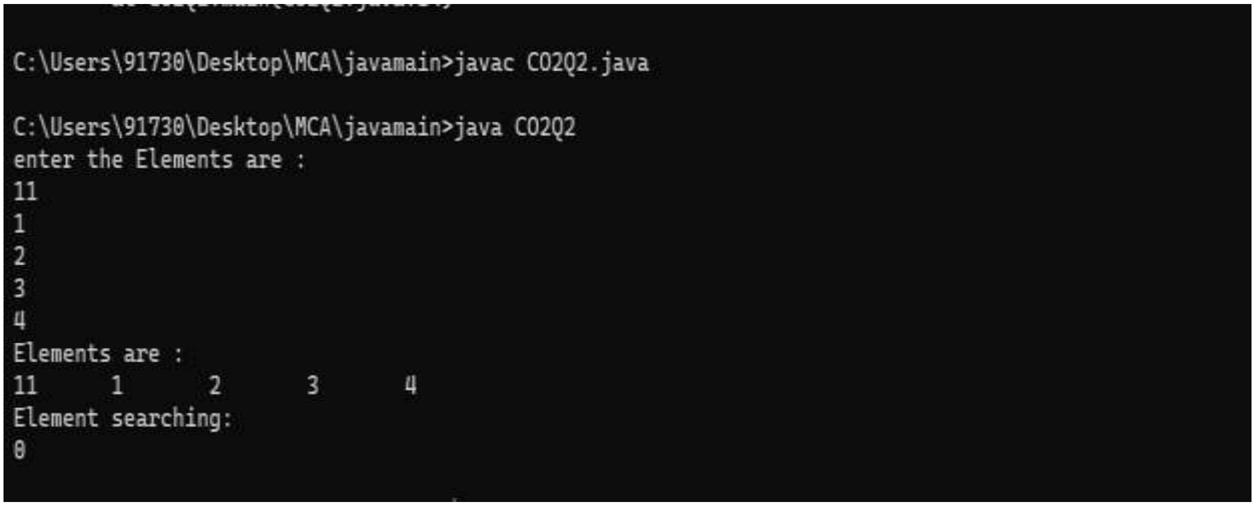
CO2:

Familiarization and understanding of arrays and strings.

Procedure :

```
import java.util.Scanner;
public class CO2Q2{
                                //search an element in an array
        public static void main(String args[]){
                Scanner s = new Scanner(System.in);
                int i ;
                 int arr[]=new int[5];
                        System.out.println("enter the Elements are :");
                for(i=0;i<arr.length;i++)
                {
                        arr[i] = s.nextInt();
                }
                System.out.println("Elements are :");
                for(i=0;i<arr.length;i++)
                {
                        System.out.print(arr[i]+"\t");
                }
                System.out.println("\nElement searching:");
                int val = s.nextInt();
                for(i=0;i<arr.length;i++)
                {
                        if(arr[i]==val)
                        {
                                System.out.println("value found");
                                System.out.println("Position of element is :"+(i+1));
                        }}}}
```

Output Screenshot :



```
C:\Users\91730\Desktop\MCA\javamain>javac CO2Q2.java

C:\Users\91730\Desktop\MCA\javamain>java CO2Q2
enter the Elements are :
11
1
2
3
4
Elements are :
11      1       2       3       4
Element searching:
0
```

Result :

The program was executed and the result was successfully obtained. Thus CO2 was obtained.

Experiment No.: 7

Aim :

Perform string manipulations

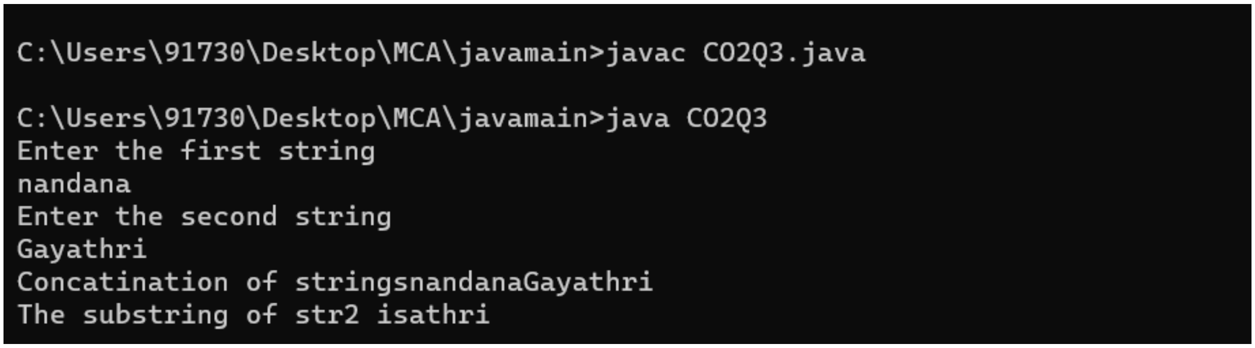
CO2:

Familiarization and understanding of arrays and strings.

Procedure :

```
import java.util.Scanner;

public class CO2Q3 {


                                        //string manipulation

  public static void main(String[] args) {

    Scanner object = new Scanner(System.in);


        System.out.println("Enter the first string");


         String str1 = object.nextLine();


        System.out.println("Enter the second string");


        String str2 = object.nextLine();


                    System.out.println("Concatination of strings"+ str1.concat(str2));


                    System.out.println("The substring of str2 is"+ str2.substring(3));


  }}
```

Output Screenshot :



```
C:\Users\91730\Desktop\MCA\javamain>javac CO2Q3.java

C:\Users\91730\Desktop\MCA\javamain>java CO2Q3
Enter the first string
nandana
Enter the second string
Gayathri
Concatination of stringsnandanaGayathri
The substring of str2 isathri
```

Result :

The program was executed and the result was successfully obtained. Thus CO2 was obtained.

Experiment No.: 8

Aim :

Program to create a class for Employee having attributes eNo, eName eSalary. Read n employ information and Search for an employee given eNo, using the concept of Array of Objects.

CO2 :

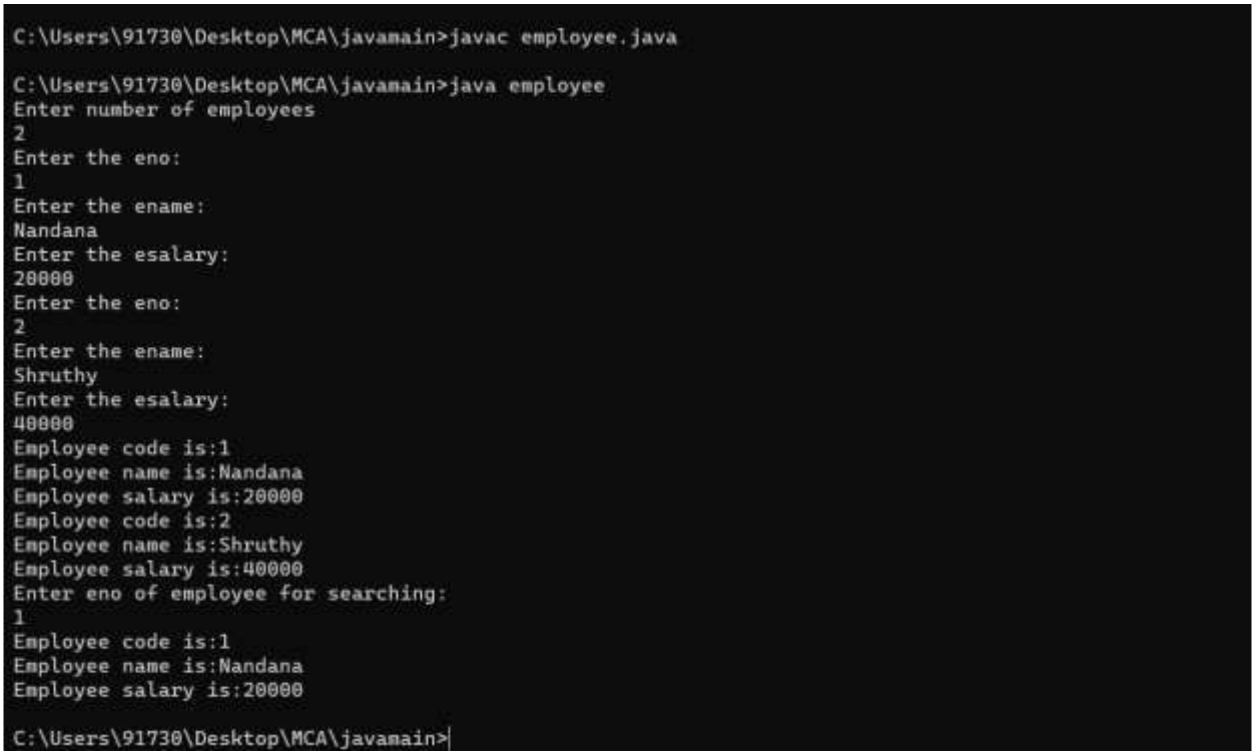
Familiarization and understanding of arrays and strings.

Procedure :

```
import java.util.Scanner;
public class employee{
        int eno;
        String ename;
        int esalary;
public void get(){
Scanner sc=  new Scanner(System.in);
System.out.println("Enter the eno:");
eno=sc.nextInt();
System.out.println("Enter the ename:");
ename=sc.next();
System.out.println("Enter the esalary:");
esalary=sc.nextInt();
}
public void put(){
        System.out.println("Employee code is:"+eno);
        System.out.println("Employee name is:"+ename);
        System.out.println("Employee salary is:"+esalary);
}
public static void main(String args[]){
```

```
Scanner sc = new Scanner(System.in);
 System.out.println("Enter number of employees");
  int n=sc.nextInt();
 employee e[] = new employee[n];
 for(int i=0;i<n;i++)
 {
             e[i] = new employee();
             e[i].get();
 }
 for(int i=0;i<n;i++)
 {
             e[i].put();
 }
 System.out.println("Enter eno of employee for searching:");
 int value  = sc.nextInt();
 int flag=0;
 for(int i=0;i<n;i++)
 {
             if(e[i].eno==value)
             {
             zflag=1;
                         e[i].put();
                         break;              }}
 if(flag==0)
 {
             System.out.println("not found");
 }}}
```

Output Screenshot :



```
C:\Users\91730\Desktop\MCA\javamain>javac employee.java

C:\Users\91730\Desktop\MCA\javamain>java employee
Enter number of employees
2
Enter the eno:
1
Enter the ename:
Nandana
Enter the esalary:
20000
Enter the eno:
2
Enter the ename:
Shruthy
Enter the esalary:
40000
Employee code is:1
Employee name is:Nandana
Employee salary is:20000
Employee code is:2
Employee name is:Shruthy
Employee salary is:40000
Enter eno of employee for searching:
1
Employee code is:1
Employee name is:Nandana
Employee salary is:20000

C:\Users\91730\Desktop\MCA\javamain>
```

Result :

The program was executed and the result was successfully obtained. Thus CO2 was obtained.

Experiment No.: 9

Aim :

Area of different shapes using overloaded functions

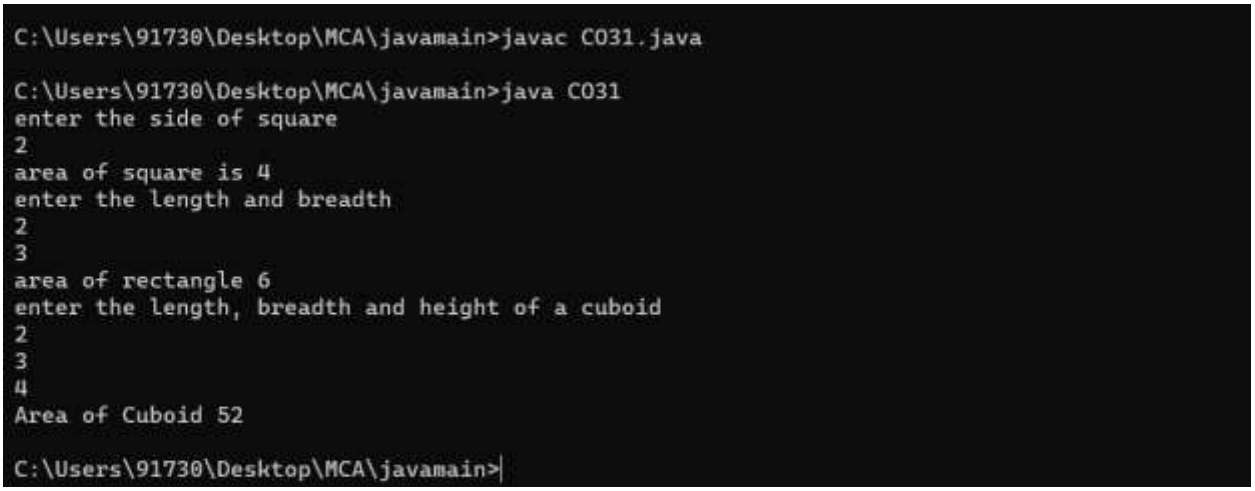
CO3 :

Understand and implement object-oriented concepts like inheritance, overloading and interfaces.

Procedure :

```
import java.util.Scanner;


class areaShapes{
    void area(int a){
        System.out.println("area of square is "+a*a);
    }
    void area(int a, int b){
        System.out.println("area of rectangle "+a*b);
    }
    void area(int length, int breadth, int height){
        System.out.println("Area of Cuboid
"+(2*(length*breadth)+2*(length*height)+2*(height*breadth)));
 }}
public class CO31 {
    public static void main(String[] args) {
        int a,b,c;
        Scanner s= new Scanner(System.in);
        areaShapes obj=new areaShapes();
        System.out.println("enter the side of square");
        a= s.nextInt();
        obj.area(a);
        System.out.println("enter the length and breadth");
        a=s.nextInt();
        b=s.nextInt();
        obj.area(a,b);
        System.out.println("enter the length, breadth and height of a cuboid");
        a=s.nextInt();
        b=s.nextInt();
        c=s.nextInt();
        obj.area(a,b,c);
    }}
```

Output Screenshot :



```
C:\Users\91730\Desktop\MCA\javamain>javac CO31.java

C:\Users\91730\Desktop\MCA\javamain>java CO31
enter the side of square
2
area of square is 4
enter the length and breadth
2
3
area of rectangle 6
enter the length, breadth and height of a cuboid
2
3
4
Area of Cuboid 52

C:\Users\91730\Desktop\MCA\javamain>
```

Result :

The program was executed and the result was successfully obtained. Thus CO3 was obtained.

Experiment No.: 10

Aim :

Create a class ‘Employee’ with data members Empid, Name, Salary, Address and constructors to initialize the data members. Create another class ‘Teacher’ that inherit the properties of class employee and contain its own data members department, Subjects taught and constructors to initialize these data members and also include display function to display all the data members. Use array of objects to display details of N teachers.

CO3 :

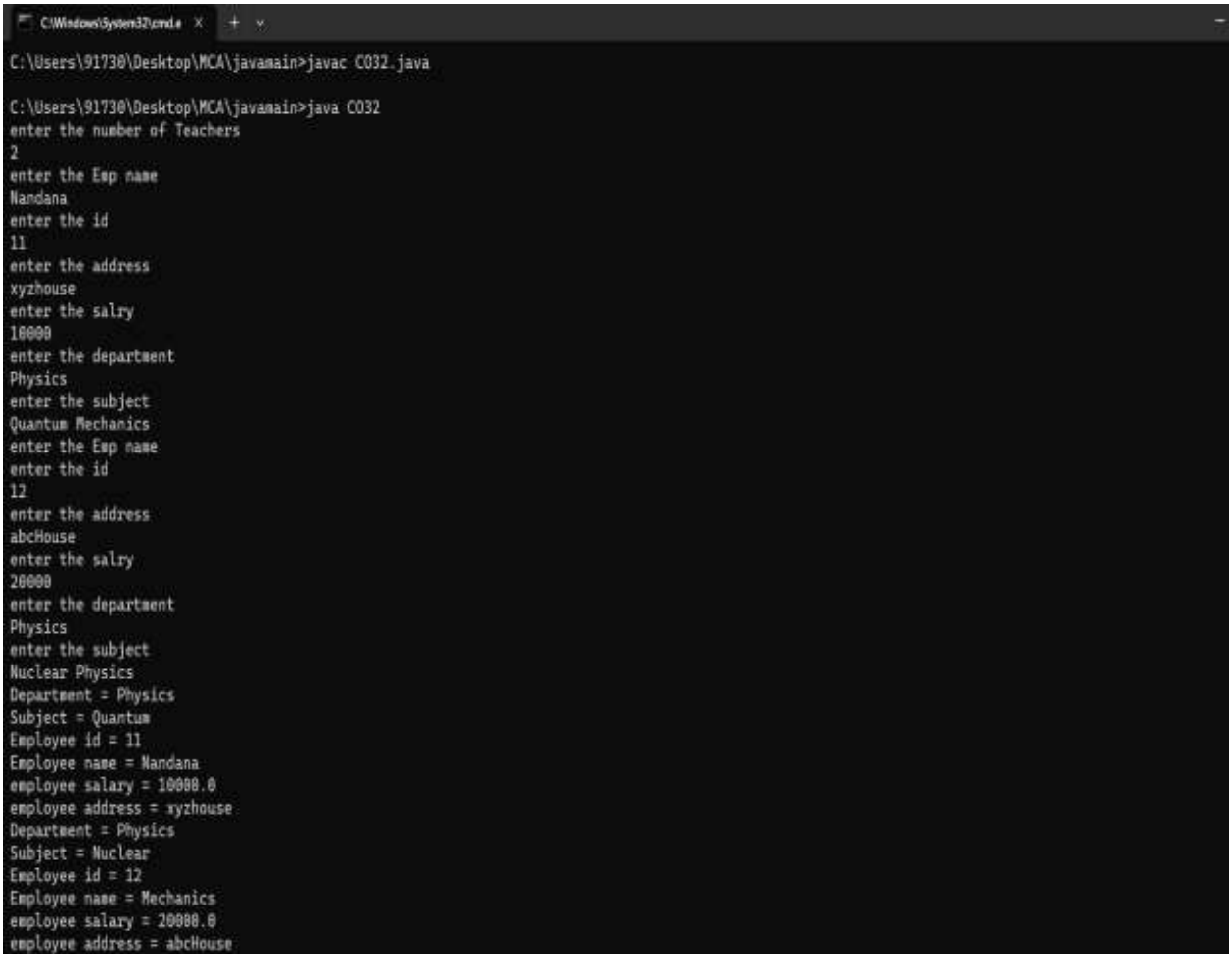
Understand and implement object-oriented concepts like inheritance, overloading and interfaces.

