Experiment No.: 1

Aim: Define a class 'product' with data members pcode, pname and price. Create 3 objects of the class and find the product having the lowest price

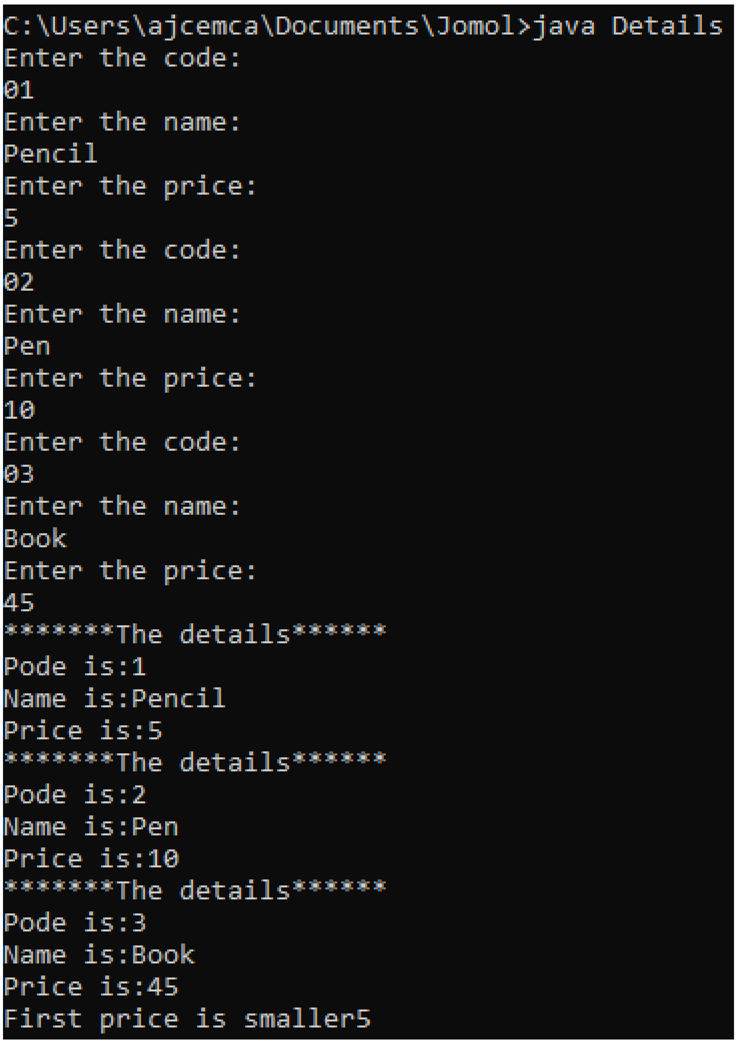
CO1:

Understand object-oriented concepts and design classes and objects to solve problems.

Procedure

```
import java.util.*;

public class product
{
int pcode,price;
String pname;
public void get()
{
Scanner sc=new Scanner(System.in);
System.out.println("Enter pcode: ");
pcode=sc.nextInt();
System.out.println("Enter pname: ");
pname=sc.next();
System.out.println("Enter price: ");
price=sc.nextInt();
}
public void display()
{
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)
{
product p1=new product();
product p2=new product();
product p3=new product();
p1.get();
p2.get();
p3.get();
p1.display();
p2.display();
p3.display();
if(p1.price<p2.price&&p1.price<p3.price)
{
        System.out.println("Price of first product is less");
}
else if(p2.price<p1.price&&p2.price<p3.price)
{
        System.out.println("Price of product 3 is less");
}
else
{
		System.out.println("Price of product 3 is less");
}
}
}
```

Output Screenshot



```
C:\Users\ajcemca\Documents\Jomol>java Details
Enter the code:
01
Enter the name:
Pencil
Enter the price:
5
Enter the code:
02
Enter the name:
Pen
Enter the price:
10
Enter the code:
03
Enter the name:
Book
Enter the price:
45
*******The details******
Pode is:1
Name is:Pencil
Price is:5
*******The details******
Pode is:2
Name is:Pen
Price is:10
*******The details******
Pode is:3
Name is:Book
Price is:45
First price is smaller5
```

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 perform matrix addition.

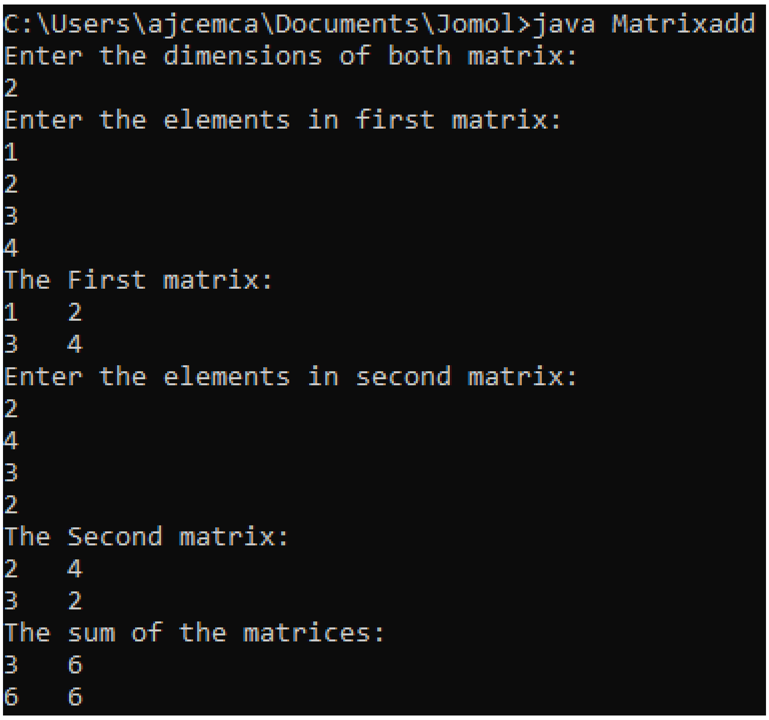
CO1

Understand object-oriented concepts and design classes and objects to solve problems.

Procedure.

```
import java.util.*;
public class Matrixadd{
        public static void main(String[] args){
                Scanner Snr = new Scanner(System.in);
                System.out.println("Enter the dimensions of both matrix: ");
                int rw = Snr.nextInt();
                int cl = rw;
                int Mtx1[][] = new int[rw][cl];
                int Mtx2[][] = new int[rw][cl];
                int Mtx3[][] = new int[rw][cl];
                System.out.println("Enter the elements in first matrix: ");
                for(int i=0;i<rw;i++){
                        for(int j=0;j<cl;j++){
                                Mtx1[i][j] = Snr.nextInt();
                        }
                }
                System.out.println("The First matrix: ");
                for(int i=0;i<rw;i++){
                        for(int j=0;j<cl;j++){
                                System.out.print(Mtx1[i][j] + "   ");
                        }
                        System.out.println(" ");
                }
                System.out.println("Enter the elements in second matrix: ");
                for(int i=0;i<rw;i++){
                        for(int j=0;j<cl;j++){
                                Mtx2[i][j] = Snr.nextInt();
                        }
                }
                System.out.println("The Second matrix: ");
                for(int i=0;i<rw;i++){
                        for(int j=0;j<cl;j++){
                                System.out.print(Mtx2[i][j] + "   ");
                        }
                        System.out.println(" ");
                }
                System.out.println("The sum of the matrices: ");
                for(int i=0;i<rw;i++){
                        for(int j=0;j<cl;j++){
                                Mtx3[i][j] = Mtx1[i][j] + Mtx2[i][j];
                        }
                }
                for(int i=0;i<rw;i++){
                        for(int j=0;j<cl;j++){
                                System.out.print(Mtx3[i][j] + "   ");
                        }
                        System.out.println(" ");
                }
        }
}
```

Output Screenshot



```
C:\Users\ajcemca\Documents\Jomol>java Matrixadd
Enter the dimensions of both matrix:
2
Enter the elements in first matrix:
1
2
3
4
The First matrix:
1   2
3   4
Enter the elements in second matrix:
2
4
3
2
The Second matrix:
2   4
3   2
The sum of the matrices:
3   6
6   6
```

Result

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

Experiment No.: 3

Aim: Add complex numbers

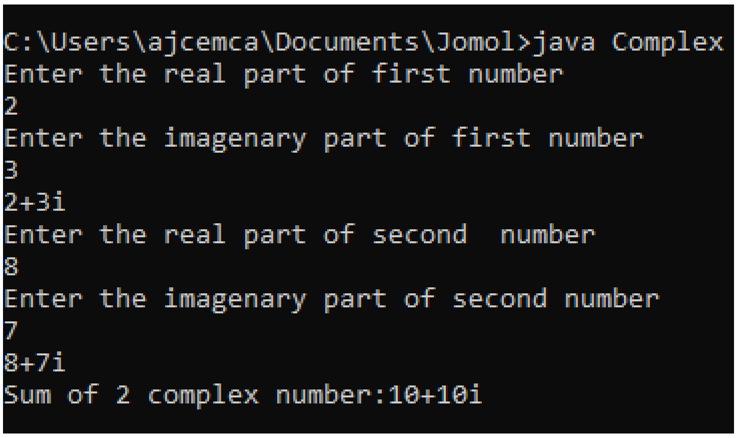
CO1

Understand object-oriented concepts and design classes and objects to solve problems.

Procedure

```
import java.util.*;
public class Complex
{
        public static void main(String args[])
        {
                Scanner obj=new Scanner(System.in);
                System.out.println("Enter the real part of first number");
                int real1=obj.nextInt();
                System.out.println("Enter the imagenary part of first number");
                int img1=obj.nextInt();
                System.out.println(real1+"+"+img1+"i");
                System.out.println("Enter the real part of second  number");
                int real2=obj.nextInt();
                System.out.println("Enter the imagenary part of second number");
                int img2=obj.nextInt();
                System.out.println(real2+"+"+img2+"i");
                int real=real1+real2;
                int img=img1+img2;
                System.out.println("Sum of 2 complex number:"+real+"+"+img+"i");
        }
}
```

Output Screenshot



```
C:\Users\ajcemca\Documents\Jomol>java Complex
Enter the real part of first number
2
Enter the imagenary part of first number
3
2+3i
Enter the real part of second  number
8
Enter the imagenary part of second number
7
8+7i
Sum of 2 complex number:10+10i
```

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

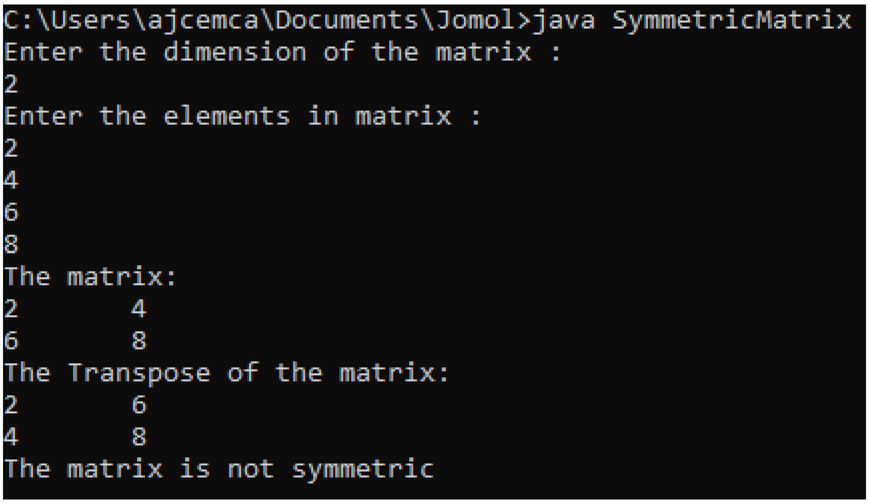
Understand object-oriented concepts and design classes and objects to solve problems.