Procedure :

```
import java.util.Scanner;
class Employee {
    int e_id;
    String e_name, e_address;
    double e_sal;
  Employee(int id, String name, double sal, String address) {
        e_id = id;
        e_name = name;
        e_address = address;
        e_sal = sal;                                                  }
         void display() {
        System.out.println("Employee name = " + e_name);
        System.out.println("employee salary = " + e_sal);
        System.out.println("employee address = " + e_address);                   }}
class Teacher extends Employee {
    String t_dept, t_sub;
 Teacher(String dept, String sub, String name, int id, String address, double sal) {
        super(id, name, sal, address)
        t_dept = dept;
        t_sub = sub;                              }
        void display() {
        System.out.println("Department = " + t_dept);
        System.out.println("Subject = " + t_sub);
        System.out.println("Employee id = " + e_id);
        System.out.println("Employee name = " + e_name);
        System.out.println("employee salary = " + e_sal);
        System.out.println("employee address = " + e_address)                    }}
public class CO32 {
    public static void main(String[] args) {
        int count, id;
        String name, dept, sub, address;
        double sal;
        Scanner s = new Scanner(System.in);
        System.out.println("enter the number of Teachers");
        count = s.nextInt();
        Teacher[] obj = new Teacher[count];
        for (int i = 0; i < count; i++) {
            System.out.println("enter the Emp name");
            name = s.next();
            System.out.println("enter the id");
            id = s.nextInt();
            System.out.println("enter the address");
            address = s.next();
            System.out.println("enter the salry");
            sal = s.nextInt();
            System.out.println("enter the department");
            dept = s.next();
```

```
      System.out.println("enter the subject");
      sub = s.next();
      obj[i] = new Teacher(dept, sub, name, id, address, sal) ;        }
        for (int i = 0; i < count; i++) {
      obj[i].display();
    }}}
```

Output Screenshot :



```
C:\Users\91730\Desktop\MCA\javamain>javac CO32.java

C:\Users\91730\Desktop\MCA\javamain>java CO32
enter the number of Teachers
2
enter the Emp name
Nandana
enter the id
11
enter the address
xyzhouse
enter the salry
10000
enter the department
Physics
enter the subject
Quantum Mechanics
enter the Emp name
enter the id
12
enter the address
abcHouse
enter the salry
20000
enter the department
Physics
enter the subject
Nuclear Physics
Department = Physics
Subject = Quantum
Employee id = 11
Employee name = Nandana
employee salary = 10000.0
employee address = xyzhouse
Department = Physics
Subject = Nuclear
Employee id = 12
Employee name = Mechanics
employee salary = 20000.0
employee address = abcHouse
```

Result :

The program was executed and the result was successfully obtained. Thus CO3 was obtained.

Experiment No.: 11

Aim :

Create a class ‘Person’ with data members Name, Gender, Address, Age and a constructor to initialize the data members and another class ‘Employee’ that inherits the properties of class Person and also contains its own data members like Empid, Company_name,Qualification, Salary and its own constructor. Create another class ‘Teacher’ that inherits the properties of class Employee and contains its own data members like Subject, Department, Teacherid and also contain constructors and methods to display the data members. Use array of objects to display details of N teachers.

CO3 :

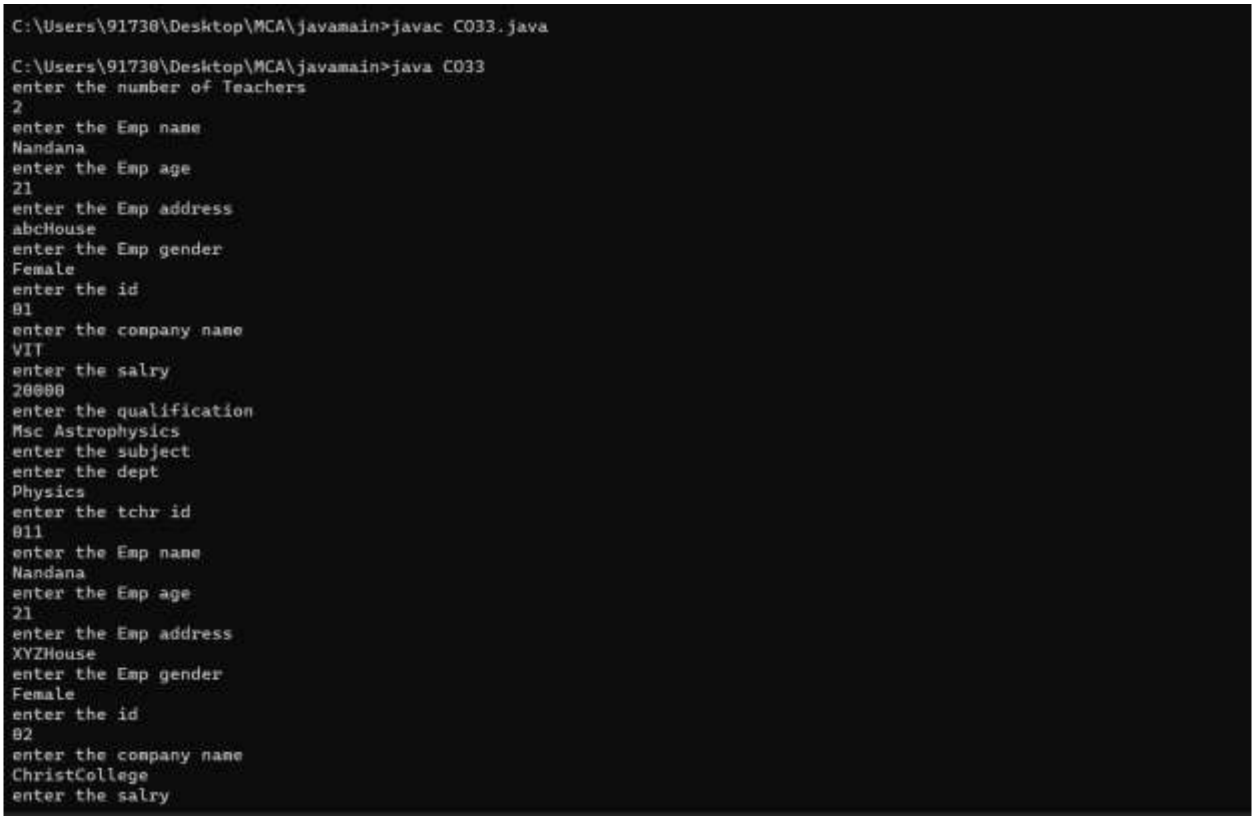
Understand and implement object-oriented concepts like inheritance, overloading and interfaces.

Procedure :

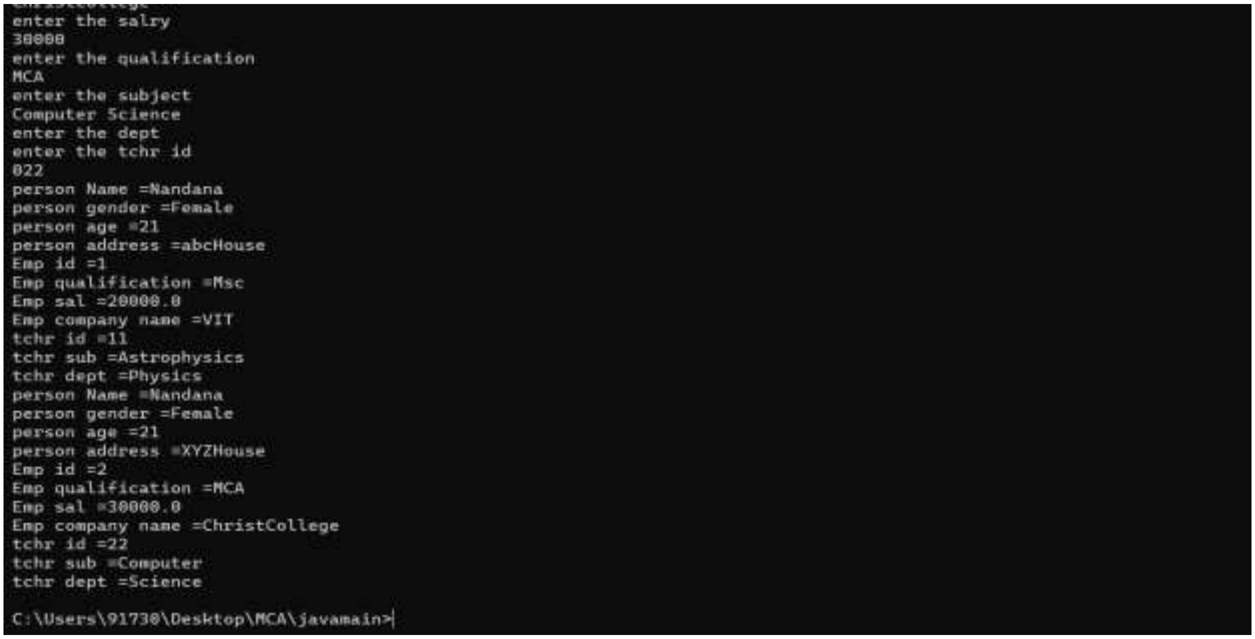
```
import java.util.Scanner;
class Person1{
   int p_age;
   String p_name,p_address,p_gender;
   Person1(String name,String address, int age, String gender){
      p_age=age;
      p_name=name;
      p_address=address;
      p_gender=gender;                                           }   }
class Employee1 extends Person1{
   String e_c_name, e_qualification;
   int e_id;
   double e_sal;
   Employee1(int id, String qualification, String comany_name, double sal, String name,String address, int age, String gender){
      super(name,address,age,gender);
      e_c_name=comany_name;
```

```
        e_qualification=qualification;
        e_id=id;
        e_sal=sal;                                    }}
class Teacher1 extends Employee1{
    String t_dept,t_sub;
    int t_id;
    Teacher1(String dept, String sub, int tr_id, int id, String qualification, String comany_name,
double sal, String name,String address, int age, String gender){
        super(id, qualification, comany_name, sal, name, address, age, gender);
        t_dept=dept;
        t_sub=sub;
        t_id=tr_id;                          }
  void display(){
        System.out.println("person Name ="+p_name);
        System.out.println("person gender ="+p_gender);
        System.out.println("person age ="+p_age);
        System.out.println("person address ="+p_address);
        System.out.println("Emp id ="+e_id);
        System.out.println("Emp qualification ="+e_qualification);
        System.out.println("Emp sal ="+e_sal);
        System.out.println("Emp company name ="+e_c_name);
        System.out.println("tchr id ="+t_id);
        System.out.println("tchr sub ="+t_sub);
        System.out.println("tchr dept ="+t_dept);                    }}
public class CO33 {
    public static void main(String[] args){
        int count,e_id,t_id,p_age;
        String p_name,t_dept,t_sub,p_address,p_gender,qualification,c_name;
        double sal;
        Scanner s=new Scanner(System.in);
    System.out.println("enter the number of Teachers");
    count=s.nextInt();
 Teacher1 [] obj=new Teacher1[count];
 for(int i=0;i<count;i++){
        System.out.println("enter the Emp name");
        p_name=s.next();
        System.out.println("enter the Emp age");
        p_age=s.nextInt();
        System.out.println("enter the Emp address");
        p_address=s.next();
        System.out.println("enter the Emp gender");
        p_gender=s.next();
        System.out.println("enter the id");
        e_id=s.nextInt();
        System.out.println("enter the company name");
        c_name=s.next();
        System.out.println("enter the salry");
        sal=s.nextDouble();
        System.out.println("enter the qualification");
        qualification=s.next();
        System.out.println("enter the subject");
        t_sub=s.next();
        System.out.println("enter the dept");
        t_dept=s.next();
        System.out.println("enter the tchr id");
        t_id=s.nextInt();
        obj[i]=new Teacher1(t_dept,t_sub,t_id, e_id,qualification,c_name, sal,p_name,p_address,
p_age,p_gender); }
 for(int i=0;i<count;i++){
        obj[i].display();                         } }}
```

Output Screenshot :



```
C:\Users\91730\Desktop\MCA\javamain>javac CO33.java

C:\Users\91730\Desktop\MCA\javamain>java CO33
enter the number of Teachers
2
enter the Emp name
Nandana
enter the Emp age
21
enter the Emp address
abcHouse
enter the Emp gender
Female
enter the id
01
enter the company name
VIT
enter the salry
20000
enter the qualification
Msc Astrophysics
enter the subject
enter the dept
Physics
enter the tchr id
011
enter the Emp name
Nandana
enter the Emp age
21
enter the Emp address
XYZHouse
enter the Emp gender
Female
enter the id
02
enter the company name
ChristCollege
enter the salry
```



```
enter the salry
30000
enter the qualification
MCA
enter the subject
Computer Science
enter the dept
enter the tchr id
022
person Name =Nandana
person gender =Female
person age =21
person address =abcHouse
Emp id =1
Emp qualification =Msc
Emp sal =20000.0
Emp company name =VIT
tchr id =11
tchr sub =Astrophysics
tchr dept =Physics
person Name =Nandana
person gender =Female
person age =21
person address =XYZHouse
Emp id =2
Emp qualification =MCA
Emp sal =30000.0
Emp company name =ChristCollege
tchr id =22
tchr sub =Computer
tchr dept =Science

C:\Users\91730\Desktop\MCA\javamain>
```

Result :

The program was executed and the result was successfully obtained. Thus CO3 was obtained.

Experiment No.: 12

Aim :

Write a program has class Publisher, Book, Literature and Fiction. Read the information and print the details of books from either the category, using inheritance.

CO3:

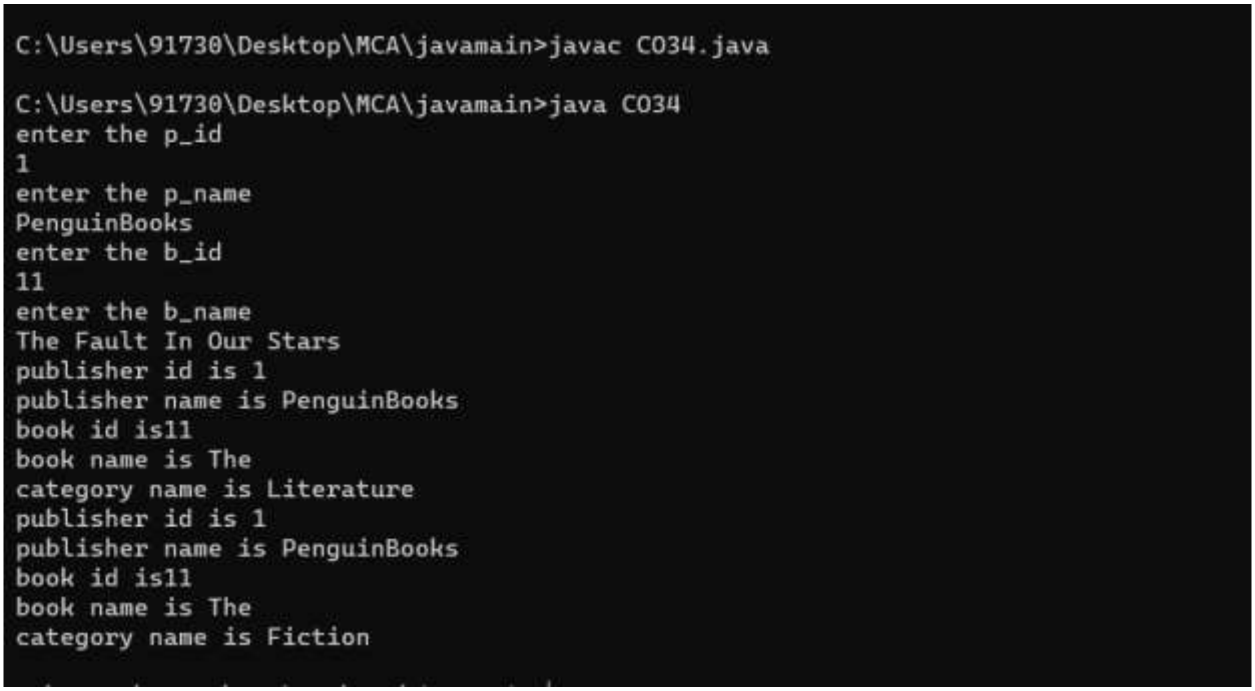
Understand and implement object-oriented concepts like inheritance, overloading and interfaces.

Procedure :

```
import java.util.Scanner;
class Publisher{
   int p_id;
   String p_name;
   Publisher(int p_id,String p_name){
      this.p_id=p_id;
      this.p_name=p_name;                    }}
class Book extends Publisher{
   int b_id;
   String b_name;
   Book(int p_id, String p_name, int b_id, String b_name){
      super(p_id, p_name);
      this.b_id=b_id;
      this.b_name=b_name;
   }}
class Literature extends Book{
   String cat;
   Literature(int p_id, String p_name, int b_id, String b_name){
      super(p_id, p_name, b_id, b_name);
      this.cat="Literature";                                       }
```

```
    void Display4(){
        System.out.println("publisher id is "+this.p_id);
        System.out.println("publisher name is "+ this.p_name);
        System.out.println("book id is" +this.b_id);
        System.out.println("book name is "+ this.b_name);
        System.out.println("category name is "+this.cat);
    }}
class Fiction extends Book{
    String cat;
    Fiction(int p_id, String p_name, int b_id, String b_name){
        super(p_id, p_name, b_id, b_name);
        this.cat="Fiction";                                         }
    void Display4(){
        System.out.println("publisher id is "+this.p_id);
        System.out.println("publisher name is "+ this.p_name);
        System.out.println("book id is" +this.b_id);
        System.out.println("book name is "+ this.b_name);
        System.out.println("category name is "+this.cat);
    }}
public class CO34 {
    public static void main(String[] args){
        int p_id, b_id;
        String p_name, b_name;
        Scanner s=new Scanner(System.in);
        System.out.println("enter the p_id");
        p_id=s.nextInt();
        System.out.println("enter the p_name");
        p_name=s.next();
        System.out.println("enter the b_id");
        b_id=s.nextInt();
        System.out.println("enter the b_name");
        b_name=s.next();
        Literature lit=new Literature(p_id, p_name, b_id, b_name);
        Fiction fic=new Fiction(p_id, p_name, b_id, b_name);
        lit.Display4();
        fic.Display4();
     }}
```

Output Screenshot :



```
C:\Users\91730\Desktop\MCA\javamain>javac CO34.java

C:\Users\91730\Desktop\MCA\javamain>java CO34
enter the p_id
1
enter the p_name
PenguinBooks
enter the b_id
11
enter the b_name
The Fault In Our Stars
publisher id is 1
publisher name is PenguinBooks
book id is11
book name is The
category name is Literature
publisher id is 1
publisher name is PenguinBooks
book id is11
book name is The
category name is Fiction
```

Result :

The program was executed and the result was successfully obtained. Thus CO3 was obtained.

Experiment No.: 13

Aim :

Create classes Student and Sports. Create another class Result inherited from Student and Sports. Display the academic and sports score of a student.

CO3:

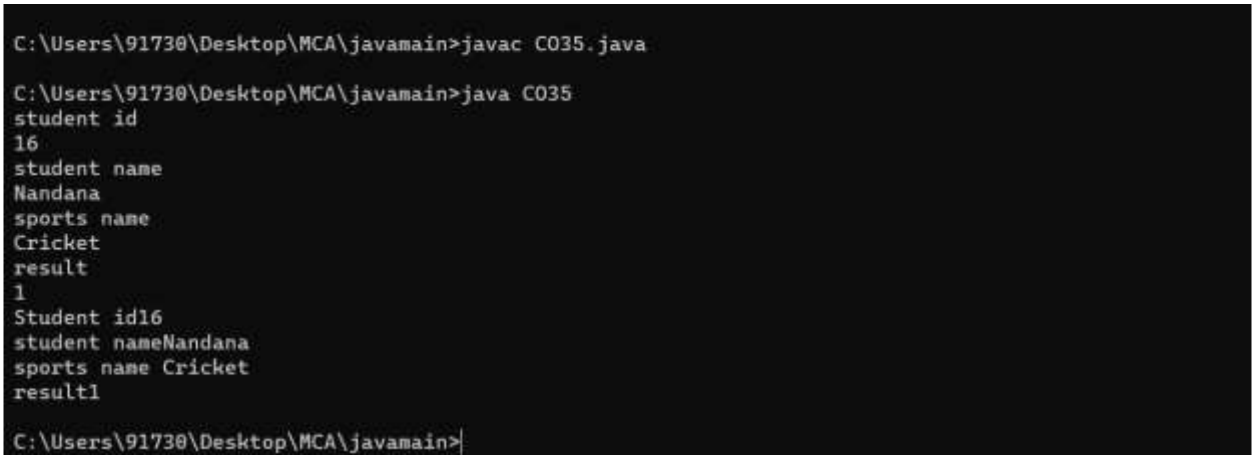
Understand and implement object-oriented concepts like inheritance, overloading and interfaces.

Procedure :

```
import java.util.Scanner;
class Student{
   int s_id;
   String s_name;
   Student(String name,int id){
      s_id=id;
      s_name=name;
   }   }
class Sports extends Student{
   String sp_name;
   Sports(String name, int id, String s_name){
      super(name,id);
      sp_name=s_name;
  }}
class Result extends Sports{
   String res;
   Result(String name, int id, String s_name, String result){
      super(name, id, s_name);
      res=result;
        }
```

```
 void display5(){
        System.out.println("Student id" +s_id);
        System.out.println("student name"+ s_name);
        System.out.println("sports name "+sp_name);
        System.out.println("result"+ res);
 }
}
public class CO35 {
    public static void main(String[] args){
        int id;
        String name,s_name,result;
        Scanner s=new Scanner(System.in);
        System.out.println("student id ");
        id=s.nextInt();
        System.out.println("student name ");
        name=s.next();
        System.out.println("sports name ");
        s_name=s.next();
        System.out.println("result ");
        result=s.next();
        Result obj = new Result(name,id, s_name,result);
        obj.display5();
    }
}
```

Output Screenshot :



```
C:\Users\91730\Desktop\MCA\javamain>javac CO35.java

C:\Users\91730\Desktop\MCA\javamain>java CO35
student id
16
student name
Nandana
sports name
Cricket
result
1
Student id16
student nameNandana
sports name Cricket
result1

C:\Users\91730\Desktop\MCA\javamain>
```

Result :

The program was executed and the result was successfully obtained. Thus CO3 was obtained.

Experiment No.: 14

Aim :

Create an interface having prototypes of functions area() and perimeter(). Create two classes Circle and Rectangle which implements the above interface. Create a menu driven program to find area and perimeter of objects.

CO3:

Understand and implement object-oriented concepts like inheritance, overloading and interfaces.

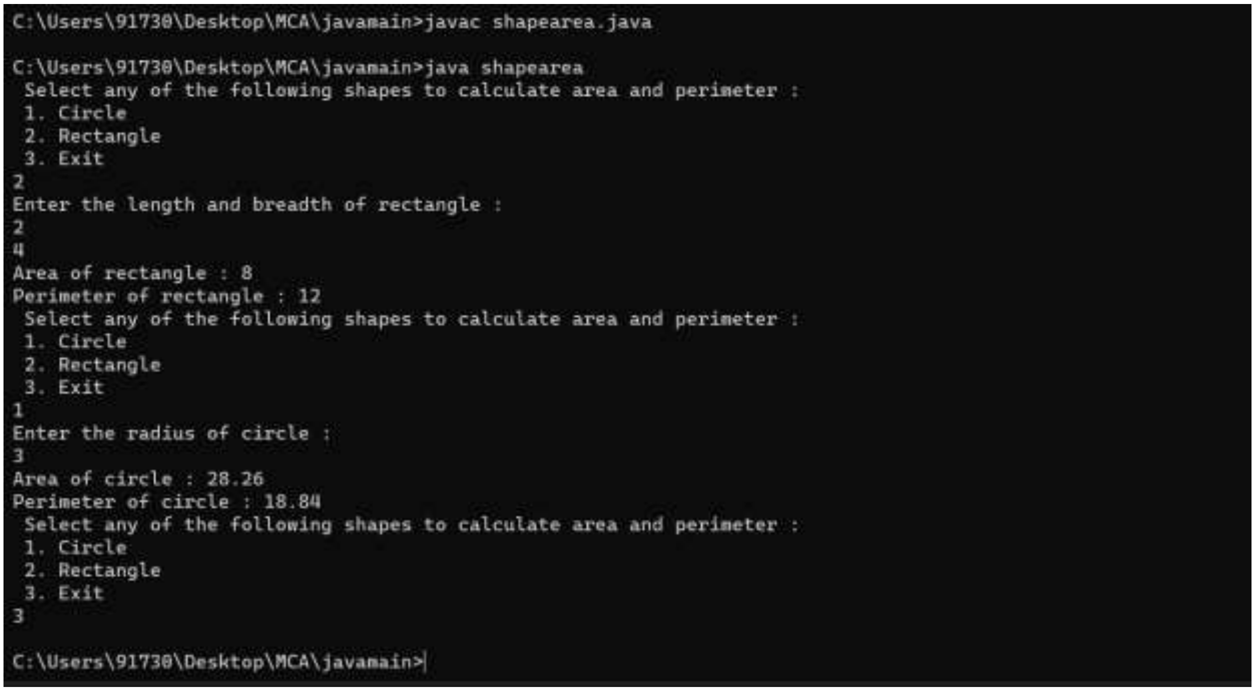
Procedure :

```
import java.util.*;
interface Proto
{
        void area();
        void peri();
}
class Rectangle implements Proto
{       int l,b,a,p;
        public void area()
        {
                Scanner in = new Scanner(System.in);
                System.out.println("Enter the length and breadth of rectangle : ");
                l = in.nextInt();
                b = in.nextInt();
                a=l*b;
                System.out.println("Area of rectangle : "+a);
        }
        public void peri()
        {
```

```
		p=2*(l+b);
		System.out.println("Perimeter of rectangle : "+p);
	}
}
class Circle implements Proto
{
	float r,a,p;
	public void area()
	{
		Scanner in = new Scanner(System.in);
		System.out.println("Enter the radius of circle : ");
		r = in.nextFloat();
		a=3.14f*(r*r);
		System.out.println("Area of circle : "+a);
	}
	public void peri()
	{
		p=(2*3.14f)*r;
		System.out.println("Perimeter of circle : "+p);
	}
}
public class shapearea
{
	public static void main(String args[])
	{
		Scanner in = new Scanner(System.in);
		int op=1;
		do
		{
```

```
            System.out.println(" Select any of the following shapes to calculate area
and perimeter : ");
            System.out.println(" 1. Circle ");
            System.out.println(" 2. Rectangle ");
            System.out.println(" 3. Exit ");
            int ch=in.nextInt();
            switch(ch)
            {
                case 1 :Circle c = new Circle();
                        c.area();
                        c.peri();
                        break;
                case 2 :Rectangle r = new Rectangle();
                        r.area();
                        r.peri();
                        break;
                case 3 :System.exit(0);
                        break;
                default : System.out.println("Invalid choice ");
            }
        }
        while(op!=0);
    }
}
```

Output Screenshot :



```
C:\Users\91730\Desktop\MCA\javamain>javac shapearea.java

C:\Users\91730\Desktop\MCA\javamain>java shapearea
 Select any of the following shapes to calculate area and perimeter :
 1. Circle
 2. Rectangle
 3. Exit
2
Enter the length and breadth of rectangle :
2
4
Area of rectangle : 8
Perimeter of rectangle : 12
 Select any of the following shapes to calculate area and perimeter :
 1. Circle
 2. Rectangle
 3. Exit
1
Enter the radius of circle :
3
Area of circle : 28.26
Perimeter of circle : 18.84
 Select any of the following shapes to calculate area and perimeter :
 1. Circle
 2. Rectangle
 3. Exit
3

C:\Users\91730\Desktop\MCA\javamain>
```

Result :

The program was executed and the result was successfully obtained. Thus CO3 was obtained.

Experiment No.: 15

Aim :

Prepare bill with the given format using calculate method from interface.

Order No.

Date :

ProductId	Name	Quantity	unitprice	Total
101	A	2	25	50
102	B	1	100	100

Net.Amount 150

CO3 :

Understand and implement object-oriented concepts like inheritance, overloading and interfaces.

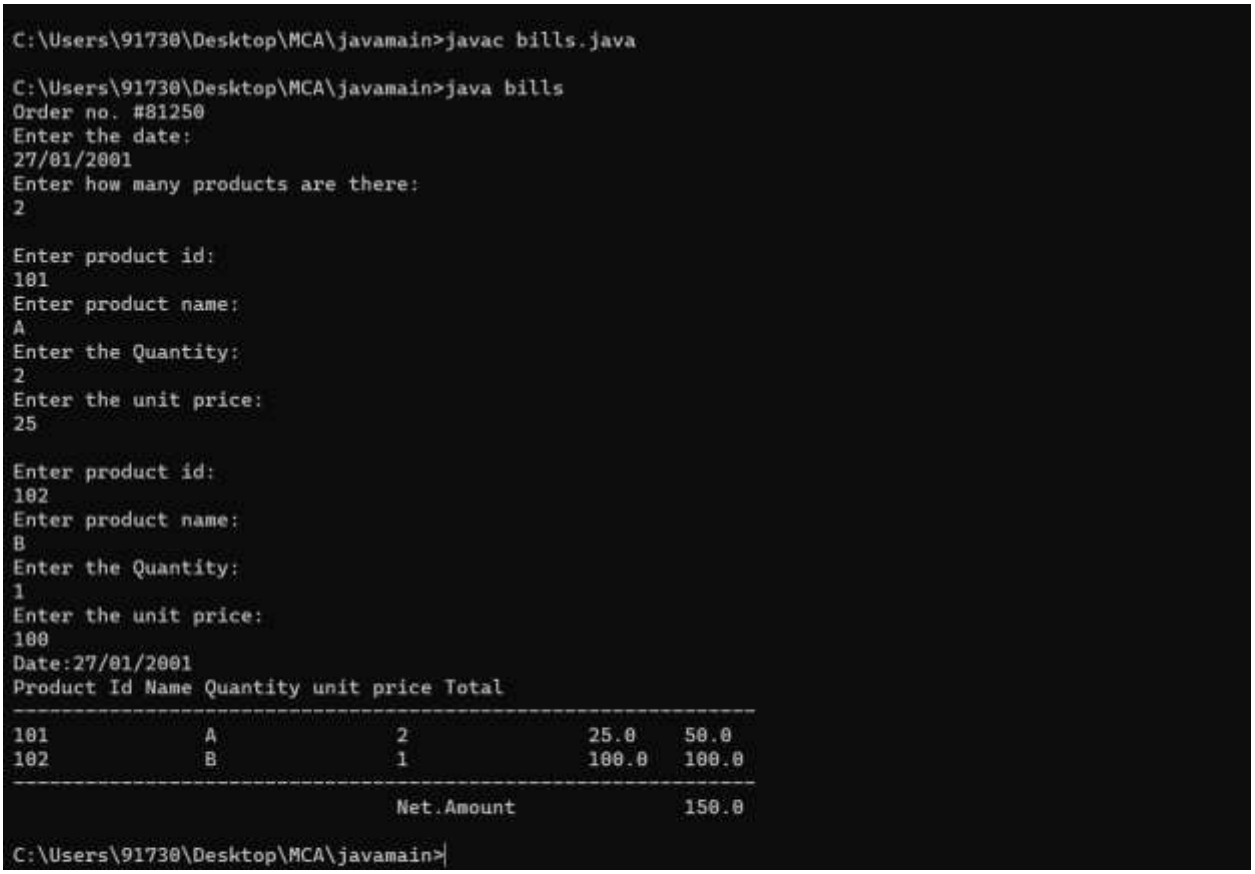
Procedure :

```
import java.util.Scanner;
interface calc
{
void calculate();
}
class bill implements calc
{
String date,name,p_id;
int quantity;
double unit_price,total,namount;
Scanner sc = new Scanner(System.in);
public void getdata()
{
System.out.println("\nEnter product id:");
p_id = sc.nextLine();
```

```
System.out.println("Enter product name:");
name = sc.nextLine();
System.out.println("Enter the Quantity:");
quantity = sc.nextInt();
System.out.println("Enter the unit price:");
unit_price = sc.nextDouble();
}
public void calculate()
{
total = quantity * unit_price;
}
public void display()
{
System.out.println(p_id+"\t\t"+name+"\t\t"+quantity+"\t\t"+unit_price+"\t"+total);
}
}
public class bills
{
public static void main(String[] args)
{
int n,i;
double namount=0,t;
int ran;
String date;
t = Math.random() *1000000;
ran = (int) t;
Scanner sc = new Scanner(System.in);
System.out.println("Order no. #"+ran);
System.out.println("Enter the date:");
```

```
date = sc.nextLine();
System.out.println("Enter how many products are there:");
n = sc.nextInt();
bill ob[] = new bill[n];
for(i=0;i<n;i++)
ob[i] = new bill();
for(i=0;i<n;i++){
ob[i].getdata();
ob[i].calculate();
}
System.out.println("Date:"+date);
System.out.println("Product Id Name Quantity unit price Total ");
System.out.println("----------------------------------------------------------------");
for(i=0;i<n;i++){
ob[i].display();
namount += ob[i].total;
}
System.out.println("----------------------------------------------------------------");
System.out.println("\t\t\t\tNet.Amount\t\t"+ namount);
}}
```

Output Screenshot :



```
C:\Users\91730\Desktop\MCA\javamain>javac bills.java

C:\Users\91730\Desktop\MCA\javamain>java bills
Order no. #81250
Enter the date:
27/01/2001
Enter how many products are there:
2

Enter product id:
101
Enter product name:
A
Enter the Quantity:
2
Enter the unit price:
25

Enter product id:
102
Enter product name:
B
Enter the Quantity:
1
Enter the unit price:
100
Date:27/01/2001
Product Id Name Quantity unit price Total
--------------------------------------------------------------
101             A               2               25.0    50.0
102             B               1               100.0   100.0
--------------------------------------------------------------
                                Net.Amount              150.0

C:\Users\91730\Desktop\MCA\javamain>
```

Result :

The program was executed and the result was successfully obtained. Thus CO3was obtained.

Experiment No.: 16

Aim :

Create a Graphics package that has classes and interfaces for figures Rectangle, Triangle, Square and Circle. Test the package by finding the area of these figures.

CO4 :

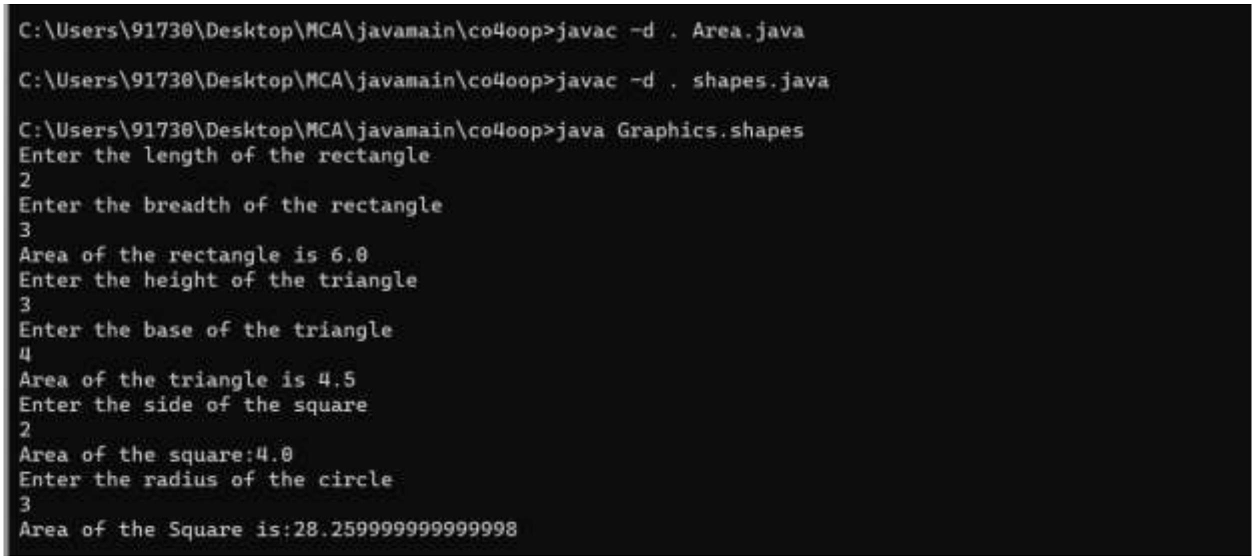
Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

Procedure :

```
package graphics;
import java.util.*;
interface shapes{
   public double RecArea();
   public double CircArea();
   public double SquareArea();
   public double TriangArea();
}
public class Graphics implements shapes {
    Scanner obj = new Scanner(System.in);
    int r,l,b,s;
    double pi = 3.14, area;
    public  double RecArea(){
        System.out.print("Enter the Length of Rectangle: ");
        l=obj.nextInt();
        System.out.print("Enter the Breadth of Rectangle: ");
        b=obj.nextInt();
        area=l*b;
        return area;
    }
    public double CircArea(){
        System.out.print("Enter the Radius of Circle: ");
        r =obj.nextInt();
        area = pi * r * r;
        return area;
    }
    public double SquareArea(){
        System.out.print("Enter the Side of the Square: ");
        s = obj.nextInt();
        area = s * s;
        return area;
    }
    public double TriangArea(){
        System.out.print("Enter the Width of the Triangle: ");
        double base = obj.nextDouble();
        System.out.print("Enter the Height of the Triangle: ");
        double height = obj.nextDouble();
        double area = (base* height)/2;
        return area;
    }
}

import graphics.Graphics;
import java.util.*;
```

```
public class Areas{
    public static void main(String []args){
        Scanner sc = new Scanner(System.in);
        Graphics Obj = new Graphics();
        int choice = 0;
        while(choice != 5){
            System.out.println("-------AREAS OF SHAPES--------\n1. Rectangle\n2.
Circle\n3. Square\n4. Triangle\n5. Exit");

            System.out.print("Enter your choice: ");
            choice = sc.nextInt();
            switch(choice){
                case 1:
                    System.out.println("Area of Rectangle: " + Obj.RecArea());
                    break;
                case 2:
                    System.out.println("Area of Circle: " + Obj.CircArea());
                    break;
                case 3:
                    System.out.println("Area of Square: " + Obj.SquareArea());
                    break;
                case 4:
                    System.out.println("Area of Triangle: " + Obj.TriangArea());
                    break;
                case 5:
                    System.exit(0);
                    break;
                default:
                    System.out.println("Select a valid option!");
            }}}}
```

Output Screenshot :



```
C:\Users\91730\Desktop\MCA\javamain\co4oop>javac -d . Area.java

C:\Users\91730\Desktop\MCA\javamain\co4oop>javac -d . shapes.java

C:\Users\91730\Desktop\MCA\javamain\co4oop>java Graphics.shapes
Enter the length of the rectangle
2
Enter the breadth of the rectangle
3
Area of the rectangle is 6.0
Enter the height of the triangle
3
Enter the base of the triangle
4
Area of the triangle is 4.5
Enter the side of the square
2
Area of the square:4.0
Enter the radius of the circle
3
Area of the Square is:28.259999999999998
```

Result :

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

Experiment No.: 17

Aim :

Create an Arithmetic package that has classes and interfaces for the 4 basic arithmetic operations. Test the package by implementing all operations on two given numbers.