Procedure

```
import java.util.*;
public class SymMatrix
{
            public static void main(String[] args)
            {
                        Scanner Snr = new Scanner(System.in);
                        System.out.println("Enter the dimension of the matrix :");
                        int sz = Snr.nextInt();
                        int Arr[][] = new int[sz][sz];
                        int Arr1[][] = new int[sz][sz];
                        System.out.println("Enter the elements in matrix :");
                        for(int i=0;i<sz;i++)
                        {
                        for(int j=0;j<sz;j++)
                                    {
                                                Arr[i][j] = Snr.nextInt();
                                    }}
                        System.out.println("The matrix: ");
                        for(int i=0;i<sz;i++)
                        {
                                    for(int j=0;j<sz;j++)
                                    {
                                                System.out.print(Arr[i][j] + "              ");
                                    }
                                    System.out.println(" ");}
                        System.out.println("The Transpose of the matrix: ");
                        for(int i=0;i<sz;i++)
                        {
                                    for(int j=0;j<sz;j++)
                                    {
                                                Arr1[i][j]=Arr[j][i];
                                    }
                        }
                                                for(int i=0;i<sz;i++)
                                                {
                                    for(int j=0;j<sz;j++)
                                    {
                                                System.out.print(Arr1[i][j] + "            ");
                                    }
                                    System.out.println(" ");
```

```
		}
		int flag=0;
		for(int i=0;i<sz;i++)
		{
			for(int j=0;j<sz;j++)
			{
				if(Arr[i][j] != Arr1[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\ajcemca\Documents\Jomol>java SymmetricMatrix
Enter the dimension of the matrix :
2
Enter the elements in matrix :
2
4
6
8
The matrix:
2       4
6       8
The Transpose of the matrix:
2       6
4       8
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

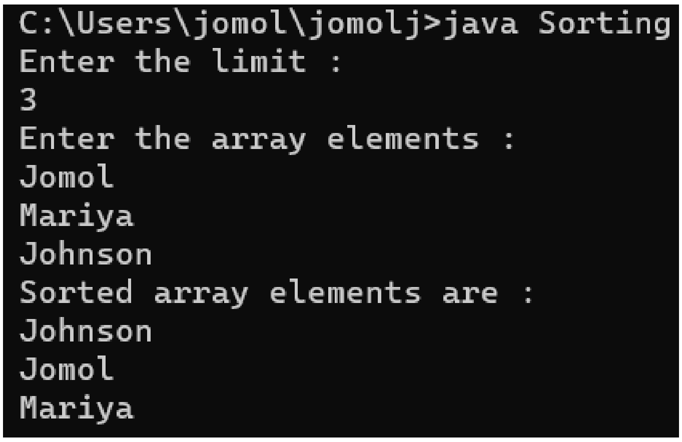
CO2

Implement arrays and strings

Procedure

```
import java.util.*;
public class StrCom
{
public static void main(String[] args)
{
            String temp;

Scanner sm=new Scanner(System.in);
System.out.println("Enter the limit");
int l=sm.nextInt();
System.out.println("Enter the string: ");
String a[]=new String[l];
for(int i=0;i<l;i++)
{
            a[i]=sm.nextLine();
}
for(int i=0;i<l;i++)
{
for(int j=i+1;j<l;j++)
{
            if(a[i].compareTo(a[j])>0)
            {
               temp=a[i];
                        a[i]=a[j];
                        a[j]=temp;
            }
}
}
System.out.println("Sorted form");
for(int i=0;i<l;i++)
{
            System.out.println(a[i]);
}
}
}
```

Output Screenshot



```
C:\Users\jomol\jomolj>java Sorting
Enter the limit :
3
Enter the array elements :
Jomol
Mariya
Johnson
Sorted array elements are :
Johnson
Jomol
Mariya
```

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.

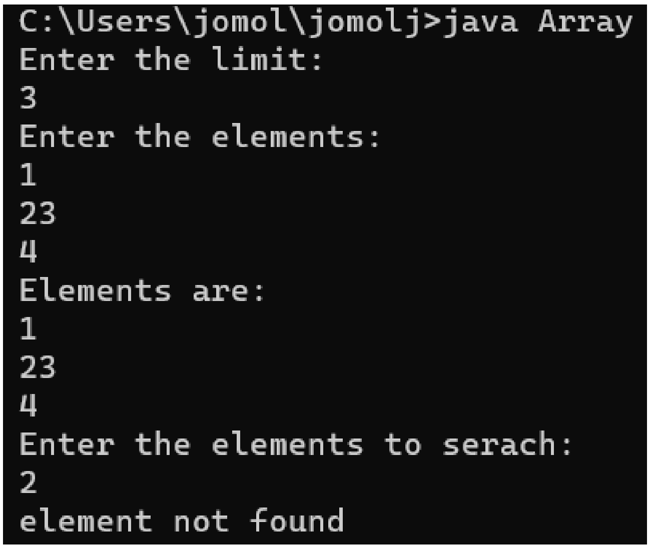
CO2

Implement arrays and strings

Procedure

```
import java.util.*;
public class Array
{
	public static void main(String args[])
	{
		Scanner sc = new Scanner(System.in);
		System.out.println("Enter the size of array : ");
		int n = sc.nextInt();
		int a[] = new int[n];
		int flag=0;
		System.out.println("Enter the array elements : ");
		for(int i=0;i<n;i++)
		{
			a[i] = sc.nextInt();
		}
		System.out.println("Array elements are : ");
		for(int i=0;i<n;i++)
		{
			System.out.print(" "+a[i]);
		}
		System.out.println("\nEnter the element to be searched : ");
		int s = sc.nextInt();
		for(int i=0;i<n;i++)
		{
			if(a[i]==s)
			{
				flag=1;
				System.out.println("\nElement "+s+" found at location 
"+(i+1));
				break;
			}
		}
		if(flag==0)
		{
			System.out.println("Element "+s+" not found ");
		}
	}
}
```

Output Screenshot



```
C:\Users\jomol\jomolj>java Array
Enter the limit:
3
Enter the elements:
1
23
4
Elements are:
1
23
4
Enter the elements to serach:
2
element not found
```

Result

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

Experiment No.: 7

Aim: Perform string manipulations

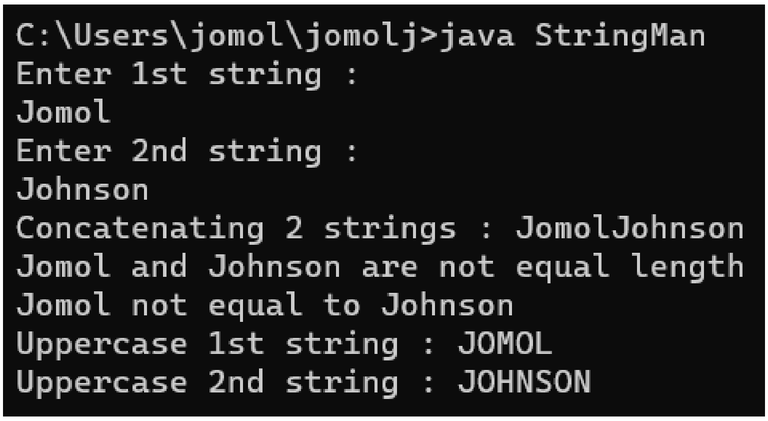
CO2

Implement arrays and strings

Procedure

```
import java.util.*;
public class StringM
{

public static void main(String[] args)
{
Scanner sm = new Scanner(System.in);
System.out.println("Enter the string: ");
String a=sm.nextLine();
System.out.println("Enter the second string: ");
String b=sm.nextLine();
System.out.println("Concatination of strings: "+a.concat(b));
System.out.println("lenth:"+a.length());
System.out.println("Length:"+b.length());
System.out.println("Uppercase: "+a.toUpperCase());
System.out.println("Lowercase: "+a.toLowerCase());
System.out.println("Uppercase: "+b.toUpperCase());
System.out.println("Lowercase: "+b.toLowerCase());
System.out.println("Strings equal? "+a.equals(b));
System.out.println("Substring from first string: "+a.substring(3));
System.out.println("Substring from first string: "+b.substring(3));
}}
```

Output Screenshot



```
C:\Users\jomol\jomolj>java StringMan
Enter 1st string :
Jomol
Enter 2nd string :
Johnson
Concatenating 2 strings : JomolJohnson
Jomol and Johnson are not equal length
Jomol not equal to Johnson
Uppercase 1st string : JOMOL
Uppercase 2nd string : JOHNSON
```

Result

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

Experiment No.: 8

Aim: 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.

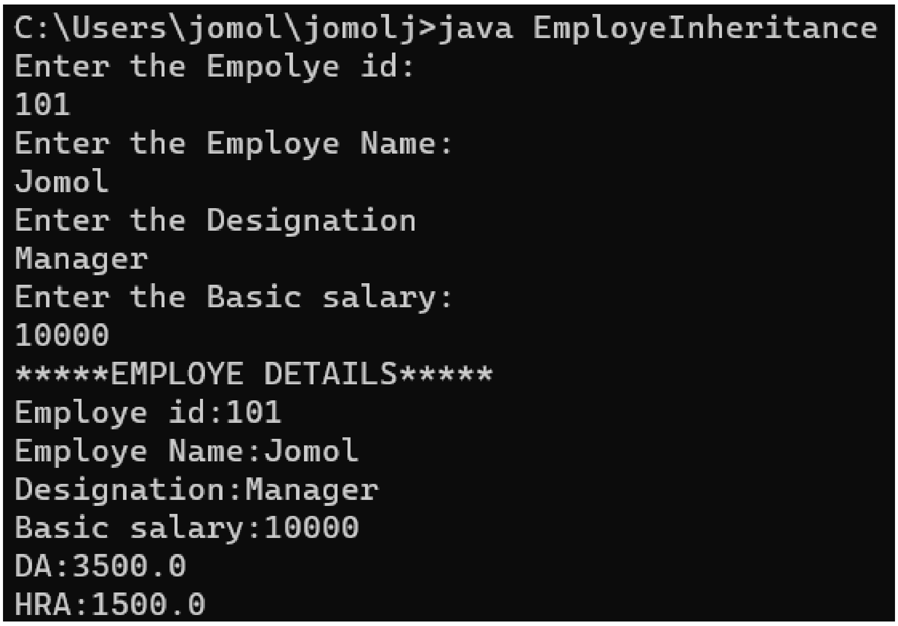
CO2

Implement arrays and strings

Procedure