CO4 :

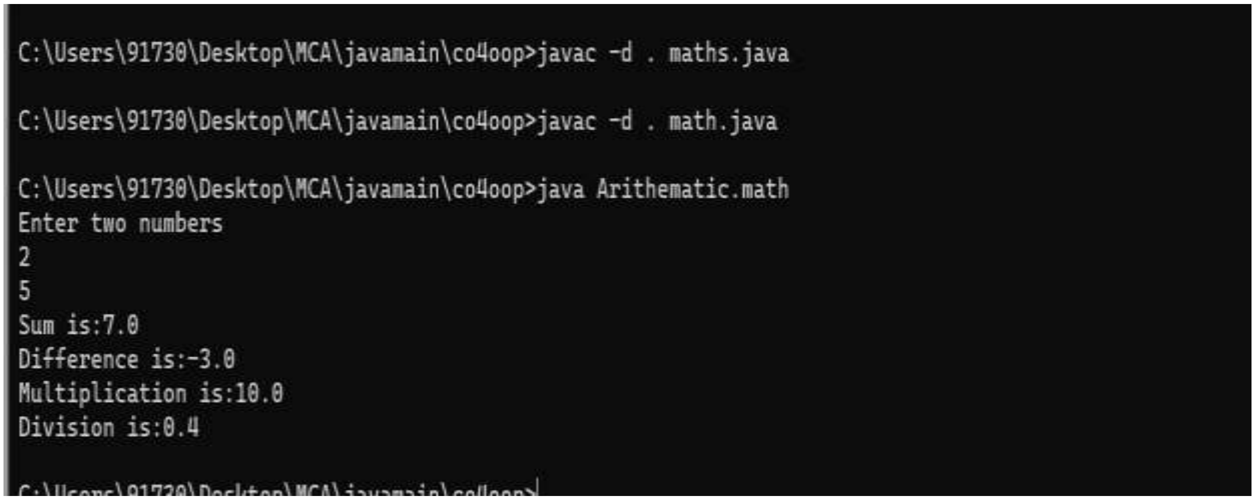
Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

Procedure :

```
package Arithematic;
interface operations
{
        public void input();
        public void addition();
        public void subtract();
        public void multiply();
        public void division();
}
package Arithematic;
import java.util.*;
public class basic implements operations
{
        double a,b,add,diff,mul,div;
        public void input()
        {
                Scanner sc=new Scanner(System.in);
                System.out.println("Enter two numbers");
                a=sc.nextInt();
                b=sc.nextInt();


        }
        public void addition()
        {
                add=a+b;
                System.out.println("Sum is:"+add);


        }
        public void subtract()
        {
                diff=a-b;
                System.out.println("Difference is:"+diff);
        }
        public void multiply()
        {
                mul=a*b;
                System.out.println("Multiplication is:"+mul);
        }
        public void division()
        {
                div=a/b;
                System.out.println("Division is:"+div);
```

```
    }
    public static void main(String args[])
    {
        basic b=new basic();
        b.input();
        b.addition();
        b.subtract();
        b.multiply();
        b.division();
    }
}
```

<u>Output Screenshot</u> :



```
C:\Users\91730\Desktop\MCA\javamain\co4oop>javac -d . maths.java

C:\Users\91730\Desktop\MCA\javamain\co4oop>javac -d . math.java

C:\Users\91730\Desktop\MCA\javamain\co4oop>java Arithematic.math
Enter two numbers
2
5
Sum is:7.0
Difference is:-3.0
Multiplication is:10.0
Division is:0.4

C:\Users\91730\Desktop\MCA\javamain\co4oop>
```

<u>Result</u> :

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

Experiment No.: 18

Aim :

Write a user defined exception class to authenticate the user name and password.

CO4:

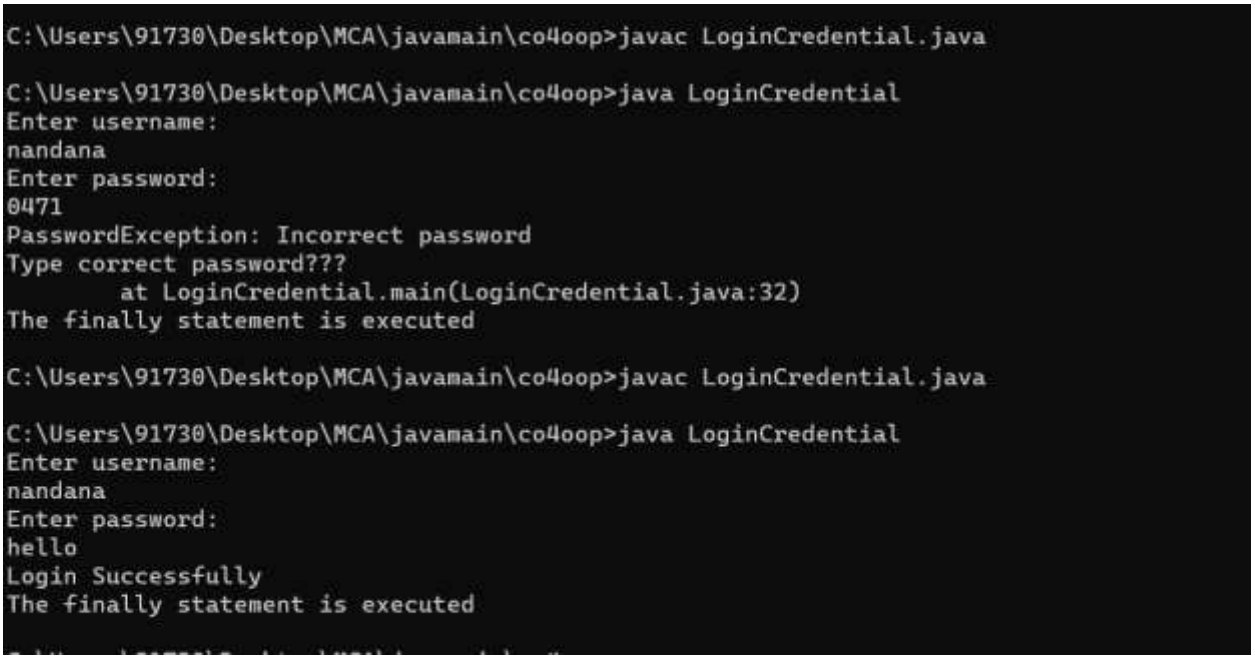
Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

Procedure :

```
import java.util.*;

class UsernameException extends Exception

{

        public UsernameException(String msg)

        {

                super(msg);

        }

}

class PasswordException extends Exception

{

        public PasswordException(String msg)

        {

                super(msg);

        }

}

public class LoginCredential

{

        public static void main(String args[])

        {
```

```
Scanner sc=new Scanner(System.in);
String username,password;
System.out.println("Enter username:");
username=sc.nextLine();
System.out.println("Enter password:");
password=sc.nextLine();
int length=username.length();
try
{
if(length<6)
throw new UsernameException("Username must be greater than 6 characters???");
else if(!password.equals("hello"))
throw new PasswordException("Incorrect password\nType correct password???");
else
System.out.println("Login Successfully");
}
catch(UsernameException u)
{
u.printStackTrace();
}
catch(PasswordException p)
{
p.printStackTrace();
}
finally
{
System.out.println("The finally statement is executed");
}
}
```

Output Screenshot :



```
C:\Users\91730\Desktop\MCA\javamain\co4oop>javac LoginCredential.java

C:\Users\91730\Desktop\MCA\javamain\co4oop>java LoginCredential
Enter username:
nandana
Enter password:
0471
PasswordException: Incorrect password
Type correct password???
        at LoginCredential.main(LoginCredential.java:32)
The finally statement is executed

C:\Users\91730\Desktop\MCA\javamain\co4oop>javac LoginCredential.java

C:\Users\91730\Desktop\MCA\javamain\co4oop>java LoginCredential
Enter username:
nandana
Enter password:
hello
Login Successfully
The finally statement is executed
```

Result :

The program was executed and the result was successfully obtained. Thus CO1 was obtained.

Experiment No.: 19

Aim :

Find the average of N positive integers, raising a user defined exception for each negative input.

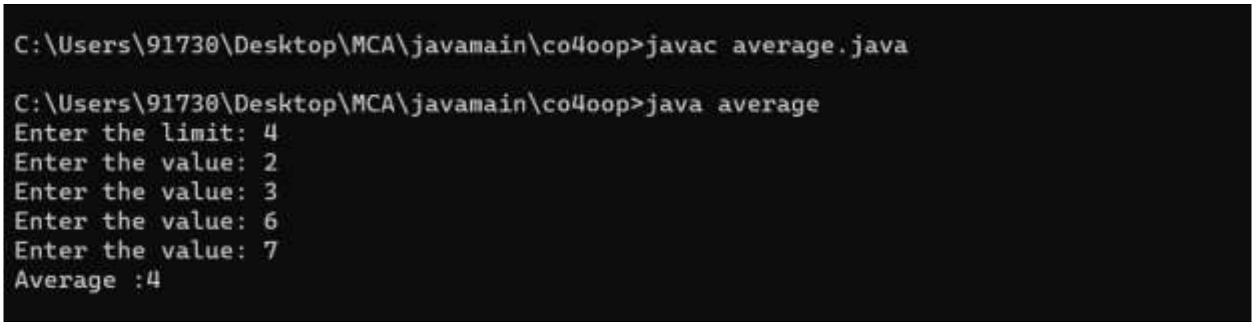
CO4 :

Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

Procedure :

```
import java.util.Scanner;

class nIntExcep extends Exception{

   public nIntExcep(String str){

          super(str);

   }}public class average{

   public static void main(String[] args){

      Scanner Snr=new Scanner(System.in);

      int arr[];

      int sz, total=0, avg, count=0;

      System.out.print("Enter the limit: ");

      sz = Snr.nextInt();

      arr = new int[sz];

      for(int i=0;i<sz;i++)

      {

         System.out.print("Enter the value: ");

         int val = Snr.nextInt();

         arr[i] = val;

      }

      try {

         for(int i=0;i<sz;i++) {
            if(arr[i]<0){
               throw new nIntExcep("Numbers must be positive");
            }
            else{
               total += arr[i];
               count++;
            } }
         avg=total/count;
         System.out.println("Average :"+avg);
      }
      catch(nIntExcep e){
         System.out.println(":: An Exception Occurred :: "+ e);
     }  }}
```

Output Screenshot :



```
C:\Users\91730\Desktop\MCA\javamain\co4oop>javac average.java

C:\Users\91730\Desktop\MCA\javamain\co4oop>java average
Enter the limit: 4
Enter the value: 2
Enter the value: 3
Enter the value: 6
Enter the value: 7
Average :4
```

Result :

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

Experiment No.: 20

Aim :

Define 2 classes; one for generating multiplication table of 5 and other for displaying first N prime numbers. Implement using threads. (Thread class).

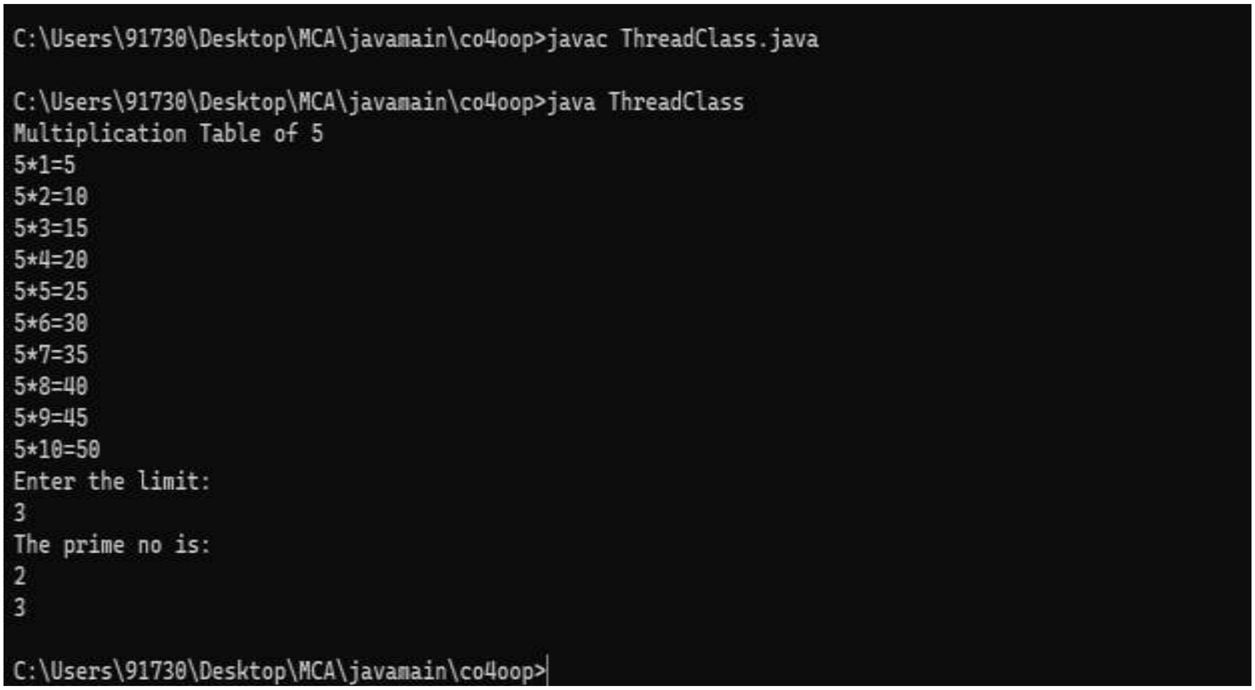
CO4:

Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

Procedure :

```
import java.util.*;
class MulTable extends Thread
{
        public void run()
        {
                int num=5;
                System.out.println("Multiplication Table of 5");
                for(int i=1;i<=10;++i)
                {
                        System.out.println(num+"*"+i+"="+num*i);
                }
        }
}
class PrimeNo extends Thread
{
        public void run()
        {
                int i,j,flag;
                Scanner s=new Scanner(System.in);
                System.out.println("Enter the limit:");
                int N=s.nextInt();
                System.out.println("The prime no is:");
                for(i=1;i<=N;i++)
                {
                        if(i==1||i==0)
                        continue;
                        flag=1;
                        for(j=2;j<=i/2;++j)
                        {
                                if(i%j==0)
                                {
                                        flag=0;
                                        break;
                                }
                        }
                        if(flag==1)
                        System.out.println(i+"");
                }
        }
}
public class ThreadClass
{
        public static void main(String args[])throws InterruptedException
        {
                MulTable m=new MulTable();
                m.start();
                m.sleep(3000);
                PrimeNo p=new PrimeNo();
                p.start();
                p.sleep(1000);
        }
}
```

Output Screenshot :



```
C:\Users\91730\Desktop\MCA\javamain\co4oop>javac ThreadClass.java

C:\Users\91730\Desktop\MCA\javamain\co4oop>java ThreadClass
Multiplication Table of 5
5*1=5
5*2=10
5*3=15
5*4=20
5*5=25
5*6=30
5*7=35
5*8=40
5*9=45
5*10=50
Enter the limit:
3
The prime no is:
2
3

C:\Users\91730\Desktop\MCA\javamain\co4oop>
```

Result :

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

Experiment No.: 21

Aim :

Define 2 classes; one for generating Fibonacci numbers and other for displaying even numbers in a given range. Implement using threads. (Runnable Interface).

CO4:

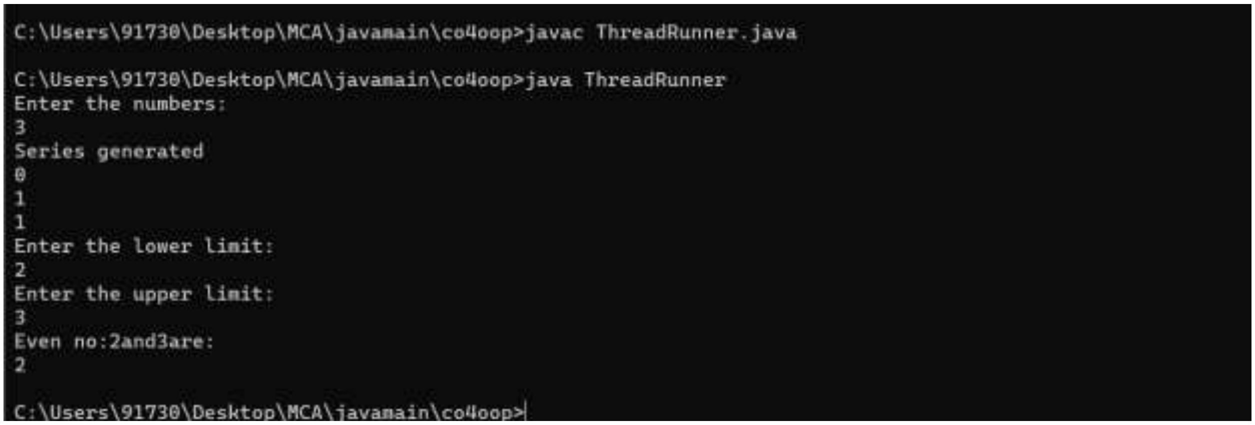
Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

Procedure :

```
import java.util.*;
class Fibonacci implements Runnable
{
        public void run()
        {
        int first=0,second=1,next;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the numbers:");
        int n=sc.nextInt();
        System.out.println("Series generated");
        for(int i=1;i<=n;++i)
                {
                System.out.println(first+"");
                next=first+second;
                first=second;
                second=next;
                }
        }
}
```

```
class EvenNo implements Runnable
{
        public void run()
        {
        Scanner sc=new Scanner(System.in);
        int lower,upper;
        System.out.println("Enter the lower limit:");
        lower=sc.nextInt();
        System.out.println("Enter the upper limit:");
        upper=sc.nextInt();
        System.out.println("Even no:"+lower + "and" + upper+ "are:");
        for(int i=lower;i<=upper;i++)
        if(i%2!=0)
        continue;
        else
        {
                System.out.println(i+"");
        }
        }
}
public class ThreadRunner
{
        public static void main(String args[])throws InterruptedException
        {
        Fibonacci obj1=new Fibonacci();
        Thread a=new Thread(obj1);
        a.start();
        a.sleep(5000);
        EvenNo obj2=new EvenNo();
        Thread b=new Thread(obj2);
        b.start();
        b.sleep(1000);
        }
}
```

Output Screenshot :



```
C:\Users\91730\Desktop\MCA\javamain\co4oop>javac ThreadRunner.java

C:\Users\91730\Desktop\MCA\javamain\co4oop>java ThreadRunner
Enter the numbers:
3
Series generated
0
1
1
Enter the lower limit:
2
Enter the upper limit:
3
Even no:2and3are:
2

C:\Users\91730\Desktop\MCA\javamain\co4oop>
```

Result :

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

Experiment No.: 22

Aim :

Program to create a generic stack and do the Push and Pop operations.