```
import java.util.*;
class Employe
{
        int empid;
        String empname;
        String designation;
        Employe()
        {
                Scanner sc=new Scanner(System.in);
                System.out.println("Enter the Empolye id:");
                empid=sc.nextInt();
                System.out.println("Enter the Employe Name:");
                empname=sc.next();
                System.out.println("Enter the Designation");
                designation=sc.next();
        }
}
class Salary extends Employe
{
                int basic;
                float DA,HRA,gross_salary;
                Salary()
                {
                        Scanner sc=new Scanner(System.in);
                        System.out.println("Enter the Basic salary:");
                        basic=sc.nextInt();
                        DA=(basic*35)/100;
                        HRA=(basic*15)/100;
                        gross_salary=basic+DA+HRA;
                }
                public void display()
                {
                        System.out.println("*****EMPLOYE DETAILS*****");
                        System.out.println("Employe id:"+empid);
                        System.out.println("Employe Name:"+empname);
                        System.out.println("Designation:"+designation);
                        System.out.println("Basic salary:"+basic);
                        System.out.println("DA:"+DA);
                        System.out.println("HRA:"+HRA);
                }
}
public class EmployeInheritance
{
        public static void main(String atgs[])
        {
                Salary obj=new Salary();
                obj.display();
        }
}
```

Output Screenshot



```
C:\Users\jomol\jomolj>java EmployeInheritance
Enter the Empolye id:
101
Enter the Employe Name:
Jomol
Enter the Designation
Manager
Enter the Basic salary:
10000
*****EMPLOYE DETAILS*****
Employe id:101
Employe Name:Jomol
Designation:Manager
Basic salary:10000
DA:3500.0
HRA:1500.0
```

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

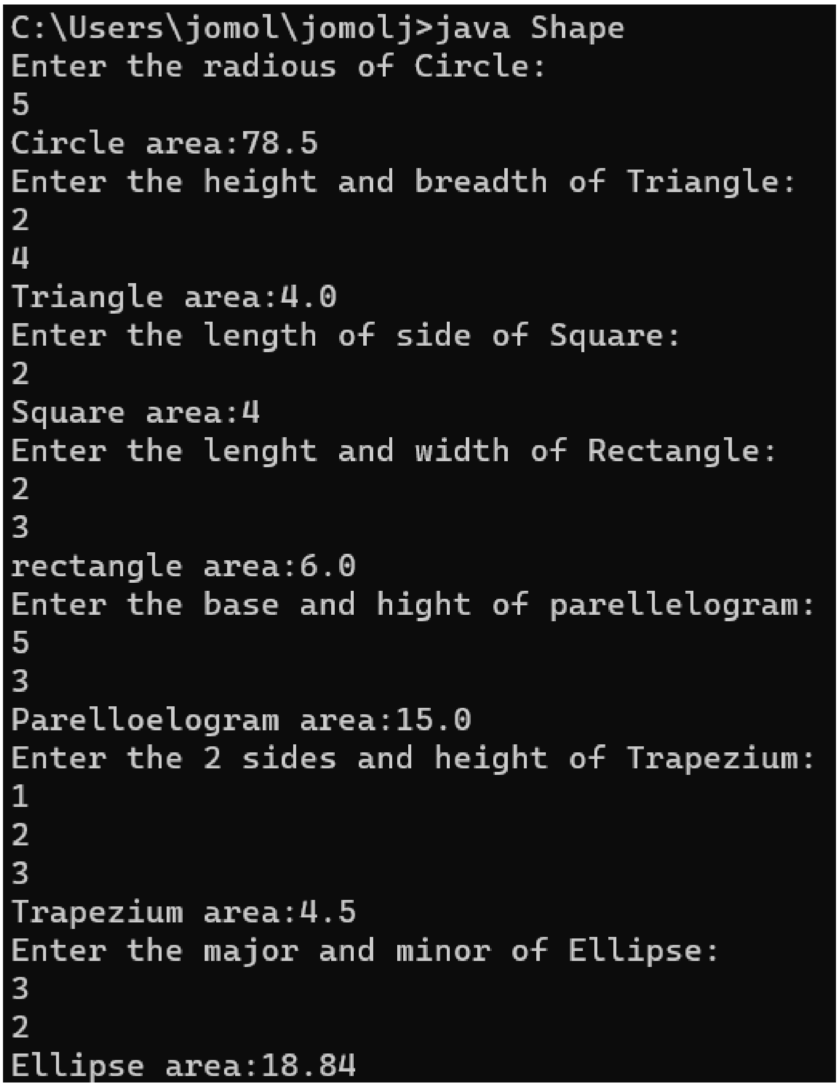
Implement object-oriented concepts like inheritance, overloading and interfaces.

Procedure

```
import java.util.*;
class Area
{
   int a,b,h;
   double r,l,w;
   public void ashape(double r)
   {
      double radious=3.14*(r*r);
      System.out.println("Circle area:"+radious);
   }
   public void ashape(int b,int h)
   {
      double radious=0.5*b*h;
      System.out.println("Triangle area:"+radious);
   }
   public void ashape(int a)
   {
      int radious=a*a;
      System.out.println("Square area:"+radious);
   }
   public void ashape(double l,double w)
   {
      double radious=l*w;
      System.out.println("rectangle area:"+radious);
   }
   ////////////////////////////////////////////////////////////
   public void ashape(double b,int h)
   {
      double radious=b*h;
      System.out.println("Parelloelogram area:"+radious);
   }
   public void ashape(int a,int b,double h)
   {
      double radious=0.5*(a+b)*h;
      System.out.println("Trapezium area:"+radious);
   }
   public void ashape(float a,float b)
   {
      double radious=3.14*a*b;;
      System.out.println("Ellipse area:"+radious);
   }
}
```

```
public class Shape
{
   public static void main(String args[])
   {
      Scanner sc=new Scanner(System.in);
      Area obj=new Area();
      System.out.println("Enter the radious of Circle:");
      double p=sc.nextDouble();
      obj.ashape(p);
      System.out.println("Enter the height and breadth of Triangle:");
      int q=sc.nextInt();
      int r=sc.nextInt();
      obj.ashape(q,r);
      System.out.println("Enter the length of side of Square:");
      int e=sc.nextInt();
      obj.ashape(e);
      System.out.println("Enter the lenght and width of Rectangle:");
      double f=sc.nextDouble();
      double g=sc.nextDouble();
      obj.ashape(f,g);
      System.out.println("Enter the base and hight of parellelogram:");
      double b=sc.nextDouble();
      int h=sc.nextInt();
      obj.ashape(b,h);
      System.out.println("Enter the 2 sides and height of Trapezium:");
      int a=sc.nextInt();
      int s=sc.nextInt();
      double j=sc.nextDouble();
      obj.ashape(a,s,j);
      System.out.println("Enter the major and minor of Ellipse:");
      float u=sc.nextFloat();
      float d=sc.nextFloat();
      obj.ashape(u,d);
   }
}
```

Output Screenshot



```
C:\Users\jomol\jomolj>java Shape
Enter the radious of Circle:
5
Circle area:78.5
Enter the height and breadth of Triangle:
2
4
Triangle area:4.0
Enter the length of side of Square:
2
Square area:4
Enter the lenght and width of Rectangle:
2
3
rectangle area:6.0
Enter the base and hight of parellelogram:
5
3
Parelloelogram area:15.0
Enter the 2 sides and height of Trapezium:
1
2
3
Trapezium area:4.5
Enter the major and minor of Ellipse:
3
2
Ellipse area:18.84
```

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

Implement object-oriented concepts like inheritance, overloading and interfaces.

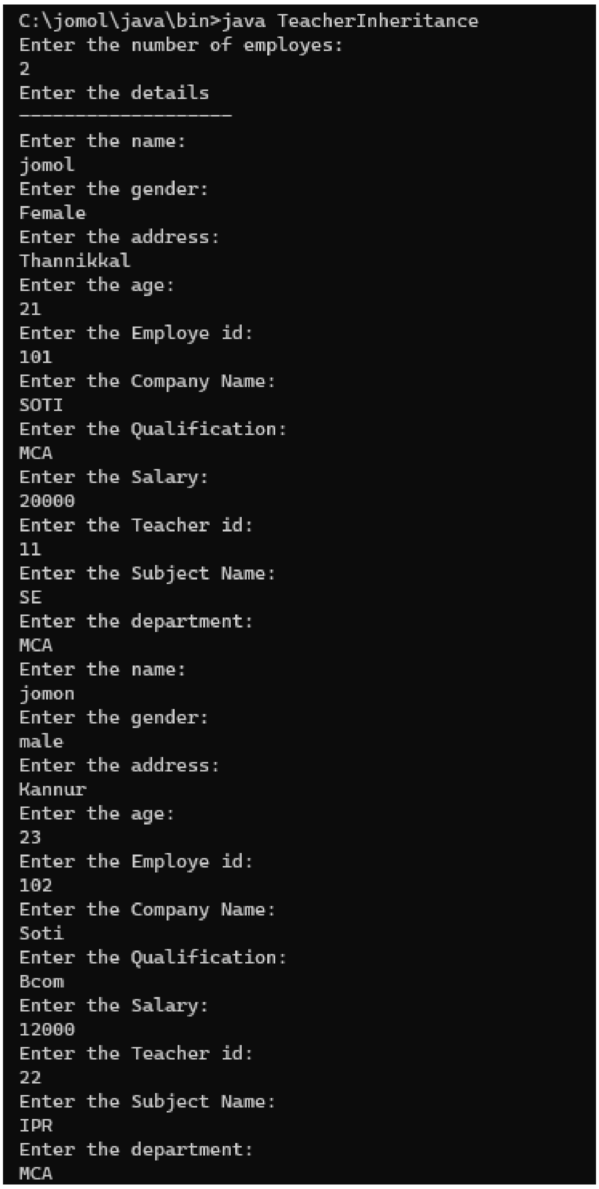
Procedure