CO4 :

Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

Procedure :

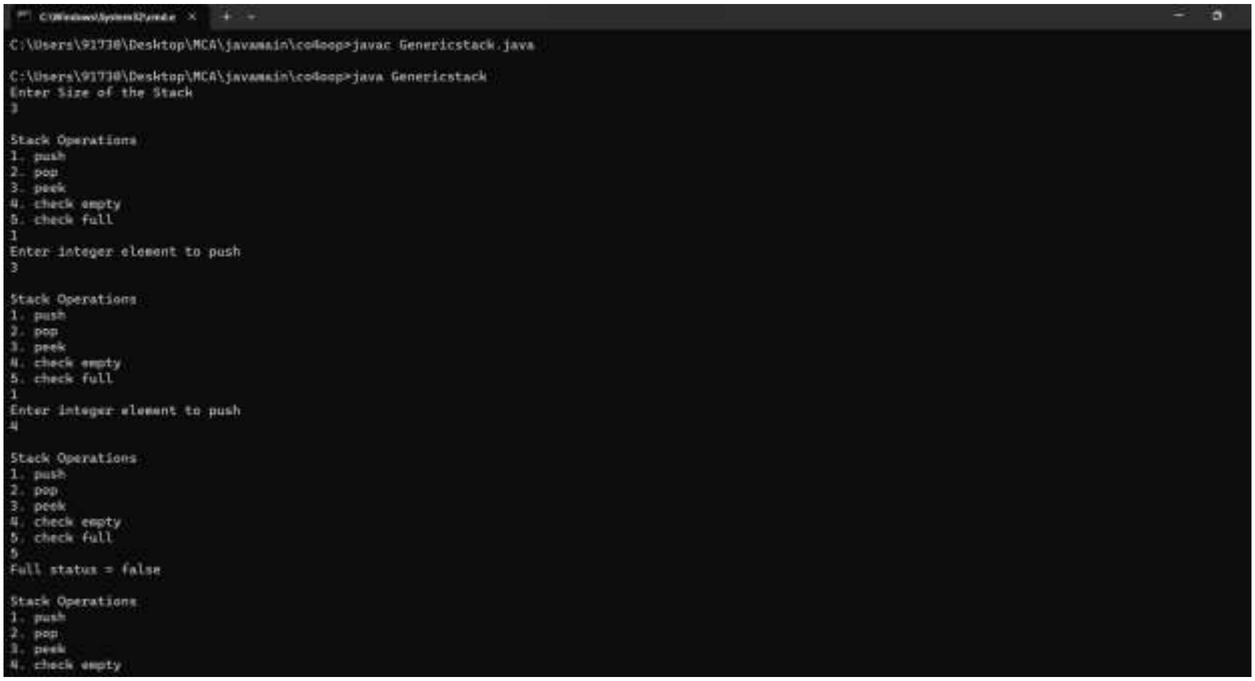
```
import java.util.*;


class arrayStack
{
    public int arr[];
    public int top, size, len;


    public arrayStack(int n)
    {
        size = n;
        len = 0;
        arr = new int[size];
        top = -1;
    }
    public boolean isEmpty()
    {
        return top == -1;
    }
    public boolean isFull()
```

```
   {
      return top == size -1 ;
   }
   public int peek()
   {
      return arr[top];
   }
   public void push(int k)
   {
      if(top + 1 >= size)
                              System.out.println(" overflow ");
      if(top + 1 < size )
         arr[++top] = k;
   }
   public int pop()
   {
      if( isEmpty() )
                              System.out.println(" underflow ");
      return arr[top--];
   }
   public void display()
   {
      System.out.print("\nStack = ");
      for (int i = top; i >= 0; i--)
         System.out.print(arr[i]+" ");
      System.out.println();
   }
}
public class Genericstack
```

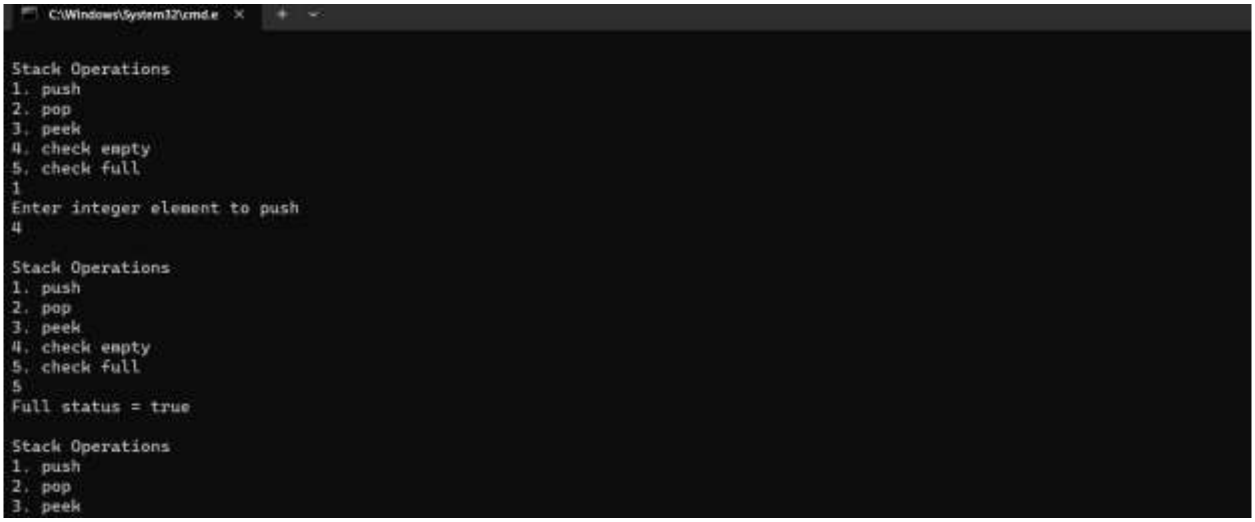
```
{
   public static void main(String[] args)
   {
      Scanner obj = new Scanner(System.in);
      System.out.println("Enter Size of the Stack ");
      int n = obj.nextInt();

      arrayStack stk = new arrayStack(n);

      int ch = 7;
      do{
         System.out.println("\nStack Operations");
         System.out.println("1. push");
         System.out.println("2. pop");
         System.out.println("3. peek");
         System.out.println("4. check empty");
         System.out.println("5. check full");
         int choice = obj.nextInt();
         switch (choice)
         {
         case 1 :
            System.out.println("Enter integer element to push");
                                    stk.push( obj.nextInt() );
            break;
         case 2 :
              System.out.println("Popped Element = " + stk.pop());

            break;
         case 3 :
```

```
                                  System.out.println("Peek Element = " + stk.peek());
                break;
            case 4 :
                System.out.println("Empty status = " + stk.isEmpty());
                break;
            case 5 :
                System.out.println("Full status = " + stk.isFull());
                break;
            default :
                System.out.println("enter valid option \n ");
                break;
            }
        } while (ch != 0);
    }
}
```

Output Screenshot :



```
C:\Users\91730\Desktop\MCA\javamain\co4oop>javac Genericstack.java

C:\Users\91730\Desktop\MCA\javamain\co4oop>java Genericstack
Enter Size of the Stack
3

Stack Operations
1. push
2. pop
3. peek
4. check empty
5. check full
1
Enter integer element to push
3

Stack Operations
1. push
2. pop
3. peek
4. check empty
5. check full
1
Enter integer element to push
4

Stack Operations
1. push
2. pop
3. peek
4. check empty
5. check full
5
Full status = false

Stack Operations
1. push
2. pop
3. peek
4. check empty
```



```
Stack Operations
1. push
2. pop
3. peek
4. check empty
5. check full
1
Enter integer element to push
4

Stack Operations
1. push
2. pop
3. peek
4. check empty
5. check full
5
Full status = true

Stack Operations
1. push
2. pop
3. peek
```

Result :

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

Experiment No.: 23

Aim :

Using generic method perform Bubble sort.

CO4:

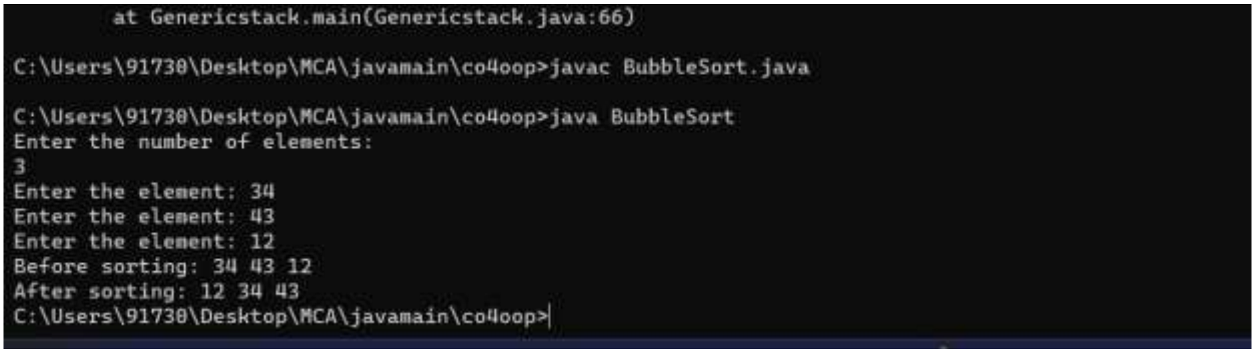
Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

Procedure :

```
import java.util.*;
public class BubbleSort{
    int sz;
    int[] Arr;
    public BubbleSort(int n){
        sz = n;
        Arr = new int[sz];
    }
    public void insert(int i, int f){
        Arr[i] = f;
    }
    public void display(int i){
        System.out.print(Arr[i]+ " ");
    }
    public void Sort(int n){
        int temp;
        for(int i=0; i<n; i++){
            for(int j=i+1; j<n; j++){
                if(Arr[i] > Arr[j]){
```

```
temp = Arr[i];
 Arr[i] = Arr[j];
 Arr[j] = temp;
              }
          }
      }
   }
   public static void main(String[] args){
       Scanner Snr= new Scanner(System.in);
       System.out.println("Enter the number of elements: ");
       int size = Snr.nextInt();
       BubbleSort arr = new BubbleSort(size);
       for(int i=0; i<size; i++){
           System.out.print("Enter the element: ");
           int val = Snr.nextInt();
           arr.insert(i, val);
       }
       System.out.print("Before sorting: ");
       for(int i=0; i<size; i++){
           arr.display(i);
       }
       System.out.print("\nAfter sorting: ");
       for(int i=0; i<size; i++){
           arr.Sort(size);
           arr.display(i);
       }}}
```

Output Screenshot :



```
        at Genericstack.main(Genericstack.java:66)

C:\Users\91730\Desktop\MCA\javamain\co4oop>javac BubbleSort.java

C:\Users\91730\Desktop\MCA\javamain\co4oop>java BubbleSort
Enter the number of elements:
3
Enter the element: 34
Enter the element: 43
Enter the element: 12
Before sorting: 34 43 12
After sorting: 12 34 43
C:\Users\91730\Desktop\MCA\javamain\co4oop>
```

Result :

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

Experiment No.: 24

Aim :

Maintain a list of Strings using ArrayList from collection framework, perform built-in operations.

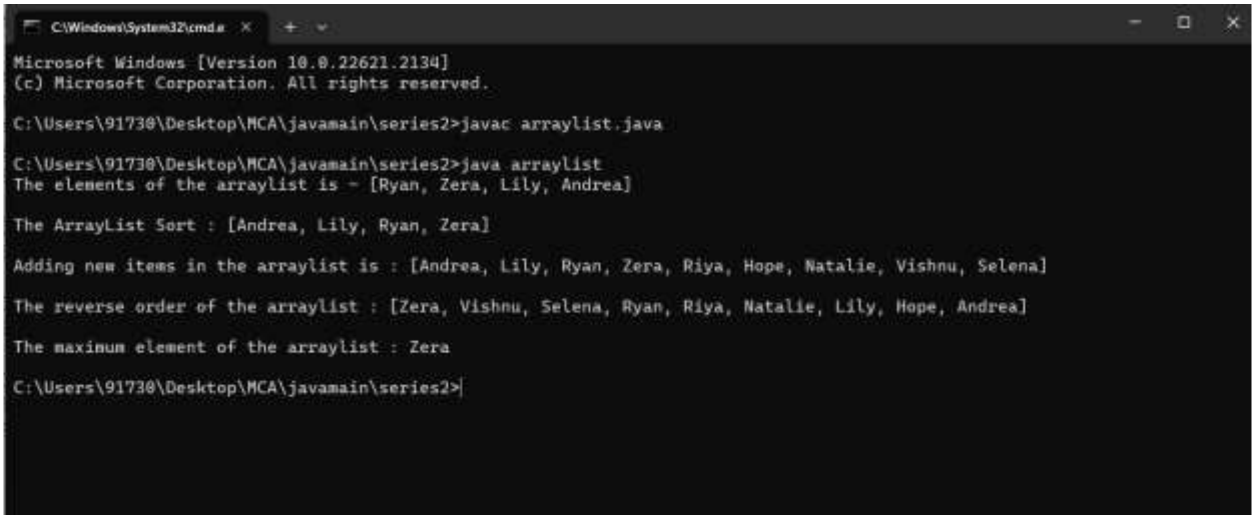
CO4 :

Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

Procedure :

```
import java.util.*;


public class arraylist{
    public static void main(String[] args) {
        ArrayList<String> arrayList= new ArrayList<>();


        arrayList.add("Ryan");
        arrayList.add("Zera");
        arrayList.add("Lily");
        arrayList.add("Andrea");
        System.out.println("The elements of the arraylist is - "+arrayList);


        Collections.sort(arrayList);
        System.out.println("\nThe ArrayList Sort : "+arrayList);


        Collections.addAll(arrayList,"Riya","Hope","Natalie","Vishnu","Selena");
        System.out.println("\nAdding new items in the arraylist is : "+arrayList);
Collections.sort(arrayList, Collections.reverseOrder());
        System.out.println("\nThe reverse order of the arraylist : "+arrayList);
        System.out.println("\nThe maximum element of the arraylist :
"+Collections.max(arrayList));
    }
}
```

Output Screenshot :



```
Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

C:\Users\91730\Desktop\MCA\javamain\series2>javac arraylist.java

C:\Users\91730\Desktop\MCA\javamain\series2>java arraylist
The elements of the arraylist is - [Ryan, Zera, Lily, Andrea]

The ArrayList Sort : [Andrea, Lily, Ryan, Zera]

Adding new items in the arraylist is : [Andrea, Lily, Ryan, Zera, Riya, Hope, Natalie, Vishnu, Selena]

The reverse order of the arraylist : [Zera, Vishnu, Selena, Ryan, Riya, Natalie, Lily, Hope, Andrea]

The maximum element of the arraylist : Zera

C:\Users\91730\Desktop\MCA\javamain\series2>
```

Result :

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

Experiment No.: 25

Aim :

Program to remove all the elements from a linked list.

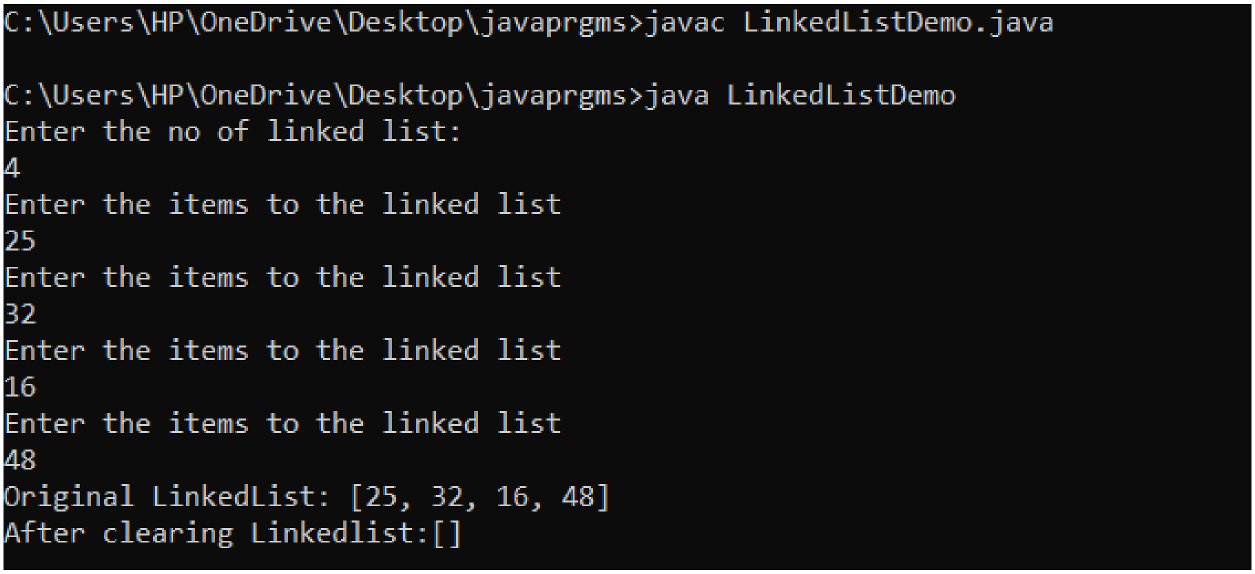
CO4 :

Implement packages, exception handling, multithreading and generic programmjavaing by using the java.util package and Collection framework.

Procedure :

```
import java.util.LinkedList;
import java.util.*;
public class LinkedListDemo
{
        public static void main(String[] args)
        {
                LinkedList<String> list = new LinkedList<String>();
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter the no of linked list: ");
                int n=sc.nextInt();
                for(int i=0;i<n;i++)
                {
                        System.out.println("Enter the items to the linked list");
                        String item=sc.next();
                        list.add(item);
                }
                System.out.println("Original LinkedList: "+list);
                list.clear();
                System.out.println("After clearing Linkedlist:" +list);
        }
}
```

Output Screenshot :



```
C:\Users\HP\OneDrive\Desktop\javaprgms>javac LinkedListDemo.java

C:\Users\HP\OneDrive\Desktop\javaprgms>java LinkedListDemo
Enter the no of linked list:
4
Enter the items to the linked list
25
Enter the items to the linked list
32
Enter the items to the linked list
16
Enter the items to the linked list
48
Original LinkedList: [25, 32, 16, 48]
After clearing Linkedlist:[]
```

Result :

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

Experiment No.: 26

Aim :

Program to remove an object from the Stack when the position is passed as parameter.

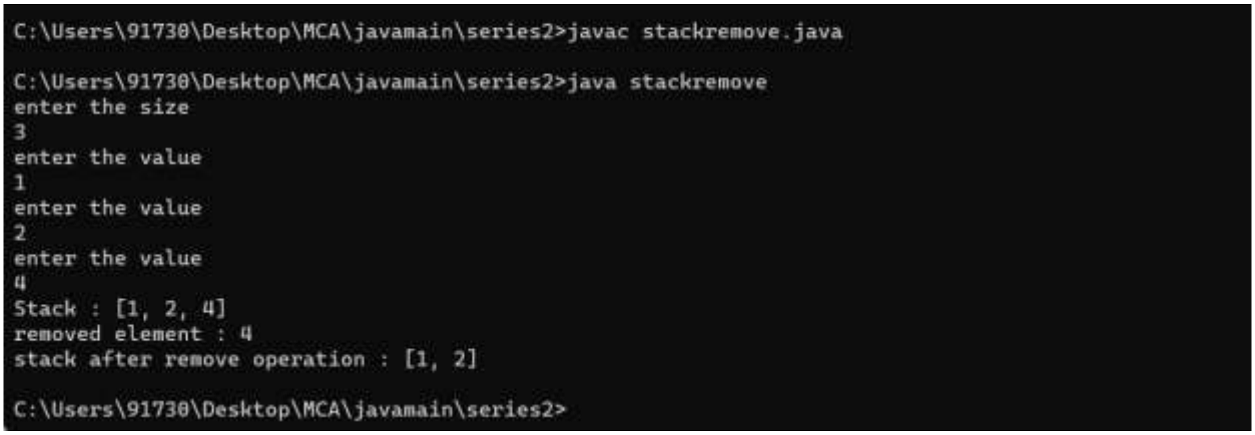
CO4 :

Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

Procedure :

```
import java.util.*;


public class stackremove{
        public static void main(String args[])
        {
                Stack<String> StackDemo=new Stack<String>();


                Scanner obj=new Scanner(System.in);
                String n;
                System.out.println("enter the size");
                int sz=obj.nextInt();
                for(int i=0;i<sz;i++)
                {
                        System.out.println("enter the value");
                        n=obj.next();
                        StackDemo.add(n);
                }
                System.out.println("Stack : "+ StackDemo);
                String rem_ele = StackDemo.remove(2);
                System.out.println("removed element : "+rem_ele);
                System.out.println("stack after remove operation : "+StackDemo);
                }}
```

Output Screenshot :



```
C:\Users\91730\Desktop\MCA\javamain\series2>javac stackremove.java

C:\Users\91730\Desktop\MCA\javamain\series2>java stackremove
enter the size
3
enter the value
1
enter the value
2
enter the value
4
Stack : [1, 2, 4]
removed element : 4
stack after remove operation : [1, 2]

C:\Users\91730\Desktop\MCA\javamain\series2>
```

Result :

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

Experiment No.: 27

Aim :

Program to demonstrate the creation of queue object using the PriorityQueue class.