```
import java.util.*;
class Person
{
   String name;
   String gender;
   String address;
   int age;
   Person()
   {
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter the name:");
      name=sc.next();
      System.out.println("Enter the gender:");
      gender=sc.next();
      System.out.println("Enter the address:");
      address=sc.next();
      System.out.println("Enter the age:");
      age=sc.nextInt();
   }
}
class Employe extends Person
{
   int empid;
   String company_name;
   String qualification;
   int salary;
   Employe()
   {
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter the Employe id:");
      empid=sc.nextInt();
      System.out.println("Enter the Company Name:");
      company_name=sc.next();
      System.out.println("Enter the Qualification:");
      qualification=sc.next();
      System.out.println("Enter the Salary:");
```

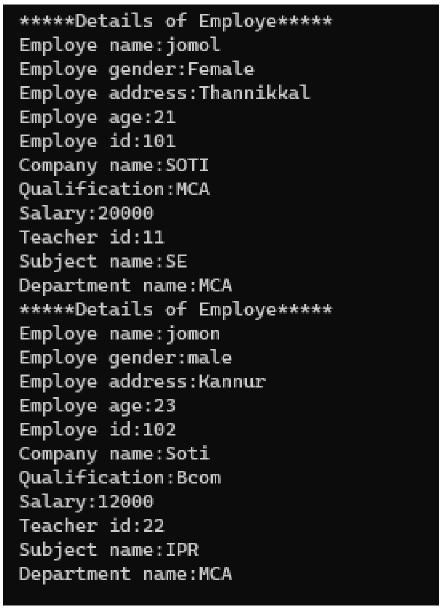
```
        salary=sc.nextInt();
    }
}
class Teacher extends Employe
{
    int teacher_id;
    String subject;
    String department;
    Teacher()
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the Teacher id:");
        teacher_id=sc.nextInt();
        System.out.println("Enter the Subject Name:");
        subject=sc.next();
        System.out.println("Enter the department:");
        department=sc.next();
    }
    public void display()
    {
        System.out.println("*****Details of Employe*****");
        System.out.println("Employe name:"+name);
        System.out.println("Employe gender:"+gender);
        System.out.println("Employe address:"+address);
        System.out.println("Employe age:"+age);
        System.out.println("Employe id:"+empid);
        System.out.println("Company name:"+company_name);
        System.out.println("Qualification:"+qualification);
        System.out.println("Salary:"+salary);
        System.out.println("Teacher id:"+teacher_id);
        System.out.println("Subject name:"+subject);
        System.out.println("Department name:"+department);
    }
}
public class TeacherInheritance
{
    public static void main(String args[])
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the number of employes:");
        int n=sc.nextInt();
        Teacher obj[]=new Teacher[n];
        System.out.println("Enter the details");
        System.out.println("-------------------");
        for(int i=0;i<n;i++)
        {
            obj[i]=new Teacher();
        }
        for(int i=0;i<n;i++)
        {
```

```
        obj[i].display();
    } }
```

Output Screenshot



```
C:\jomol\java\bin>java TeacherInheritance
Enter the number of employes:
2
Enter the details
-------------------
Enter the name:
jomol
Enter the gender:
Female
Enter the address:
Thannikkal
Enter the age:
21
Enter the Employe id:
101
Enter the Company Name:
SOTI
Enter the Qualification:
MCA
Enter the Salary:
20000
Enter the Teacher id:
11
Enter the Subject Name:
SE
Enter the department:
MCA
Enter the name:
jomon
Enter the gender:
male
Enter the address:
Kannur
Enter the age:
23
Enter the Employe id:
102
Enter the Company Name:
Soti
Enter the Qualification:
Bcom
Enter the Salary:
12000
Enter the Teacher id:
22
Enter the Subject Name:
IPR
Enter the department:
MCA
```



```
*****Details of Employe*****
Employe name:jomol
Employe gender:Female
Employe address:Thannikkal
Employe age:21
Employe id:101
Company name:SOTI
Qualification:MCA
Salary:20000
Teacher id:11
Subject name:SE
Department name:MCA
*****Details of Employe*****
Employe name:jomon
Employe gender:male
Employe address:Kannur
Employe age:23
Employe id:102
Company name:Soti
Qualification:Bcom
Salary:12000
Teacher id:22
Subject name:IPR
Department name:MCA
```

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

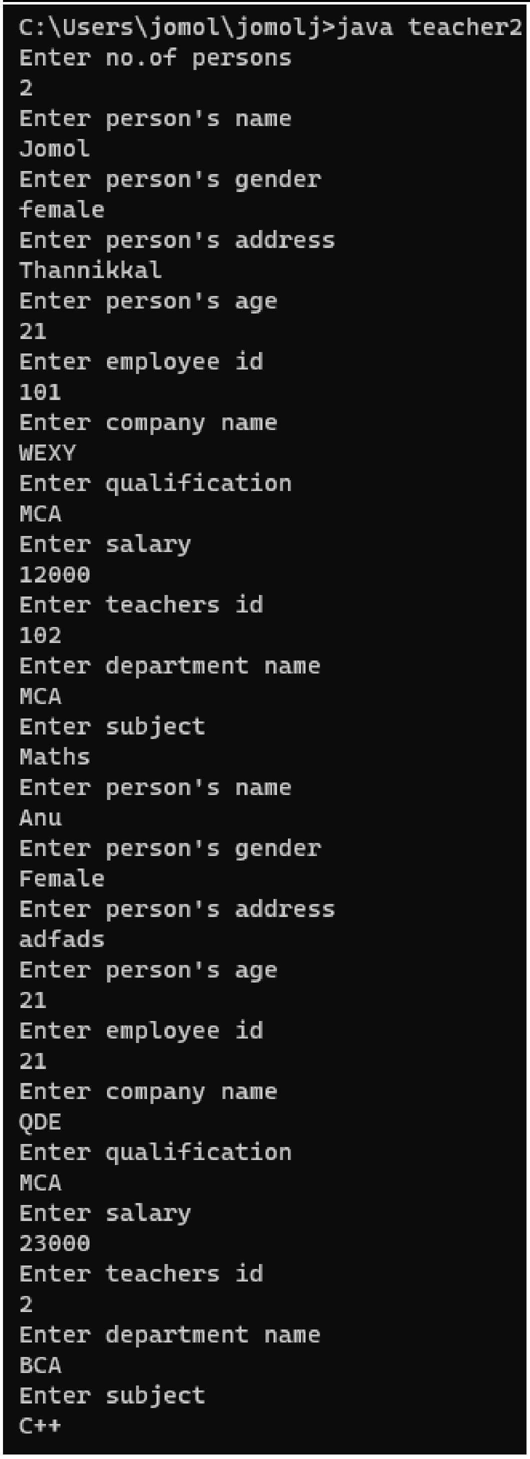
Implement object-oriented concepts like inheritance, overloading and interfaces.

Procedure

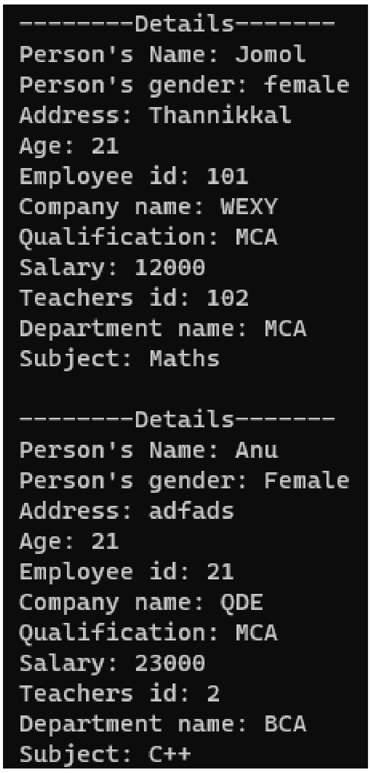
```
import java.util.*;
class person
{
            int age;
            String name,gender,add;
             person()
            {
                        Scanner sc =new Scanner(System.in);
                        System.out.println("Enter person's name");
                        name=sc.next();
                        System.out.println("Enter person's gender");
                        gender=sc.next();
                        System.out.println("Enter person's address");
                        add=sc.next();
                        System.out.println("Enter person's age");
      age=sc.nextInt();
            }
}
class employee extends person
{
            int id,sal;
            String company,quali;
 employee()
            {
                        Scanner sc=new Scanner(System.in);
                        System.out.println("Enter employee id");
      id=sc.nextInt();
                        System.out.println("Enter company name");
                        company=sc.next();
                        System.out.println("Enter qualification");
                        quali=sc.next();
                        System.out.println("Enter salary");
      sal=sc.nextInt();
```

```
                }
}
class teacher2 extends employee
{
                int tid;
                String dep,sub;
                teacher2()
                {
      Scanner sc=new Scanner(System.in);
                                System.out.println("Enter teachers id");
        tid=sc.nextInt();
                                System.out.println("Enter department name");
                                dep=sc.next();
                                System.out.println("Enter subject");
                                sub=sc.next();
                }
                void display()
                {
                     System.out.println();
                                 System.out.println("--------Details-------");
                                System.out.println("Person's Name: "+name);
                                System.out.println("Person's gender: "+gender);
                                System.out.println("Address: "+add);
                                System.out.println("Age: "+age);
                                System.out.println("Employee id: "+id);
                                System.out.println("Company name: "+company);
                                System.out.println("Qualification: "+quali);
                                System.out.println("Salary: "+sal);
                                System.out.println("Teachers id: "+tid);
                                System.out.println("Department name: "+dep);
                                System.out.println("Subject: "+sub);
                }
                public static void main(String[] args)
                {
                                Scanner sc=new Scanner(System.in);
                                System.out.println("Enter no.of persons");
                                int n=sc.nextInt();
        teacher2 t[]=new teacher2[n];
                                for(int i=0;i<n;i++)
                                {
                                                t[i]=new teacher2();
                                }
                                for(int i=0;i<n;i++)
                                {
          t[i].display();
                                }
                }
}
```

Output Screenshot



```
C:\Users\jomol\jomolj>java teacher2
Enter no.of persons
2
Enter person's name
Jomol
Enter person's gender
female
Enter person's address
Thannikkal
Enter person's age
21
Enter employee id
101
Enter company name
WEXY
Enter qualification
MCA
Enter salary
12000
Enter teachers id
102
Enter department name
MCA
Enter subject
Maths
Enter person's name
Anu
Enter person's gender
Female
Enter person's address
adfads
Enter person's age
21
Enter employee id
21
Enter company name
QDE
Enter qualification
MCA
Enter salary
23000
Enter teachers id
2
Enter department name
BCA
Enter subject
C++
```