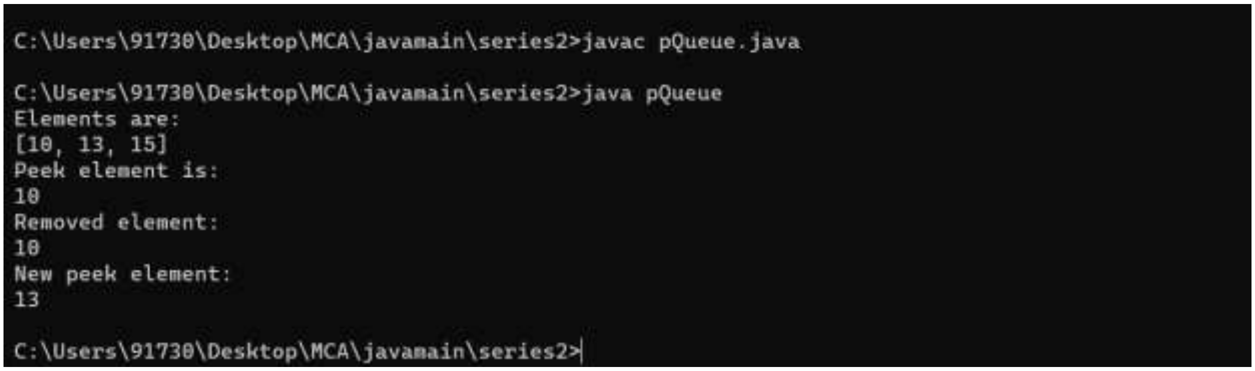
CO4 :

Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

Procedure :

```
import java.util.*;
public class pQueue {
public static void main(String[] args) {
PriorityQueue <Integer> pq = new PriorityQueue<>();
pq.add(10);
pq.add(13);
pq.add(15);
System.out.println("Elements are:");
System.out.println(pq);
System.out.println("Peek element is:");
System.out.println(pq.peek());
System.out.println("Removed element:");
System.out.println(pq.poll());
System.out.println("New peek element:");
System.out.println(pq.peek());
}
}
```

Output Screenshot :



```
C:\Users\91730\Desktop\MCA\javamain\series2>javac pQueue.java

C:\Users\91730\Desktop\MCA\javamain\series2>java pQueue
Elements are:
[10, 13, 15]
Peek element is:
10
Removed element:
10
New peek element:
13

C:\Users\91730\Desktop\MCA\javamain\series2>
```

Result :

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

Experiment No.: 28

Aim :

Program to demonstrate the addition and deletion of elements in deque.

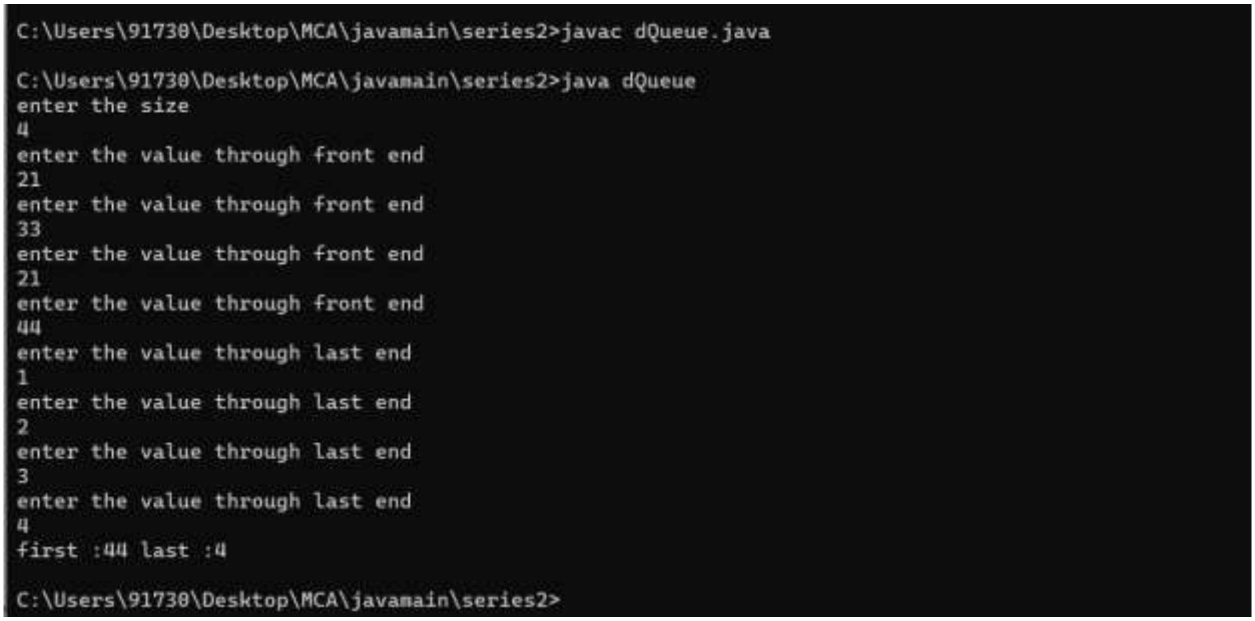
CO4 :

Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

Procedure :

```
import java.util.*;
public class dQueue{
        public static void main(String args[])
        {
                Deque<Integer> dq = new ArrayDeque<>();

                Scanner obj=new Scanner(System.in);
                int n;
                System.out.println("enter the size");
                int sz=obj.nextInt();
                for(int i=0;i<sz;i++)
                {
                        System.out.println("enter the value through front end");
                        n=obj.nextInt();
                        dq.addFirst(n);
                }
                for(int i=0;i<sz;i++)
                {
                    System.out.println("enter the value through last end");
                    n=obj.nextInt();
                    dq.addLast(n);
                }

                int first = dq.removeFirst();
                int last = dq.removeLast();
                System.out.println("first :"+ first +" last :"+ last);
        }
}
```

Output Screenshot :



```
C:\Users\91730\Desktop\MCA\javamain\series2>javac dQueue.java

C:\Users\91730\Desktop\MCA\javamain\series2>java dQueue
enter the size
4
enter the value through front end
21
enter the value through front end
33
enter the value through front end
21
enter the value through front end
44
enter the value through last end
1
enter the value through last end
2
enter the value through last end
3
enter the value through last end
4
first :44 last :4

C:\Users\91730\Desktop\MCA\javamain\series2>
```

Result :

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

Experiment No.: 29

Aim :

Program to demonstrate the creation of Set object using the LinkedHashset class.

CO4 :

Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

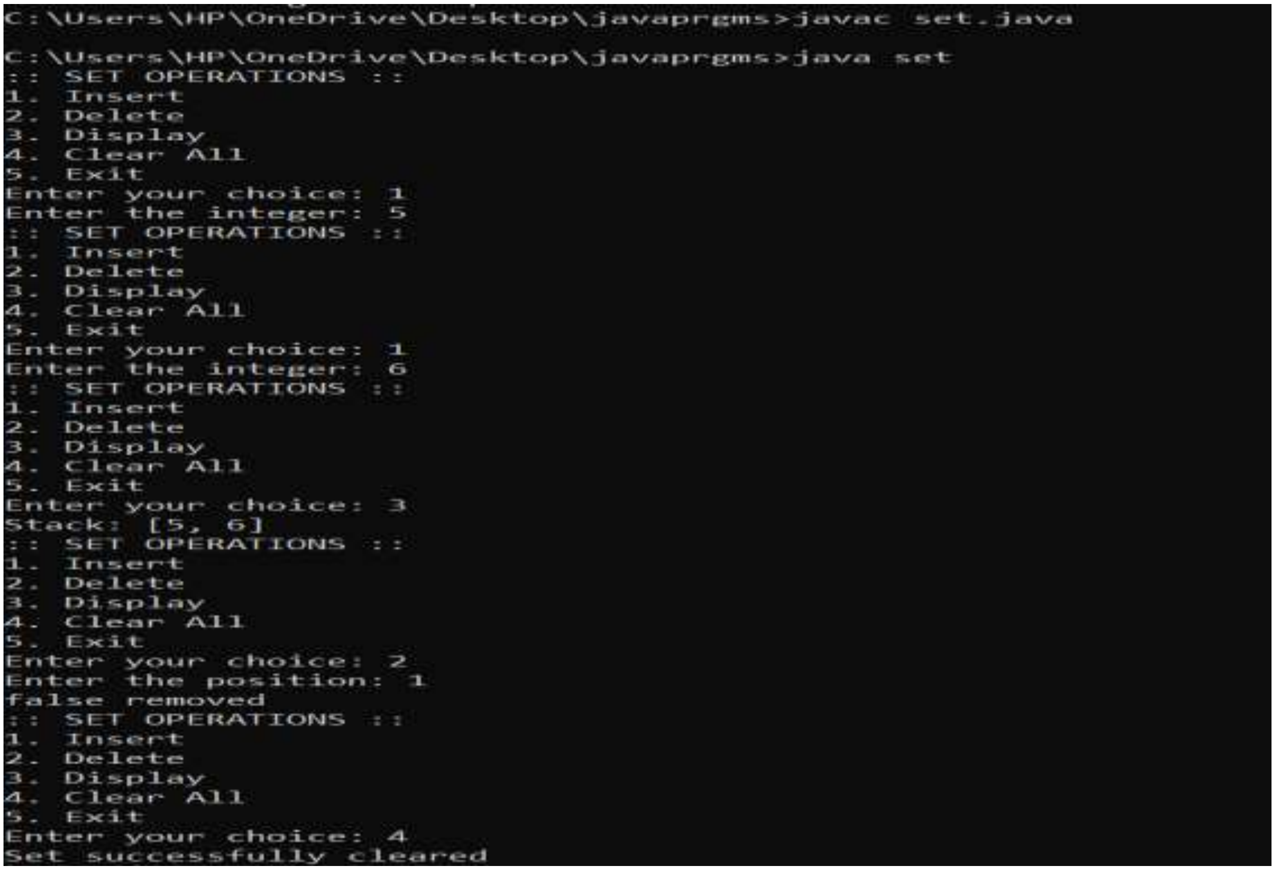
Procedure :

```
import java.util.*;

import java.util.LinkedHashSet;

import java.util.Set;

public class set

{

        public static void insert(Set<Integer> st)

        {

                Scanner Snr = new Scanner(System.in);

                System.out.print("Enter the integer: ");

                int line = Snr.nextInt();

                st.add(line);

        }

        public static void del(Set<Integer> st)

        {

                Scanner Snr = new Scanner(System.in);

                System.out.print("Enter the position: ");

                int pos = Snr.nextInt();

                pos = pos-1;

                System.out.println(st.remove(pos) + " removed");;

        }
```

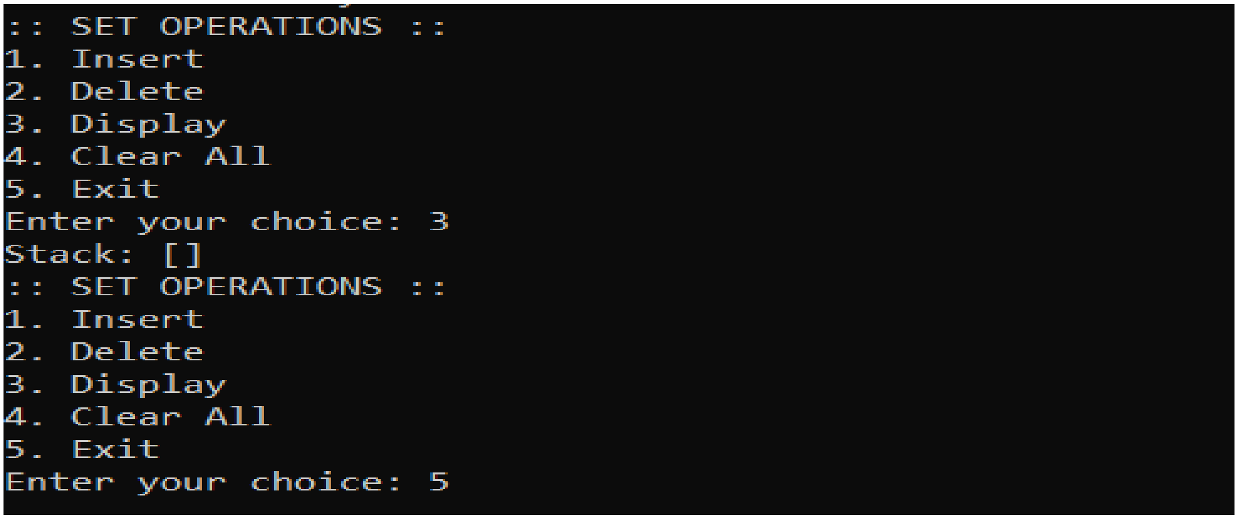
```
    public static void display(Set<Integer> st)
    {
        System.out.println("Stack: " + st);
    }
    public static void delAll(Set<Integer> st)
    {
        st.clear();
        System.out.println("Set successfully cleared");
    }
    public static void main(String[] args)
    {
        Set<Integer> st = new LinkedHashSet<Integer>();
        Scanner Snr = new Scanner(System.in);
        int choice1 = 0, choice2 = 0, choice3 = 0;
        while(choice1 != 5)
        {
            System.out.println(":: SET OPERATIONS ::");
            System.out.println("1. Insert");
            System.out.println("2. Delete");
            System.out.println("3. Display");
            System.out.println("4. Clear All");
            System.out.println("5. Exit");
            System.out.print("Enter your choice: ");
            choice1 = Snr.nextInt();
            switch(choice1){
                case 1:
                    insert(st);
                    break;
                case 2:
```

```
                del(st);
                break;
        case 3:
                display(st);
                break;
        case 4:
                delAll(st);
                break;
        case 5:
                System.exit(0);
                break;
        default:
                System.out.println("Enter a valid choice")     }}}}
```

Output Screenshot :



```
C:\Users\HP\OneDrive\Desktop\javaprgms>javac set.java

C:\Users\HP\OneDrive\Desktop\javaprgms>java set
:: SET OPERATIONS ::
1. Insert
2. Delete
3. Display
4. Clear All
5. Exit
Enter your choice: 1
Enter the integer: 5
:: SET OPERATIONS ::
1. Insert
2. Delete
3. Display
4. Clear All
5. Exit
Enter your choice: 1
Enter the integer: 6
:: SET OPERATIONS ::
1. Insert
2. Delete
3. Display
4. Clear All
5. Exit
Enter your choice: 3
Stack: [5, 6]
:: SET OPERATIONS ::
1. Insert
2. Delete
3. Display
4. Clear All
5. Exit
Enter your choice: 2
Enter the position: 1
false removed
:: SET OPERATIONS ::
1. Insert
2. Delete
3. Display
4. Clear All
5. Exit
Enter your choice: 4
Set successfully cleared
```



```
:: SET OPERATIONS ::
1. Insert
2. Delete
3. Display
4. Clear All
5. Exit
Enter your choice: 3
Stack: []
:: SET OPERATIONS ::
1. Insert
2. Delete
3. Display
4. Clear All
5. Exit
Enter your choice: 5
```

Result :

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

}

Experiment No.: 30

Aim :

Write a Java program to compare two hash set.

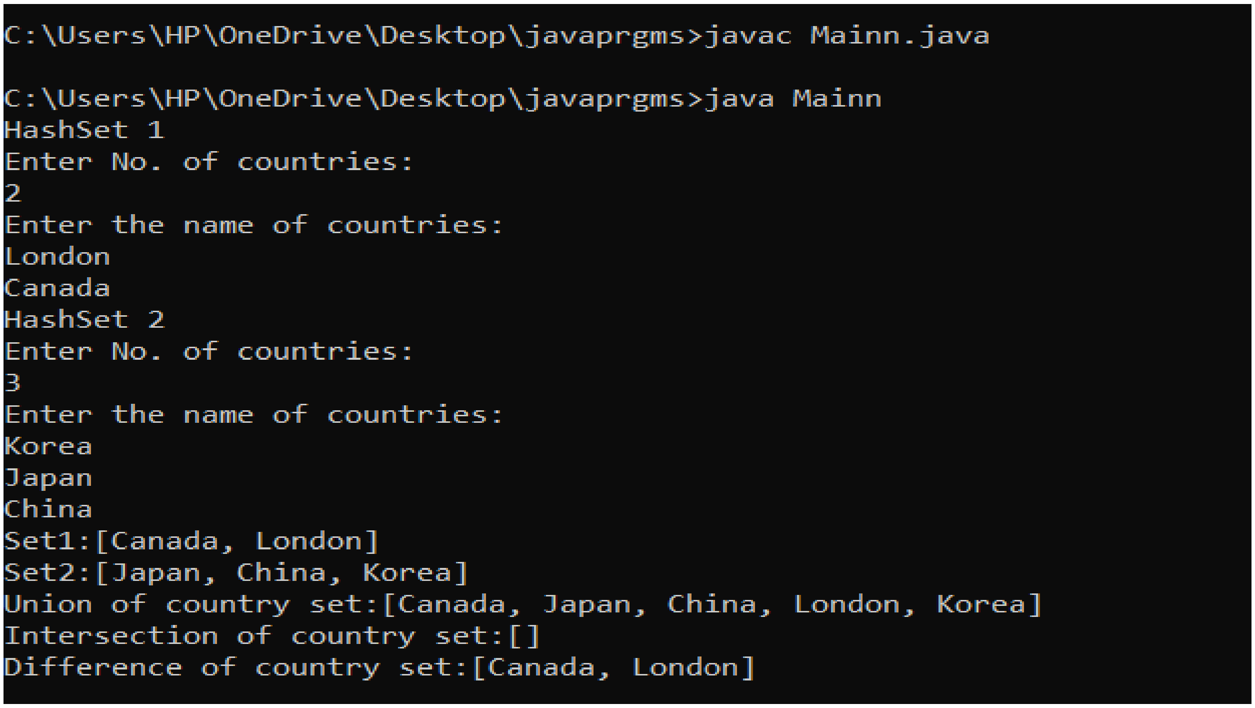
CO4 :

Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

Procedure :

```
import java.util.*;
public class Mainn
{
public static void main(String[] args)
{
int n;
String str;
HashSet<String> set1= new HashSet<String>();
System.out.println("HashSet 1");
System.out.println("Enter No. of countries:");
Scanner sc=new Scanner(System.in);
n=sc.nextInt();
System.out.println("Enter the name of countries:");
Scanner sc1=new Scanner(System.in);
for(int i=0;i<n;i++)
{
str=sc1.nextLine();
set1.add(str);
}
System.out.println("HashSet 2");
HashSet<String> set2= new HashSet<String>();
System.out.println("Enter No. of countries:");
n=sc.nextInt();

System.out.println("Enter the name of countries:");
for(int i=0;i<n;i++)
{
str=sc1.nextLine();
set2.add(str);
}
System.out.println("Set1:"+set1);
System.out.println("Set2:"+set2);

HashSet<String> a= new HashSet<String>(set1);
a.addAll(set2);

System.out.println("Union of country set:"+a);
HashSet<String> b= new HashSet<String>(set1);
b.retainAll(set2);

System.out.println("Intersection of country set:"+b);
HashSet<String> c= new HashSet<String>(set1);
c.removeAll(set2);

System.out.println("Difference of country set:"+c);
}

}
```

Output Screenshot :



```
C:\Users\HP\OneDrive\Desktop\javaprgms>javac Mainn.java

C:\Users\HP\OneDrive\Desktop\javaprgms>java Mainn
HashSet 1
Enter No. of countries:
2
Enter the name of countries:
London
Canada
HashSet 2
Enter No. of countries:
3
Enter the name of countries:
Korea
Japan
China
Set1:[Canada, London]
Set2:[Japan, China, Korea]
Union of country set:[Canada, Japan, China, London, Korea]
Intersection of country set:[]
Difference of country set:[Canada, London]
```

Result :

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

Experiment No.: 31

Aim :

Program to demonstrate the working of Map interface by adding, changing and removing elements.

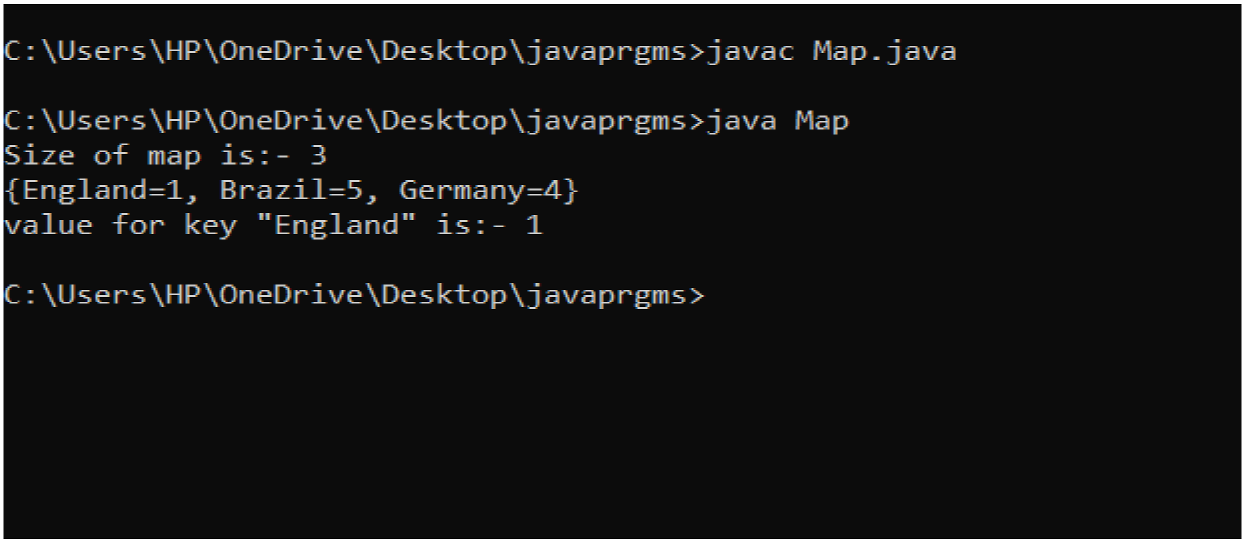
CO4 :

Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

Procedure :

```
import java.util.HashMap;
public class Map
{
public static void main(String[] args)
{
HashMap<String, Integer> map = new HashMap<>();
map.put("Germany", 4);
map.put("England", 1);
map.put("Brazil", 5);
System.out.println("Size of map is:- "+ map.size());
System.out.println(map);
if (map.containsKey("England"))
{
// Mapping
Integer a = map.get("England");
System.out.println("value for key"+ " \"England\" is:- " + a);
}}}
```

Output Screenshot :



```
C:\Users\HP\OneDrive\Desktop\javaprgms>javac Map.java

C:\Users\HP\OneDrive\Desktop\javaprgms>java Map
Size of map is:- 3
{England=1, Brazil=5, Germany=4}
value for key "England" is:- 1

C:\Users\HP\OneDrive\Desktop\javaprgms>
```

Result :

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

Experiment No.: 32

Aim :

Program to Convert HashMap to TreeMap .

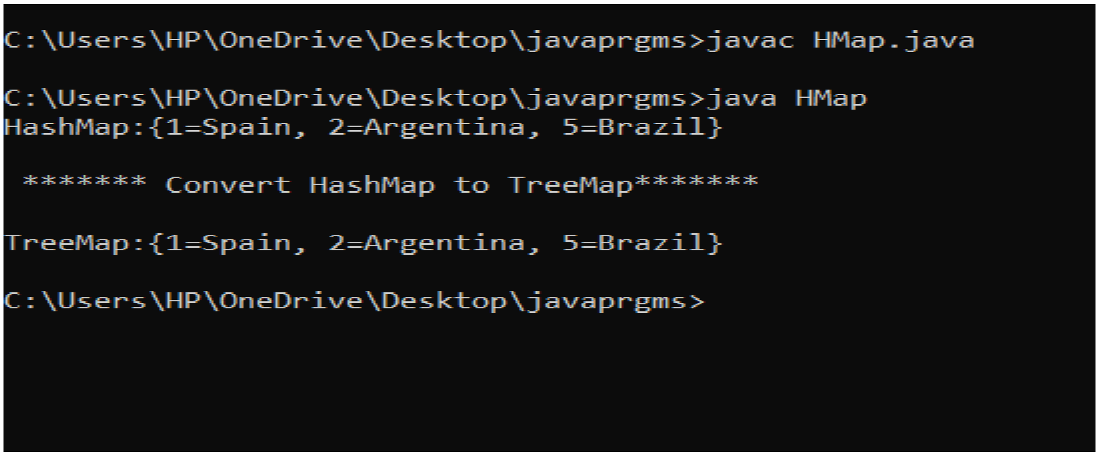
CO4 :

Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

Procedure :

```
import java.util.*;

import java.util.Map;

public class HMap

{

public static void main(String[] args)

{

Map<Integer,String> hm=new LinkedHashMap<>();

hm.put(1,"England");

hm.put(1,"Spain");

hm.put(2,"France");

hm.put(5,"Brazil");

hm.put(2,"Argentina");

System.out.println("HashMap:"+hm);

Map<Integer,String> tm=new TreeMap<>(hm);

System.out.println("\n Convert HashMap to TreeMap\n");

System.out.println("TreeMap:"+tm);

}

}
```

Output Screenshot :



```
C:\Users\HP\OneDrive\Desktop\javaprgms>javac HMap.java

C:\Users\HP\OneDrive\Desktop\javaprgms>java HMap
HashMap:{1=Spain, 2=Argentina, 5=Brazil}

 ******* Convert HashMap to TreeMap*******

TreeMap:{1=Spain, 2=Argentina, 5=Brazil}

C:\Users\HP\OneDrive\Desktop\javaprgms>
```

Result :

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

Experiment No.: 33

Aim :

Program to draw Circle, Rectangle, Line in Applet.

CO5 :

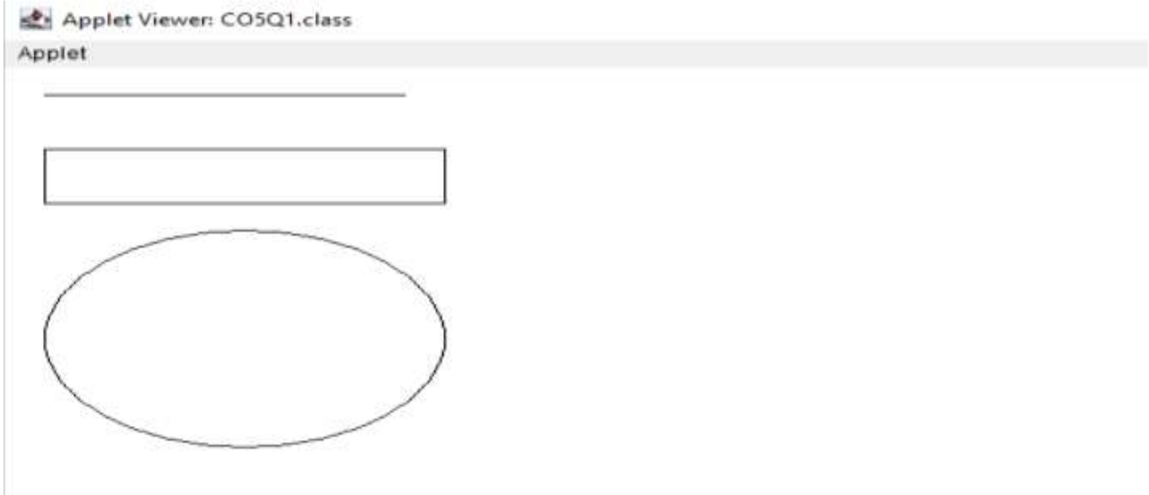
Design applications using files and network concepts.

Procedure :

```
//Program to draw Circle, Rectangle, Line in Applet.
import java.applet.*;
import java.awt.*;



public class CO5Q1 extends Applet{
    public void paint(Graphics g) {
        g.drawLine(20, 20, 200, 20);
        g.drawRect(20, 60, 200, 40);
        g.drawOval(20, 120, 200, 160);
    }

}
<HTML>
<HEAD>
</HEAD>
<BODY>
<div align="center">
<APPLET CODE="CO5Q1.class" WIDTH="800" HEIGHT="500"></APPLET>
</div>
</BODY>
```

```
</HTML>
```

Output Screenshot :



Result :

The program was executed and the result was successfully obtained. Thus CO5 was obtained.

Experiment No.: 34

Aim :

Program to find maximum of three numbers using AWT.

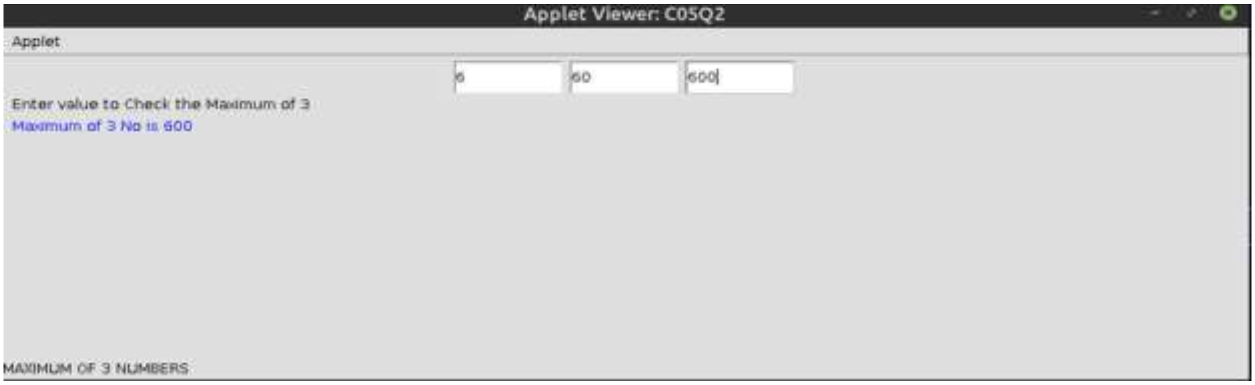
CO4 :

Design applications using files and network concepts.

Procedure :

```
import java.awt.*;
import java.applet.*;
public class MaxOfThree extends Applet
{
     TextField T1,T2,T3;
       publicvoid init(){
          T1 = new TextField(10);
          T2 = new TextField(10);
          T3 = new TextField(10);
 add(T1);
          add(T2);
          add(T3);
T1.setText("0");
          T2.setText("0");
          T3.setText("0");
        }

        publicvoid paint(Graphics g){
             int a, b, c,result;
             String str;
             g.drawString("Enter value to Check the Maximum of 3 ",10,50);


             str=T1.getText();
             a=Integer.parseInt(str);
             str=T2.getText();
             b=Integer.parseInt(str);
             str=T3.getText();
             c=Integer.parseInt(str);


              g.setColor(Color.blue);
              if (a>b) {
                  if (a>c)
                      result=a;
                  else
                      result=c;
                }
            else{
                  if (b>c)
                      result=b;
                  else
                      result=c;
              }
              g.drawString("Maximnum of 3 No is "+result,10,70);
              showStatus("MAXIMUM OF 3 NUMBERS");
            }


            public boolean action(Event e, Object o){
                  repaint();
                  return true;
            	}
            }
```

Output Screenshot :



```
6        60        600
Enter value to Check the Maximum of 3
Maximum of 3 No is 600
MAXIMUM OF 3 NUMBERS
```

Result :

The program was executed and the result was successfully obtained. Thus CO5 was obtained.

Experiment No.: 35

Aim :

Find the percentage of marks obtained by a student in 5 subjects. Display a happy face if he secures above 50% or a sad face if otherwise.

CO5 :

Design applications using files and network concepts.

Procedure :

```
import java.applet.*;

import java.awt.*;

import java.awt.event.*;


public class CO5Q3 extends Applet implements ActionListener {

    TextField t1,t2,t3,t4;

    Button b;

    Label l1,l2,l3,l4;

    public void init(){

        l1=new Label("mark1");

//l1.setBounds(100,100,200,20);

        t1= new TextField(5);

//t1.setBounds(100,50,200,20);


        l2=new Label("mark2");

//l2.setBounds(100,130,100,30);

        t2= new TextField(5);

//t2.setBounds(100,80,100,20);


        l3=new Label("mark3");

//l3.setBounds(100,160,100,20);
```

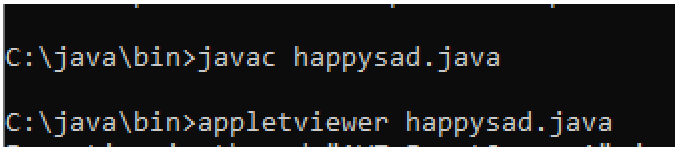
```
        t3= new TextField(5);
//t3.setBounds(100,120,100,20);

        l4=new Label("result");
//l4.setBounds(100,200,100,20);

        t4=new TextField(5);




        t1.setBounds(210,40,100,20);
        t2.setBounds(210,80,100,20);
        t3.setBounds(210,120,100,20);
        t4.setBounds(210,140,100,20);
        l1.setBounds(100,40,100,20);
        l2.setBounds(100,80,100,20);
        l3.setBounds(100,120,100,20);
        l4.setBounds(100,140,100,20);

        b=new Button("find");
        b.setBounds(230,150,60,50);

//t4.setBounds(100,200,100,20);





        add(l1);
        add(l2);
```

```
    add(l3);
    add(l4);
    add(t1);
    add(t2);
    add(t3);
    add(t4);
    add(b);
    b.addActionListener(this);
  }
  public void actionPerformed(ActionEvent e){

    int x=0;
    int y=0;
    int z=0;
    int total=0;
    x= Integer.parseInt(t1.getText());
    y= Integer.parseInt(t2.getText());
    z= Integer.parseInt(t3.getText());
    if(e.getSource()==b){

      total=(x+y+z)/3;
      t4.setText(String.valueOf(total));
    }




  }

  public void paint(Graphics g){
```

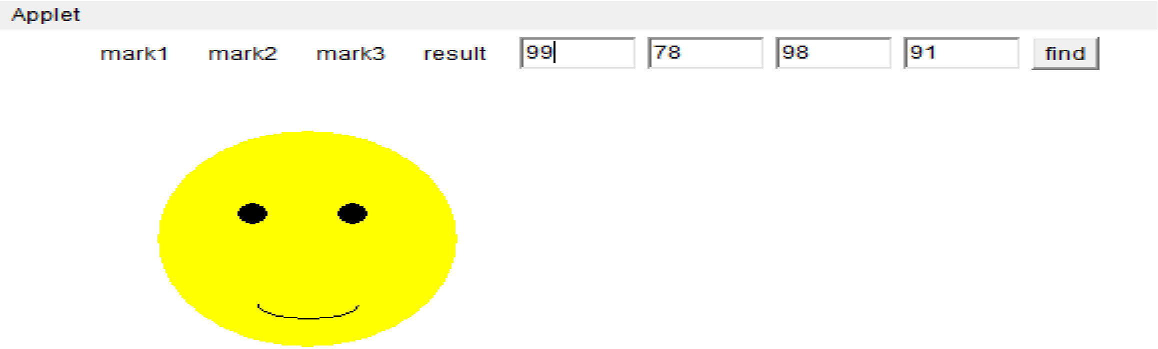
```
int x=0;
int y=0;
int z=0;
int total=0;



x= Integer.parseInt(t1.getText());
y= Integer.parseInt(t2.getText());
z= Integer.parseInt(t3.getText());

total=(x+y+z)/3;

if(total > 50){
   g.setColor(Color.YELLOW);
   g.fillOval(80,70, 150, 150);

   g.setColor(Color.BLACK);
   g.fillOval(120,120,15,15);
   g.fillOval(170,120,15,15);

   g.drawArc(130,180,50,20,180,180);
}
else
{
   g.setColor(Color.YELLOW);
   g.fillOval(80,70, 150, 150);

   g.setColor(Color.BLACK);
   g.fillOval(120,120,15,15);
```

```
        g.fillOval(170,120,15,15);
        g.drawArc(130,180,50,20,180,-180);
    }
}}
```

Output Screenshot :



```
C:\java\bin>javac happysad.java

C:\java\bin>appletviewer happysad.java
```



mark1 mark2 mark3 result 99 78 98 91

Result :

The program was executed and the result was successfully obtained. Thus CO5 was obtained.

Experiment No.: 36

Aim :

Using 2D graphics commands in an Applet, construct a house. On mouse click event, change the color of the door from blue to red.

CO5 :

Design applications using files and network concepts.

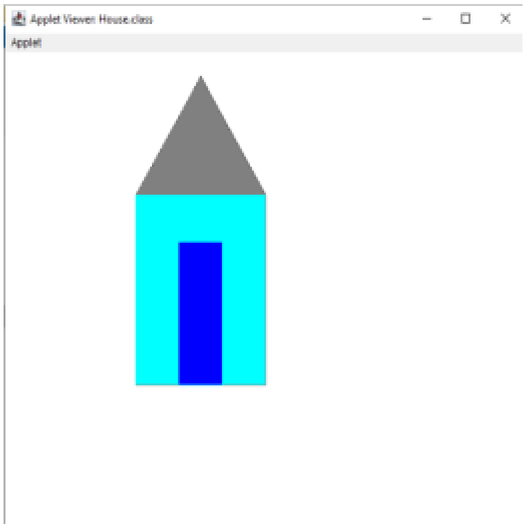
Procedure :

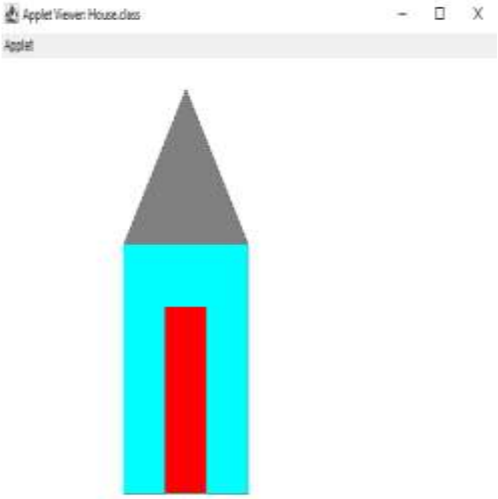
```
import java.awt.*;
import java.applet.*;
import java.awt.event.*;
public class House extends Applet implements MouseListener
{
int a,b;
public void init()
{
addMouseListener( this);
}
public void paint(Graphics g)
{
int x[]={150,300,225};
int y[]={150,150,25};
g.drawPolygon(x,y,3);
g.setColor(Color.GRAY);
g.fillPolygon(x,y,3);
g.drawRect(150,150,150,200);//House
g.setColor(Color.CYAN);
g.fillRect(150,150,150,200);
g.drawRect(200,200,50,150);//Door
g.setColor(Color.blue);
g.fillRect(200,200,50,150);
if(a>200 && a<300 && b>200 && b<300)
{
g.setColor(Color.red);
g.fillRect(200, 200, 50, 150);
}
}
public void mouseClicked(MouseEvent e)
{
}
public void mouseEntered(MouseEvent e)
{
}
@Override
public void mouseExited(MouseEvent e) {
}
public void mousePressed(MouseEvent e)
{
a=e.getX();
b=e.getY();
repaint();
}
public void mouseReleased(MouseEvent e)
{
}
}
```

Index.html

```
<html>

<body>

<applet code="House.class" width="600" height="600">

</applet>

</body>

</html>
```

Output Screenshot :





Result :

The program was executed and the result was successfully obtained. Thus CO5 was obtained.