```
--------Details-------
Person's Name: Jomol
Person's gender: female
Address: Thannikkal
Age: 21
Employee id: 101
Company name: WEXY
Qualification: MCA
Salary: 12000
Teachers id: 102
Department name: MCA
Subject: Maths

--------Details-------
Person's Name: Anu
Person's gender: Female
Address: adfads
Age: 21
Employee id: 21
Company name: QDE
Qualification: MCA
Salary: 23000
Teachers id: 2
Department name: BCA
Subject: C++
```

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

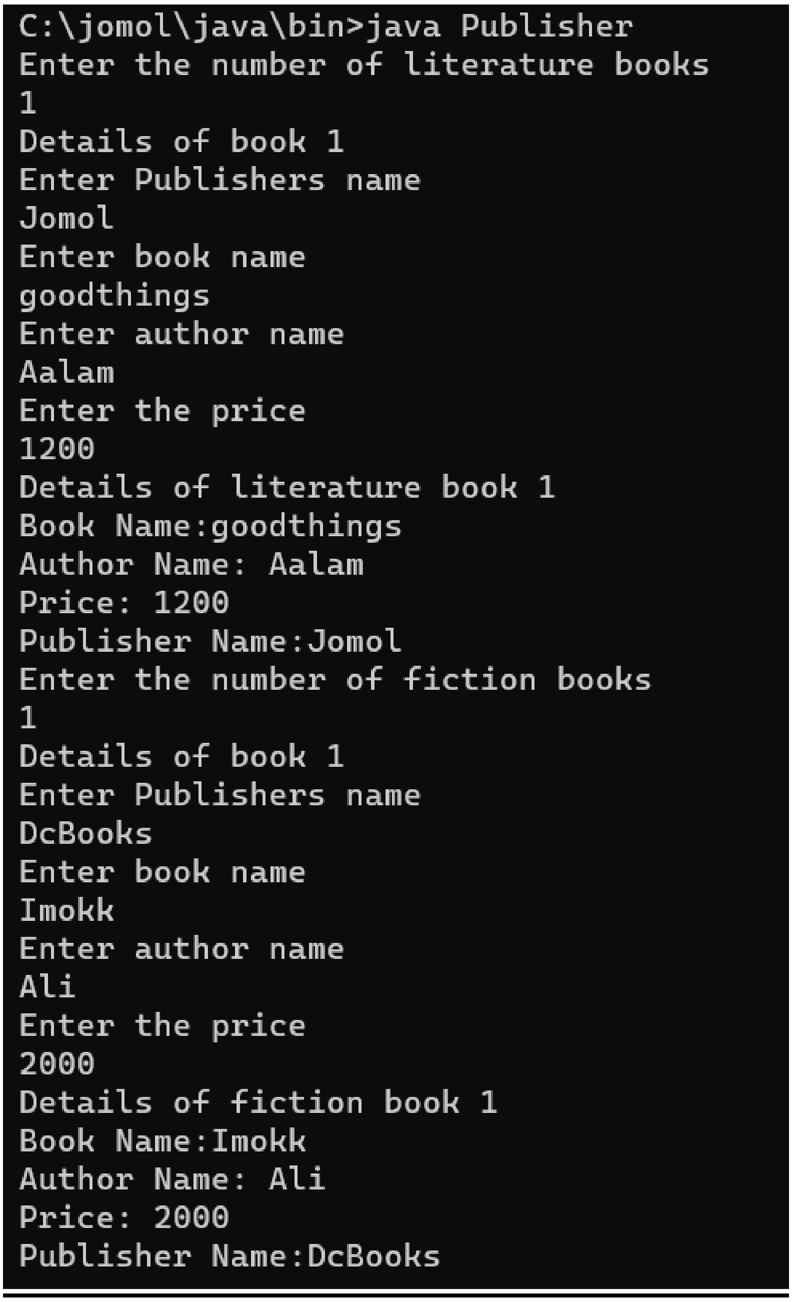
Implement object-oriented concepts like inheritance, overloading and interfaces.

Procedure

```
import java.util.*;
class Publisher
{
        String pname;
        Publisher()
        {
                Scanner sc=new Scanner(System.in);
                System.out.println("Enter Publishers name");
                pname=sc.next();
        }
        public static void main(String[] args)
        {
                int i,n,n1;
                Scanner sc=new Scanner(System.in);
                System.out.println("Enter the number of literature books");
                n=sc.nextInt();
                literature[] l=new literature[n];
                for(i=0;i<n;i++)
                {
                        System.out.println("Details of book "+(i+1));
                l[i]=new literature();
                }
                for(i=0;i<n;i++)
                {
                        System.out.println("Details of literature book "+(i+1));
                l[i].ldisp();
                }
                System.out.println("Enter the number of fiction books");
                n1=sc.nextInt();
                fiction[] f=new fiction[n1];
                for(i=0;i<n1;i++)
                {
                        System.out.println("Details of book "+(i+1));
                f[i]=new fiction();
                }
                for(i=0;i<n1;i++)
                {
                        System.out.println("Details of fiction book "+(i+1));
                f[i].fdisp();
```

```
                        }

            }
}
class book extends Publisher
{
            String author,name;
            int price;
            book()
            {
               Scanner sc=new Scanner(System.in);
                        System.out.println("Enter book name");
                        name=sc.next();
                        System.out.println("Enter author name");
                        author=sc.next();
                        System.out.println("Enter the price");
                        price=sc.nextInt();
            }
}
class literature extends book
{
             void ldisp()
             {
                        System.out.println("Book Name:"+name);
                        System.out.println("Author Name: "+author);
                        System.out.println("Price: "+price);
                        System.out.println("Publisher Name:"+pname);
             }
}
class fiction extends book
{

             void fdisp()
             {
                        System.out.println("Book Name:"+name);
      System.out.println("Author Name: "+author);
                        System.out.println("Price: "+price);
      System.out.println("Publisher Name:"+pname);
             }
}
```

Output Screenshot



```
C:\jomol\java\bin>java Publisher
Enter the number of literature books
1
Details of book 1
Enter Publishers name
Jomol
Enter book name
goodthings
Enter author name
Aalam
Enter the price
1200
Details of literature book 1
Book Name:goodthings
Author Name: Aalam
Price: 1200
Publisher Name:Jomol
Enter the number of fiction books
1
Details of book 1
Enter Publishers name
DcBooks
Enter book name
Imokk
Enter author name
Ali
Enter the price
2000
Details of fiction book 1
Book Name:Imokk
Author Name: Ali
Price: 2000
Publisher Name:DcBooks
```

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

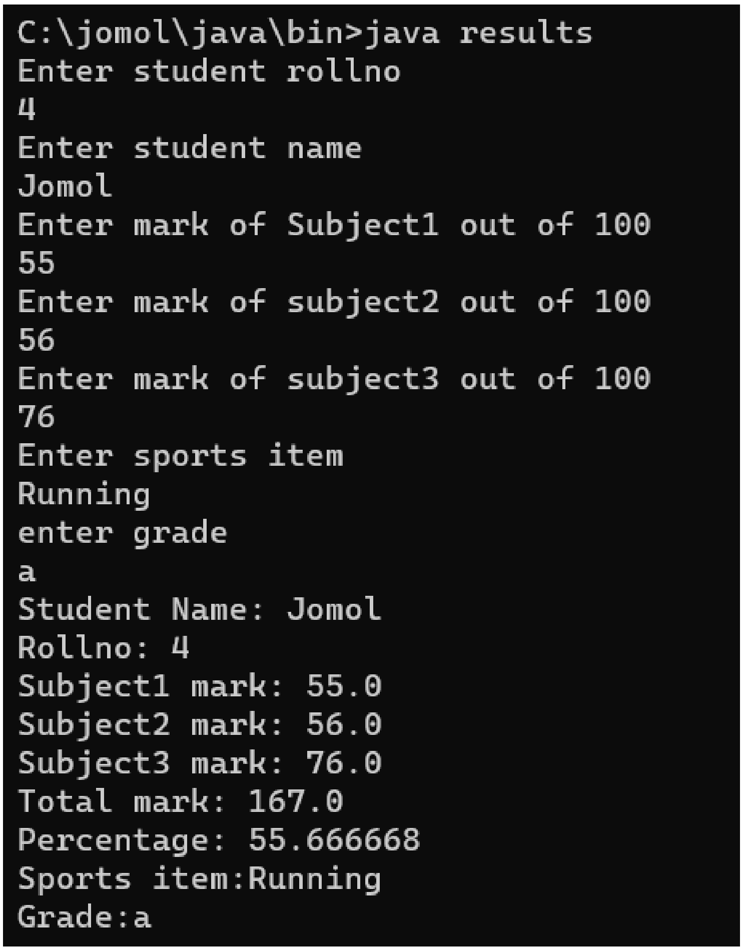
Implement object-oriented concepts like inheritance, overloading and interfaces.

Procedure

```
import java.util.*;
interface student
{
        public void get();
        public void disp();
}
interface sports
{
        public void get1();
        public void disp1();
}
class results implements student,sports
{
        float t,p;
        float m1,m2,m3;
        int rno;
        String name,sport,grade;
        public void get()
        {
                Scanner sc =new Scanner(System.in);
                System.out.println("Enter student rollno");
                rno=sc.nextInt();
                System.out.println("Enter student name");
    name=sc.next();
                System.out.println("Enter mark of Subject1 out of 100");
                m1=sc.nextFloat();
                System.out.println("Enter mark of subject2 out of 100");
                m2=sc.nextFloat();
                System.out.println("Enter mark of subject3 out of 100");
                m3=sc.nextFloat();
        }
        public void get1()
        {
                Scanner sc=new Scanner(System.in);
                System.out.println("Enter sports item");
                sport=sc.next();
                System.out.println("enter grade");
                grade=sc.next();
        }
        void cal()
```

```
		{
		t=m1+m2+m2;
		p=(t/300)*100;
		}
		public void disp()
		{

				System.out.println("Student Name: "+name);
				System.out.println("Rollno: "+rno);
				System.out.println("Subject1 mark: "+m1);
				System.out.println("Subject2 mark: "+m2);
				System.out.println("Subject3 mark: "+m3);
				System.out.println("Total mark: "+t);
				System.out.println("Percentage: "+p);
		}
		public void disp1()
		{
		System.out.println("Sports item:"+sport);
		System.out.println("Grade:"+grade);
		}


public static void main(String[] args)
		{
	results s=new results();
	s.get();
	s.get1();
	s.cal();
	s.disp();
	s.disp1();
		}
}
```

Output Screenshot



```
C:\jomol\java\bin>java results
Enter student rollno
4
Enter student name
Jomol
Enter mark of Subject1 out of 100
55
Enter mark of subject2 out of 100
56
Enter mark of subject3 out of 100
76
Enter sports item
Running
enter grade
a
Student Name: Jomol
Rollno: 4
Subject1 mark: 55.0
Subject2 mark: 56.0
Subject3 mark: 76.0
Total mark: 167.0
Percentage: 55.666668
Sports item:Running
Grade:a
```

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.

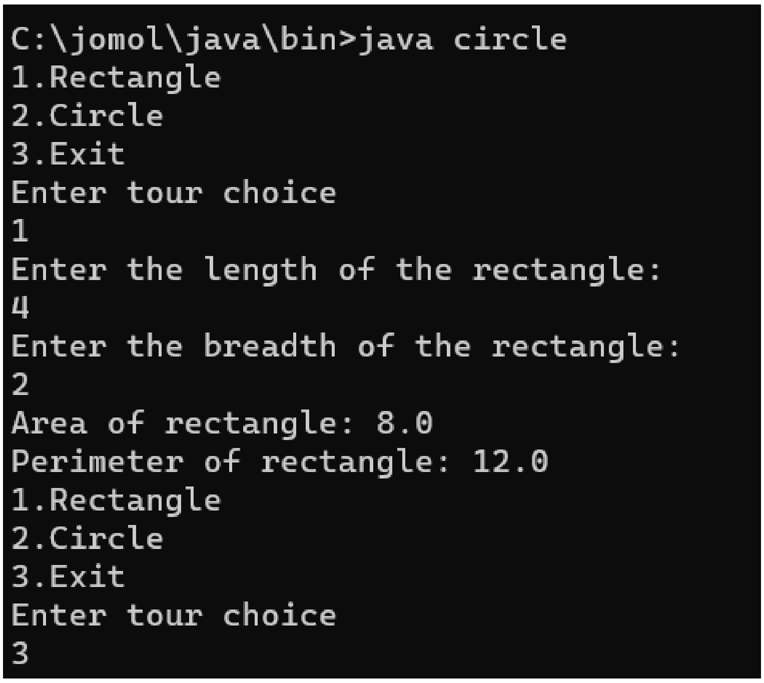
CO3

Implement object-oriented concepts like inheritance, overloading and interfaces.

Procedure

```
import java.util.*;
interface proto
{
        public void area();
        public void perimeter();
        public void get();

}
class rect implements proto
{
        int l,b;
        double r,peri;
        public void get()
        {
         Scanner sc=new Scanner(System.in);
         System.out.println("Enter the length of the rectangle: ");
         l = sc.nextInt();
         System.out.println("Enter the breadth of the rectangle: ");
         b =sc.nextInt();
        }
        public void area()
        {
         r=l*b;
         System.out.println("Area of rectangle: "+r);
        }
        public void perimeter()
        {
                peri=(l+b)*2;
                System.out.println("Perimeter of rectangle: "+peri);
        }
}
 class circle implements proto
 {
         double c,pe,r;
         public void get()
         {
                Scanner sc=new Scanner(System.in);
                System.out.println("Enter the radius");
                r=sc.nextInt();
                }
                public void area()
                {
                c=3.14*(r*r);
                 System.out.println("Area of circle: "+c);
                }
                public void perimeter()
                {
                               pe=2*(3.14*(r*r));
                               System.out.println("Perimeter of circle: "+pe);
                }


public static void main(String[] args)
{
               int c;
               Scanner sc=new Scanner(System.in);

                rect re=new rect();
     circle ci=new circle();
               do
               {
                              System.out.println("1.Rectangle"+"\n"+"2.Circle"+"\n"+"3.Exit");
                              System.out.println("Enter tour choice");
                              c=sc.nextInt();
                              switch(c)
                              {
                                             case 1:
                                              re.get();
                                              re.area();
                                              re.perimeter();
                                              break;
                                              case 2:
                                              ci.get();
                                              ci.area();
                                              ci.perimeter();
                                              break;
                              }
               }
               while(c!=2);
}
}
```

Output Screenshot



```
C:\jomol\java\bin>java circle
1.Rectangle
2.Circle
3.Exit
Enter tour choice
1
Enter the length of the rectangle:
4
Enter the breadth of the rectangle:
2
Area of rectangle: 8.0
Perimeter of rectangle: 12.0
1.Rectangle
2.Circle
3.Exit
Enter tour choice
3
```

Result

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

Experiment No.:15

Aim: Prepare bill using calculate method from interface.

CO3

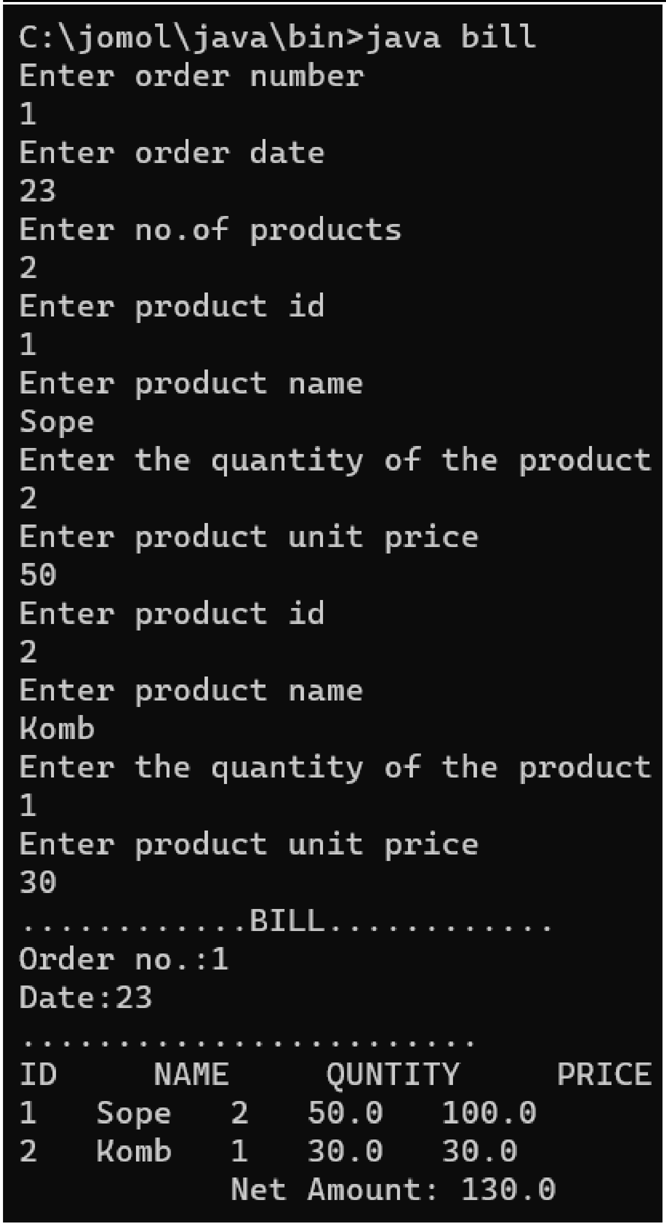
Implement object-oriented concepts like inheritance, overloading and interfaces.

Procedure

```
import java.util.*;
interface calculate
{
public void calc();
}
class bill implements calculate
{

String date,name;
int qu,id;
float uprice,total;
Scanner sc= new Scanner(System.in);
public void get()
{
                System.out.println("Enter product id");
                id=sc.nextInt();
                System.out.println("Enter product name");
                name=sc.next();
                System.out.println("Enter the quantity of the product");
                qu=sc.nextInt();
                System.out.println("Enter product unit price");
                uprice=sc.nextFloat();
}
public void calc()
{
total=qu*uprice;
}
public void disp()
{
System.out.println(id+"   "+name+"   "+qu+"   "+uprice+"   "+total);
}
public static void main(String[] args)
{
        int n,i,o;
        float net=0;
        String date;
        Scanner sc= new Scanner(System.in);
        System.out.println("Enter order number");
        o=sc.nextInt();
        System.out.println("Enter order date");
```

```
            date=sc.next();
            System.out.println("Enter no.of products");
   n=sc.nextInt();
            bill b[]=new bill[n];
            for(i=0;i<n;i++)
            {
                        b[i]=new bill();
                        b[i].get();
                        b[i].calc();
            }
System.out.println("............BILL............");
System.out.println("Order no.:"+o);
System.out.println("Date:"+date);
System.out.println("........................");
System.out.println("ID     NAME     QUNTITY     PRICE");
for(i=0;i<n;i++)
            {
                        b[i].disp();
                        net=net+b[i].total;
            }
            System.out.println("          Net Amount: "+net);
}
}
```

Output Screenshot



```
C:\jomol\java\bin>java bill
Enter order number
1
Enter order date
23
Enter no.of products
2
Enter product id
1
Enter product name
Sope
Enter the quantity of the product
2
Enter product unit price
50
Enter product id
2
Enter product name
Komb
Enter the quantity of the product
1
Enter product unit price
30
............BILL...........
Order no.:1
Date:23
........................
ID     NAME     QUNTITY     PRICE
1   Sope   2   50.0   100.0
2   Komb   1   30.0   30.0
           Net Amount: 130.0
```

Result

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

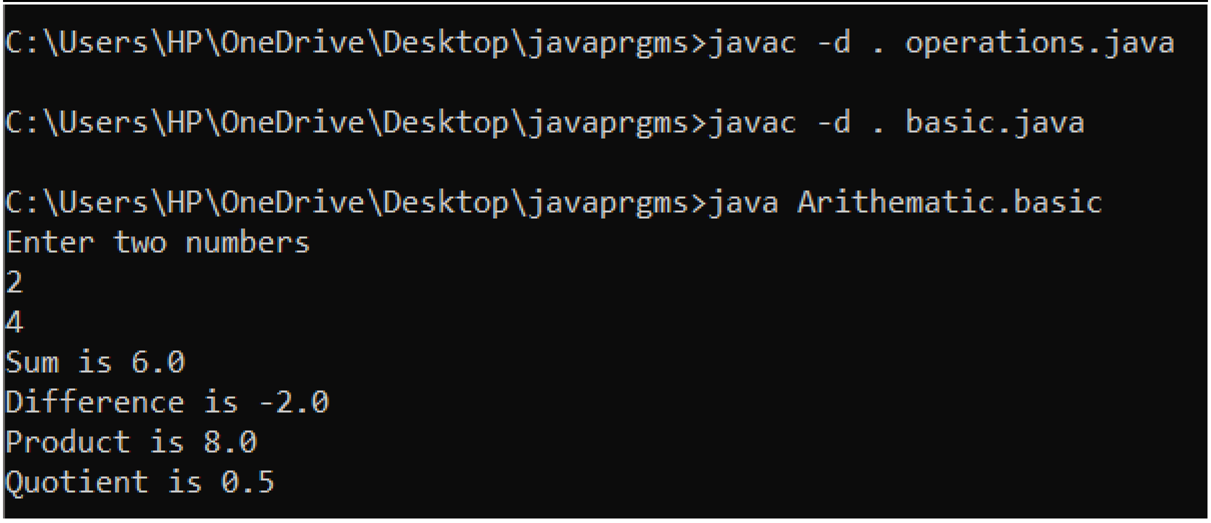
CO4

Implement packages, exception handling, multithreading and generic programming. Use java.util package and Collection framework.

Procedure