Experiment No.: 37

Aim :

Develop a program that has a Choice component which contains the names of shapes such as rectangle, triangle, square and circle. Draw the corresponding shapes for given parameters as per user’s choice.

CO5 :

Design applications using files and network concepts.

Procedure :

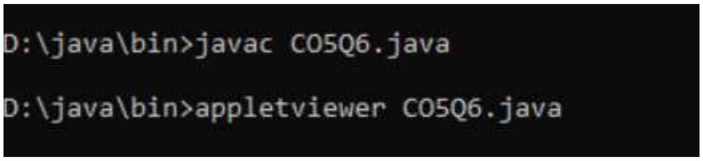
```
import java.awt.*;

import java.awt.event.*;

import java.applet.*;

import java.awt.event.ItemEvent;

import java.awt.event.ItemListener;

import java.awt.Graphics;


public class CO5Q6 extends Applet implements ItemListener

{

        Choice c;

        int n;


                public void init()

                {

                        Label l;

                        l=new Label("Select an option");

                        l.setBounds(80,100,100,50);
```

```
		add(l);
		c=new Choice();
		c.addItem("Choose shape:");
		c.addItem("Rectangle");
		c.addItem("Triangle");
		c.addItem("Square");
		c.addItem("Circle");
		c.addItemListener(this);
		add(c);
	}
	public void paint(Graphics g)
	{
		if(n==0)
		{
		}
		if(n==1)
		{
			g.drawRect(20,60,200,40);
			g.setColor( Color.cyan);
			g.fillRect(20,60,200,40);
			g.setColor(Color.blue);
		}
		if(n==2)
		{
			int[] x = new int[]{50, 50,200};
			int[] y = new int[]{500, 400, 500};
			g.drawPolygon(x,y,3);
			g.setColor(Color.green);
			g.fillPolygon(x,y,3);
```

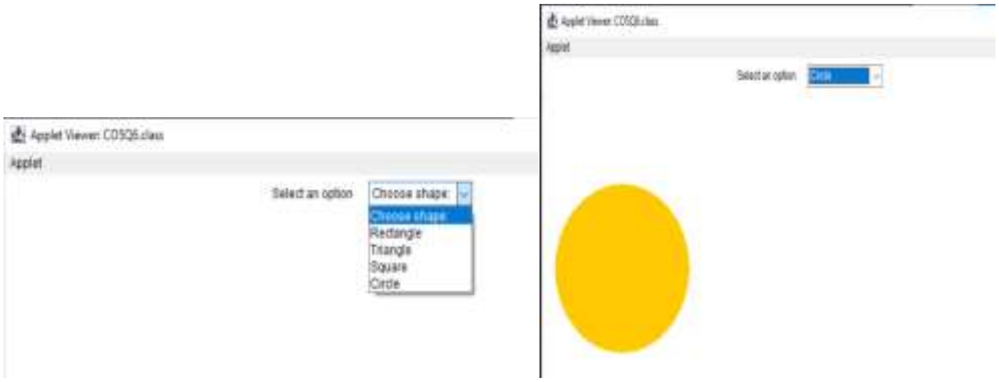
```
                }
                if(n==3)
                {
                        g.drawRect(100, 100, 100, 100);
                        g.setColor(Color.magenta);
                        g.fillRect(100,100,100,100);
                        g.setColor(Color.gray);
                }
                if(n==4)
                {
                        g.setColor(Color.orange);
                        g.drawOval(20, 120, 200, 160);
                        g.fillOval(20,120,200,160);
                        g.setColor(Color.green);
                }
        }

        public void itemStateChanged (ItemEvent e)
        {
                n = c.getSelectedIndex();
                c.repaint();
        }
}
/*
<html>
<head>
<title>
</title>
```

```
</head>
<body>
<div align="center">

<applet code="CO5Q6.class" width="800" height="500">

</applet>
</div>
</body>
</html>
*/
```

Output Screenshot :



```
D:\java\bin>javac CO5Q6.java

D:\java\bin>appletviewer CO5Q6.java
```



Result :

The program was executed and the result was successfully obtained. Thus CO5 was obtained.

Experiment No.: 38

Aim :

Develop a program to handle all mouse events and window events.

CO5 :

Design applications using files and network concepts.

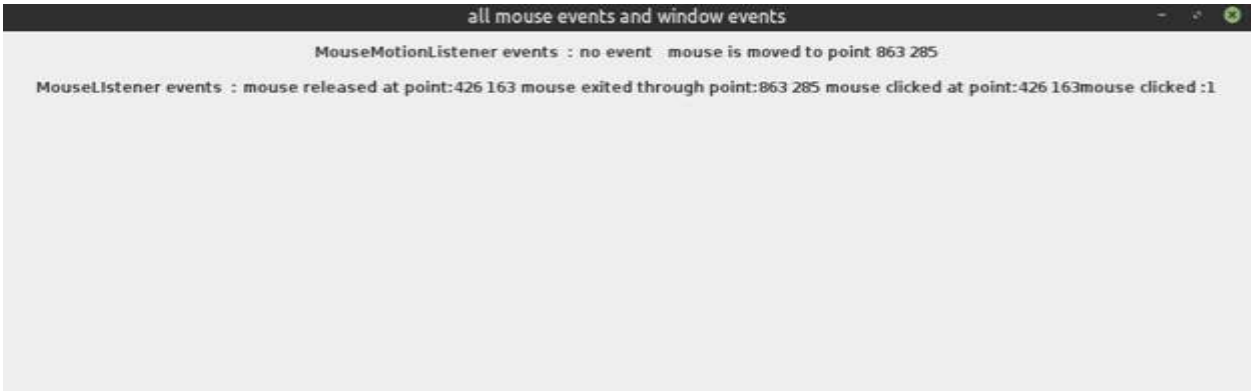
Procedure :

```
//Develop a program to handle all mouse events and window events
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
class Mouse extends Frame implements MouseMotionListener, MouseListener {

    static JLabel label1, label2, label3, label4, label5;

    Mouse()
    {
    }

    public static void main(String[] args)
    {

        JFrame f = new JFrame("all mouse events and window events");

        f.setSize(900, 300);

        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JPanel p = new JPanel();
        JPanel p1 = new JPanel();


        f.setLayout(new FlowLayout());


        JLabel l1, l2;


        l1 = new JLabel("MouseMotionListener events  :");


        l2 = new JLabel("MouseLIstener events  :");


        label1 = new JLabel("no event  ");


        label2 = new JLabel("no event  ");


        label3 = new JLabel("no event  ");


        label4 = new JLabel("no event  ");


        label5 = new JLabel("no event  ");


        Mouse m = new Mouse();


        f.addMouseMotionListener(m);
        f.addMouseListener(m);


        p.add(l1);
        p.add(label1);
        p.add(label2);
        p1.add(l2);
```

```
    p1.add(label3);

    p1.add(label4);

    p1.add(label5);


    f.add(p);

    f.add(p1);


    f.show();

  }



  public void mouseDragged(MouseEvent e)

  {


    label1.setText("mouse is dragged through point "

            + e.getX() + " " + e.getY());

  }


  public void mouseMoved(MouseEvent e)

  {


    label2.setText("mouse is moved to point "

            + e.getX() + " " + e.getY());

  }

  public void mousePressed(MouseEvent e)

  {


    label3.setText("mouse pressed at point:"

            + e.getX() + " " + e.getY());
```

```
}


public void mouseReleased(MouseEvent e)
{


   label3.setText("mouse released at point:"
         + e.getX() + " " + e.getY());
}


public void mouseExited(MouseEvent e)
{


   label4.setText("mouse exited through point:"
         + e.getX() + " " + e.getY());
}
public void mouseEntered(MouseEvent e)
{


   label4.setText("mouse entered at point:"
         + e.getX() + " " + e.getY());
}


public void mouseClicked(MouseEvent e)
{


   label5.setText("mouse clicked at point:"
         + e.getX() + " "
         + e.getY() + "mouse clicked :" + e.getClickCount());   }}
```

Output Screenshot :



```
MouseMotionListener events  : no event   mouse is moved to point 863 285
MouseListener events  : mouse released at point:426 163 mouse exited through point:863 285 mouse clicked at point:426 163mouse clicked :1
```

Result :

The program was executed and the result was successfully obtained. Thus CO5 was obtained.

Experiment No.: 39

Aim :

Write a program to write to a file, then read from the file and display the contents on the console.

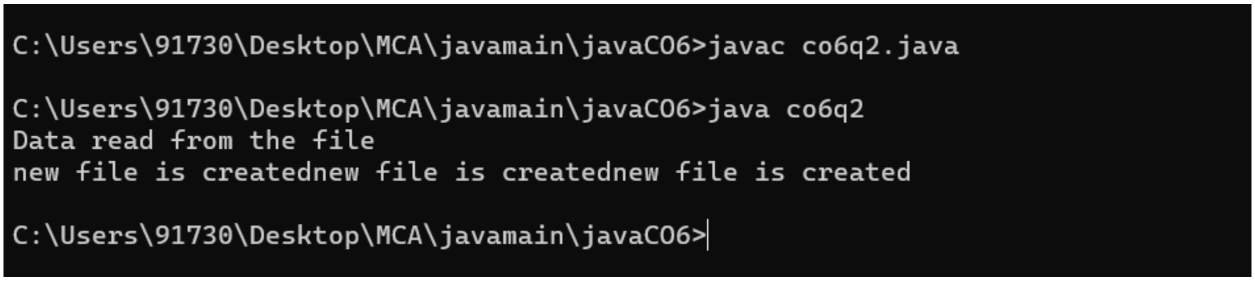
CO6 :

Design applications using files and networking concepts.

Procedure :

```
//program to write to a file, then read from the file and display the contents on the console


import java.io.BufferedReader;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;


public class co6q2 {
public static void main(String[] args) {


try {


FileWriter w = new FileWriter("java_write.txt",true);
w.write("new file is created");
w.close();


FileReader reader = new FileReader("java_write.txt");


BufferedReader br= new BufferedReader(reader);


String line;
System.out.println("Data read from the file");
while ((line = br.readLine()) != null) {
System.out.println(line);
}
reader.close();


} catch (IOException e) {
System.out.println("-----Error-----");
}}}
```

Output Screenshot :



```
C:\Users\91730\Desktop\MCA\javamain\javaCO6>javac co6q2.java

C:\Users\91730\Desktop\MCA\javamain\javaCO6>java co6q2
Data read from the file
new file is creatednew file is creatednew file is created

C:\Users\91730\Desktop\MCA\javamain\javaCO6>
```

Result :

The program was executed and the result was successfully obtained. Thus CO6 was obtained.

Experiment No.: 40

Aim :

Write a program to copy one file to another.

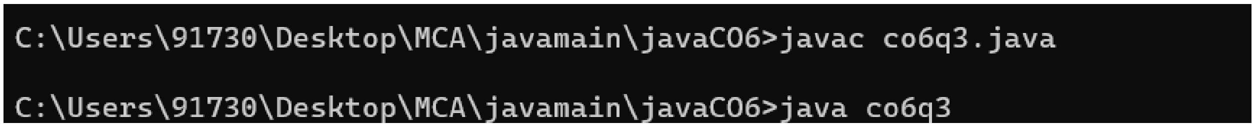
CO6:

Design applications using files and networking concepts.

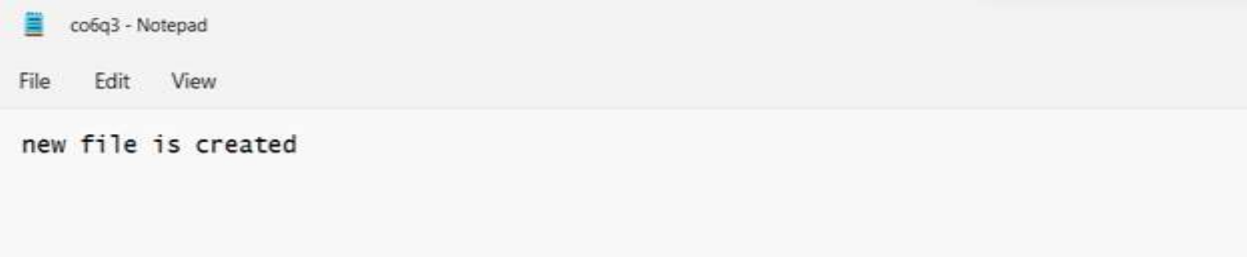
Procedure :

```
//program to copy one file to another-


import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;


public class co6q3 {
public static void main(String[] args) throws IOException{




FileInputStream fileinput = new FileInputStream("source.txt");
FileOutputStream fileoutput = new FileOutputStream("destination.txt");


int i;
while((i = fileinput.read()) != -1)
{
   fileoutput.write(i);
}
System.out.println("copied");
fileinput.close();
fileoutput.close();
}}
```

Output Screenshot :



```
C:\Users\91730\Desktop\MCA\javamain\javaCO6>javac co6q3.java

C:\Users\91730\Desktop\MCA\javamain\javaCO6>java co6q3
```



```
new file is created
```

Result :

The program was executed and the result was successfully obtained. Thus CO6 was obtained.

Experiment No.: 41

Aim :

Write a program that reads from a file having integers. Copy even numbers and odd numbers to separate files.

CO6:

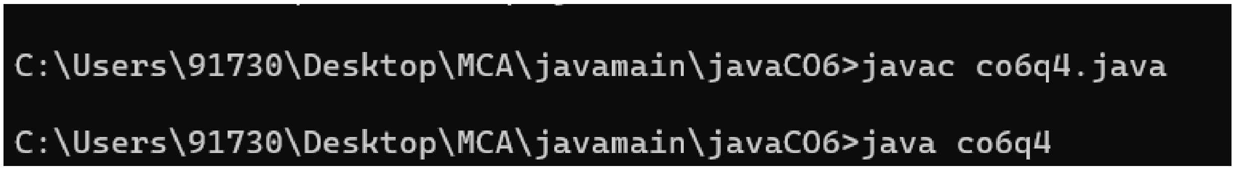
Design applications using files and networking concepts.

Procedure :

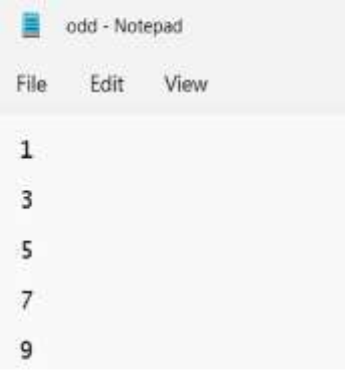
```
//Prgm that reads from a file having integers. Copy even numbers and odd numbers to separate files.

import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;

public class co6q4 {
public static void main(String[] args) throws IOException {

FileInputStream fileinput = new FileInputStream ("source.txt");
FileOutputStream destination_odd = new FileOutputStream ("odd.txt");
FileOutputStream destination_even = new FileOutputStream("even.txt");

int i;
while((i = fileinput.read()) != -1){
if(i%2==0) {
destination_even.write(i);
}
else {
```

```
destination_odd.write(i);
}
}
System.out.println("copied");


fileinput.close();
destination_even.close();
destination_odd.close();
}}
```

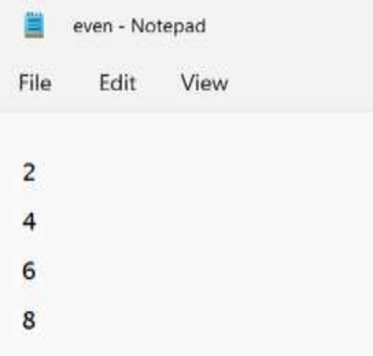
Output Screenshot :



```
C:\Users\91730\Desktop\MCA\javamain\javaCO6>javac co6q4.java

C:\Users\91730\Desktop\MCA\javamain\javaCO6>java co6q4
```



```
1
3
5
7
9
```



```
2
4
6
8
```

Result :

The program was executed and the result was successfully obtained. Thus CO6 was obtained.

Experiment No.: 42

Aim :

Client Server communication using DatagramSocket - UDP

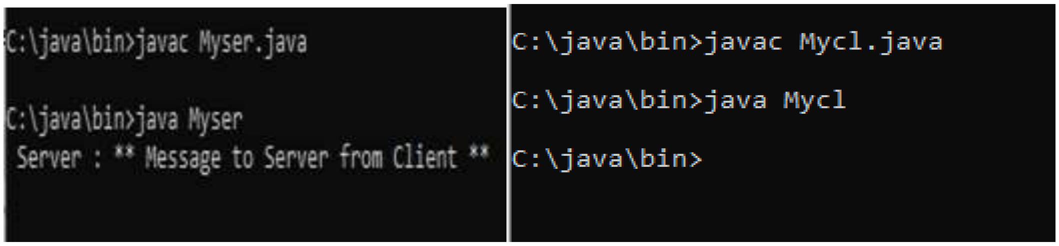
CO6:

Design applications using files and networking concepts.

Procedure :

```
import java.io.*;
import java.net.*;
public class Myser {
    public static void main(String[] args) throws IOException {
        DatagramSocket server=new DatagramSocket(9000);
        byte[] buf=new byte[256];
        DatagramPacket packet=new DatagramPacket(buf,buf.length);
        server.receive(packet);
        String response =new String(packet.getData());
        System.out.println(" Server : "+response);
        server.close();
    }
}
import java.io.*;
import java.net.*;
public class Mycl {
    public static void main(String[] args) throws IOException {
        DatagramSocket client= new DatagramSocket();
        InetAddress add=InetAddress.getByName("localhost");
        String str ="** Message to Server from Client **";
        byte[] bufBytes = str.getBytes();
        DatagramPacket datagramPacket=new DatagramPacket(bufBytes,bufBytes.length,add,9000);
        client.send(datagramPacket);
        client.close();
    }}
```

Output Screenshot :



```
C:\java\bin>javac Myser.java

C:\java\bin>java Myser
 Server : ** Message to Server from Client **
```

```
C:\java\bin>javac Mycl.java

C:\java\bin>java Mycl

C:\java\bin>
```

Result :

The program was executed and the result was successfully obtained. Thus CO6 was obtained.