```
//shapes.java

package Graphiccs;
import java.util.*;

public class shapes implements Area
{
double lr,lb,ra,th,tb,ta,saa,sa,cr,cc;
public void getRect()
{
Scanner ab= new Scanner(System.in);
System.out.println("Enter the length of the rectangle");
lr=ab.nextInt();
System.out.println("Enter the breadth of the rectangle");
lb=ab.nextInt();

}
public void Rectangle()
{
ra=lr*lb;
System.out.println("Area of Rectangle is "+ra);
}
public void getTri()
{
Scanner cb= new Scanner(System.in);
System.out.println("Enter the height of the Triangle");
th=cb.nextInt();
System.out.println("Enter the base of the Triangle");
tb=cb.nextInt();
}
public void Triangle()
{
ta=0.5*th*tb;
System.out.println("Area of Triangle angle is "+ta);
}
public void getSqr()
{
Scanner sq= new Scanner(System.in);
System.out.println("Enter the Side of the Square");
sa=sq.nextInt();
}
public void Square()
{
saa=sa*sa;
System.out.println("Area of Square is "+saa);
}

public void getCrl()
{
Scanner sc= new Scanner(System.in);
System.out.println("Enter the radius of the Circle");
cc=sc.nextInt();
}

public void Circle()
{
cr=3.14*cc*cc;
System.out.println("Area of Square is "+cr);
}
public static void main(String[] args)
{
shapes o= new shapes();
o.getRect();
o.Rectangle();
o.getTri();
o.Triangle();
o.getSqr();
o.Square();
o.getCrl();
o.Circle();
}
}

//Area.java

package Graphiccs;
interface Area
{
public void Rectangle();
public void Triangle();
public void Square();
public void Circle();
public void getRect();
public void getTri();
public void getSqr();
public void getCrl();
}
```

Output Screenshot



```
C:\Users\HP\OneDrive\Desktop\javaprgms>javac -d . operations.java

C:\Users\HP\OneDrive\Desktop\javaprgms>javac -d . basic.java

C:\Users\HP\OneDrive\Desktop\javaprgms>java Arithematic.basic
Enter two numbers
2
4
Sum is 6.0
Difference is -2.0
Product is 8.0
Quotient is 0.5
```

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

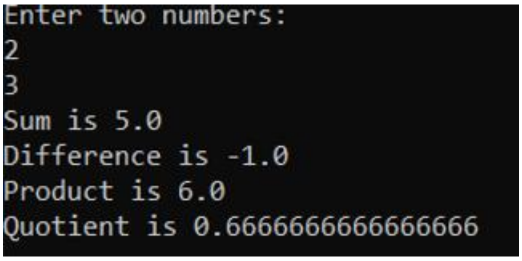
CO4

Implement packages, exception handling, multithreading and generic programming. Use java.util package and Collection framework.

Procedure

```
//basic.java

package Arithmetic;
import java.util.*;
public class basic implements operations
{
double a,b,ad,dif,mult,div;
public void input()
{
Scanner ab=new Scanner(System.in);
System.out.println("Enter two numbers:");
a=ab.nextInt();
b=ab.nextInt();
}
public void add()
{
ad=a+b;
System.out.println("Sum is "+ad);
}

public void substract()
{

dif=a-b;
System.out.println("Difference is "+dif);
}
public void multiply()
{
mult=a*b;
System.out.println("Product is "+mult);
}
public void division()
{
div=a/b;
System.out.println("Quotient is "+div);
}
public static void main(String[] args)
{
basic o=new basic();
o.input();
o.add();
o.substract();
o.multiply();
o.division();
}
}


// operations.java

package Arithmetic;
interface operations
{
public void input();
public void add();
public void substract();
public void multiply();
public void division();

}
```

Output Screenshot



```
Enter two numbers:
2
3
Sum is 5.0
Difference is -1.0
Product is 6.0
Quotient is 0.6666666666666666
```

Result

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.

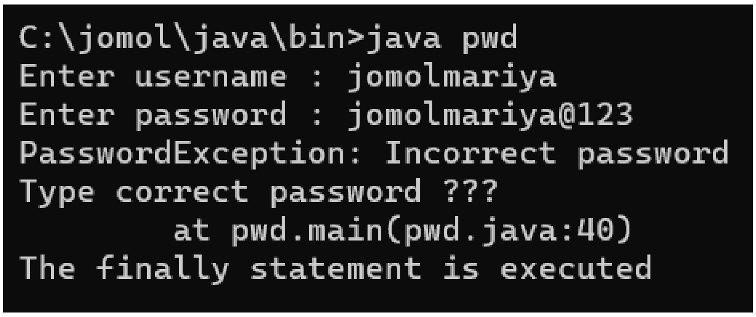
CO4

Implement packages, exception handling, multithreading and generic programming. Use java.util package and Collection framework.

Procedure

```
import java.util.Scanner;
class UsernameException extends Exception
{
public UsernameException(String msg)
{
super(msg);
}
}

class PasswordException extends Exception
{

public PasswordException(String msg)
{
super(msg);
}
}

public class pwd
{

public static void main(String[] args)
{
Scanner s = new Scanner(System.in);
String username, password;

System.out.print("Enter username : ");
username = s.nextLine();

System.out.print("Enter password : ");
password = s.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 Successful !!!");
}
catch (UsernameException u)
{
u.printStackTrace();
}
catch (PasswordException p)
{
p.printStackTrace();
}
finally
{
System.out.println("The finally statement is executed");
}
}
}
```

Output Screenshot



```
C:\jomol\java\bin>java pwd
Enter username : jomolmariya
Enter password : jomolmariya@123
PasswordException: Incorrect password
Type correct password ???
        at pwd.main(pwd.java:40)
The finally statement is executed
```

Result

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

Experiment No.:19

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

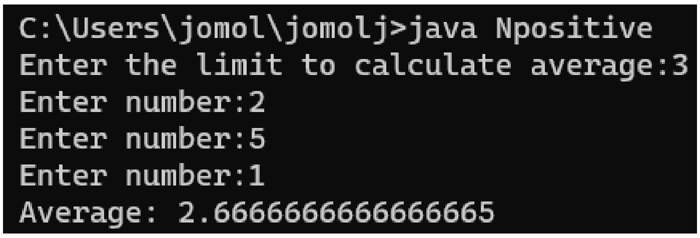
CO4

Implement packages, exception handling, multithreading and generic programming. Use java.util package and Collection framework.

Procedure

```
import java.util.Scanner;
import java.util.InputMismatchException;
public class Npositive
{
public static void main(String args[])
{
double total = 0, N, userInput;
Scanner input = new Scanner(System.in);
while (true)
{
System.out.print("Enter the limit to calculate average:");
userInput = input.nextDouble();
if (userInput > 0)
{
N = userInput;
break;
}
else

System.out.println("N must be positive.");

}
for (int i = 0; i < N; i++)
{
while (true)
{
System.out.print("Enter number:");
try
{
userInput = input.nextDouble();
total += userInput;
break;

}
catch (InputMismatchException e)
{
input.nextLine();
System.out.println("Input must bea number. Try again");
}
}
}
System.out.println("Average: "+ total / N);
}
}
```

Output Screenshot



```
C:\Users\jomol\jomolj>java Npositive
Enter the limit to calculate average:3
Enter number:2
Enter number:5
Enter number:1
Average: 2.6666666666666665
```

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

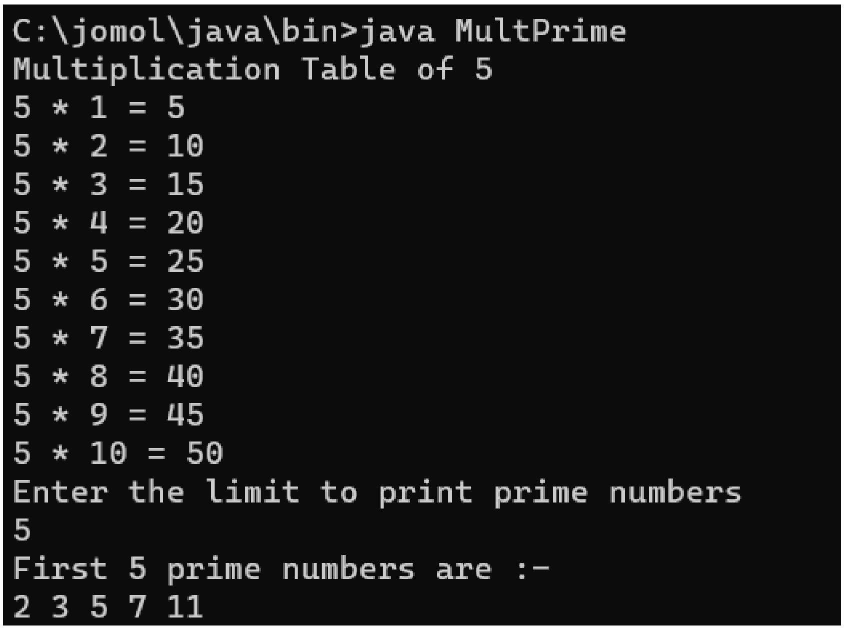
Implement packages, exception handling, multithreading and generic programming. Use java.util package and Collection framework.

Procedure

```
import java.util.*;
class ThreadA extends Thread
{
public void run( )
{
int n = 5;
for (int i = 1; i <= 10; ++i)
System.out.println(n + " * " + i +
" = " + n * i);
}
}

class ThreadB extends Thread
{
public void run( )
{
Scanner sc = new Scanner(System.in);
int i,n,p,count,flag;

System.out.println("Enter the limit to print prime numbers");
n=sc.nextInt();
System.out.println("First "+n+" prime numbers are :-");

p=2;
i=1;
while(i<=n)
{
flag=1;

for(count=2;count<=p-1;count++)
{
if(p%count==0)
{
flag=0;
break;
}
}
if(flag==1)
{
```

```
System.out.print(p+" ") ;
i++;
}
p++;
}
}
}

public class MultPrime
{
public static void main(String args[]) {
ThreadA a = new ThreadA();
ThreadB b = new ThreadB();
a.start();
b.start();
System.out.println("Multiplication Table of 5");
}
}
```

Output Screenshot



```
C:\jomol\java\bin>java MultPrime
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 to print prime numbers
5
First 5 prime numbers are :-
2 3 5 7 11
```

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)

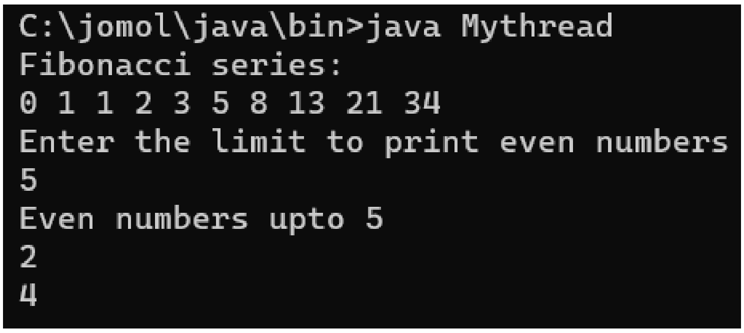
CO4

Implement packages, exception handling, multithreading and generic programming. Use java.util package and Collection framework.

Procedure

```
import java.util.*;
public class Mythread {

public static void main(String[] args) {
Runnable r = new Runnable1();
Thread t = new Thread(r);
t.start();
Runnable r2 = new Runnable2();
Thread t2 = new Thread(r2);
t2.start();
}
}

class Runnable2 implements Runnable{
public void run()
{
        Scanner sc=new Scanner(System.in);
        System.out.println("\n"+"Enter the limit to print even numbers");
        int n=sc.nextInt();
        System.out.println("Even numbers upto "+ n);
for(int i=1;i<n;i++)
{
if(i%2 == 0)
System.out.println(i);
}
}
}

class Runnable1 implements Runnable{
public void run()
{
int n1=0,n2=1,n3,i,count=10;
System.out.println("Fibonacci series: ");
System.out.print(n1+" "+n2);//printing 0 and 1

for(i=2;i<count;++i)//loop starts from 2 because 0 and 1 are already printed
{
n3=n1+n2;
System.out.print(" "+n3);
n1=n2;
n2=n3;
}
}
}
```

Output Screenshot



```
C:\jomol\java\bin>java Mythread
Fibonacci series:
0 1 1 2 3 5 8 13 21 34
Enter the limit to print even numbers
5
Even numbers upto 5
2
4
```

Result

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

Experiment No.:22

Aim: To create a generic stack and do the Push and Pop operations.

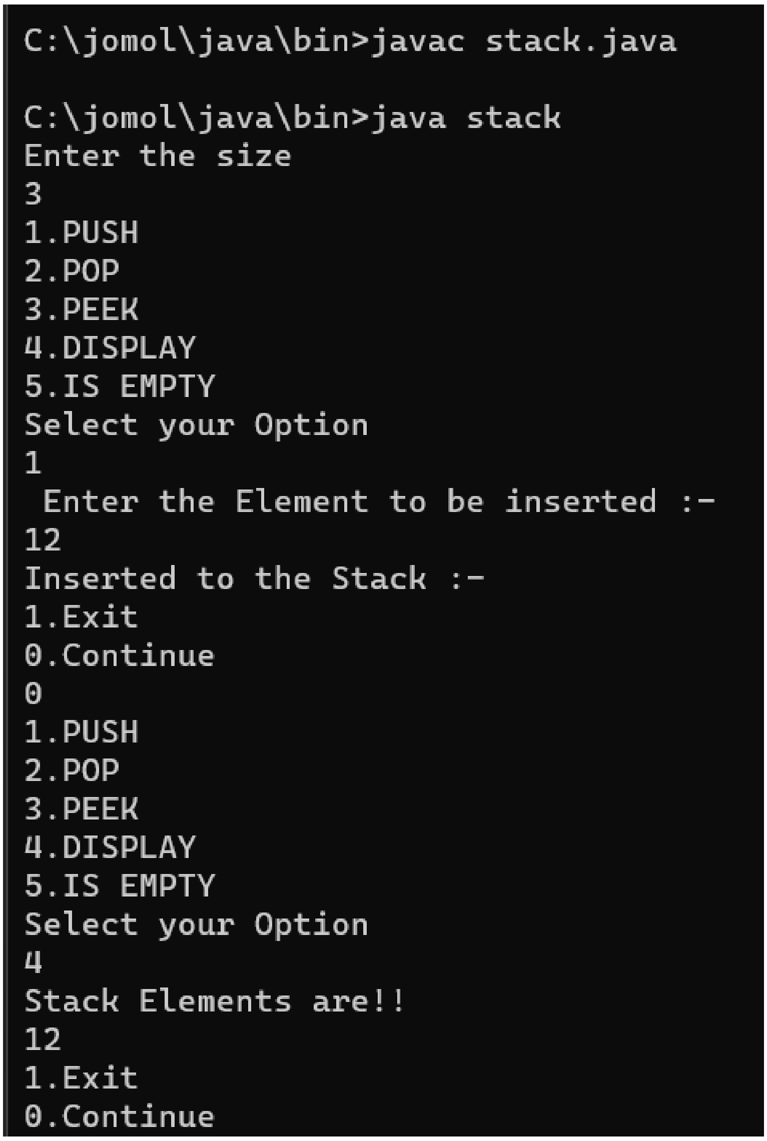
CO4

Implement packages, exception handling, multithreading and generic programming. Use java.util package and Collection framework.

Procedure

```
import java.util.*;
class stack
{
        public static void main(String args[]){
                Scanner obj =new Scanner(System.in);
                int ch;
                int top=-1;
                int n;
                System.out.println("Enter the size");
                n=obj.nextInt();
                int s[]=new int[n];

                do{
                        System.out.println("1.PUSH ");
                System.out.println("2.POP ");
                System.out.println("3.PEEK ");
                System.out.println("4.DISPLAY ");
                System.out.println("5.IS EMPTY ");
                System.out.println("Select your Option  ");
                ch=obj.nextInt();
                switch(ch){
                        case 1:
                                        if(top!=n)
                                        {
                                                System.out.println(" Enter the Element to be
inserted :- ");
                                                s[++top]=obj.nextInt();
                                                System.out.println("Inserted to the Stack :- ");
                                        }
                                break;
                        case 2:
                                if(top==-1){
                                        System.out.println("Stack Under Flow");
                                }
                                else
                                {
                                        top=top-1;
                                        System.out.println("One Element is deleted");
                                }
                                break;
                        case 3:
                                System.out.println("TOP Of Stack is :" + s[top] );
                                break;
                        case 5:
                                if(top==-1){
                                        System.out.println("Stack Is Empty!!");
                                }
                                else{
                                        System.out.println("Stack is not Empty !!");
                                }
                                break;
                        case 4 :
                                if(top==-1){
                                        System.out.println("Stack Is Empty!!");
                                }
                                else{
                                        System.out.println("Stack Elements are!!");
                                        for (int i=0;i<=top;i++){
                                                System.out.println(s[i]);
                                                }

                                }
                                break;
        }
                System.out.println("1.Exit ");
                System.out.println("0.Continue");
                ch=obj.nextInt();
                } while(ch==0);
}
}
```

Output Screenshot



```
C:\jomol\java\bin>javac stack.java

C:\jomol\java\bin>java stack
Enter the size
3
1.PUSH
2.POP
3.PEEK
4.DISPLAY
5.IS EMPTY
Select your Option
1
 Enter the Element to be inserted :-
12
Inserted to the Stack :-
1.Exit
0.Continue
0
1.PUSH
2.POP
3.PEEK
4.DISPLAY
5.IS EMPTY
Select your Option
4
Stack Elements are!!
12
1.Exit
0.Continue
```

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.

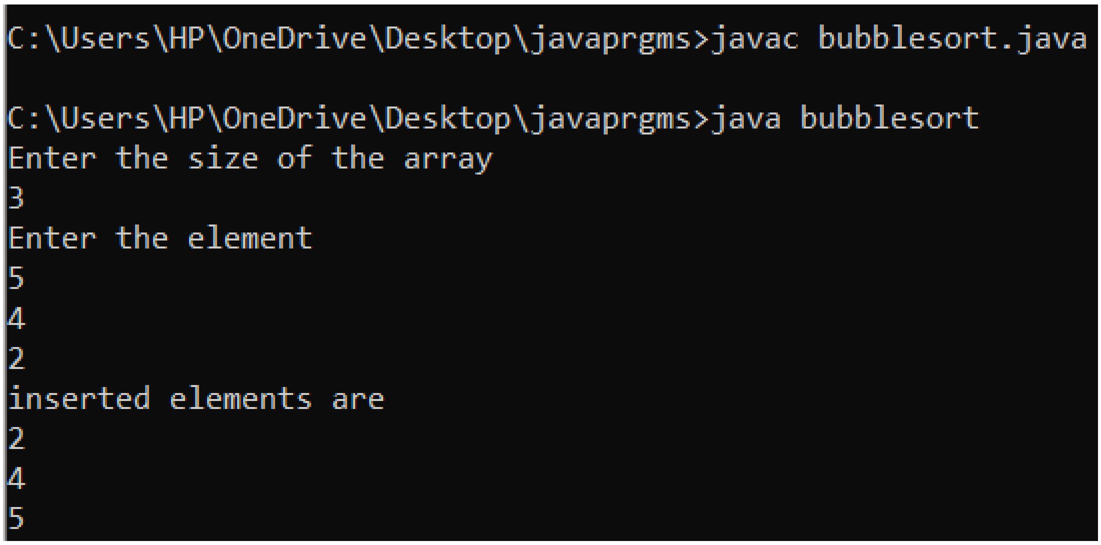
CO4

Implement packages, exception handling, multithreading and generic programming. Use java.util package and Collection framework.

Procedure

```
import java.util.*;
public class BubbleSort
{
        public static void main(String[] args)
   {
                //BubbleSort b=new BubbleSort();
                int n,temp;
                Scanner sc=new Scanner(System.in);
                System.out.println("Enter the size");
                n=sc.nextInt();
                int bub[]=new int[n];
                System.out.println("Enter the elements");
                for(int i=0;i<n;i++)
                {
                        bub[i]=sc.nextInt();
                }


                        for(int i=0;i<n;i++)
                        {
                                for(int j=i+1;j<n;j++)
                                {
                                        if(bub[i]>bub[j])
                                        {
                                                temp=bub[i];
                                                bub[i]=bub[j];
                                                bub[j]=temp;
                                        }
                                }
                        }
                        System.out.println("Bubble sorted form");
                        for(int i=0;i<n;i++)
                                {
                                        System.out.println(bub[i]);
                                }

}
}
```

Output Screenshot



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

C:\Users\HP\OneDrive\Desktop\javaprgms>java bubblesort
Enter the size of the array
3
Enter the element
5
4
2
inserted elements are
2
4
5
```

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. Use java.util package and Collection framework.

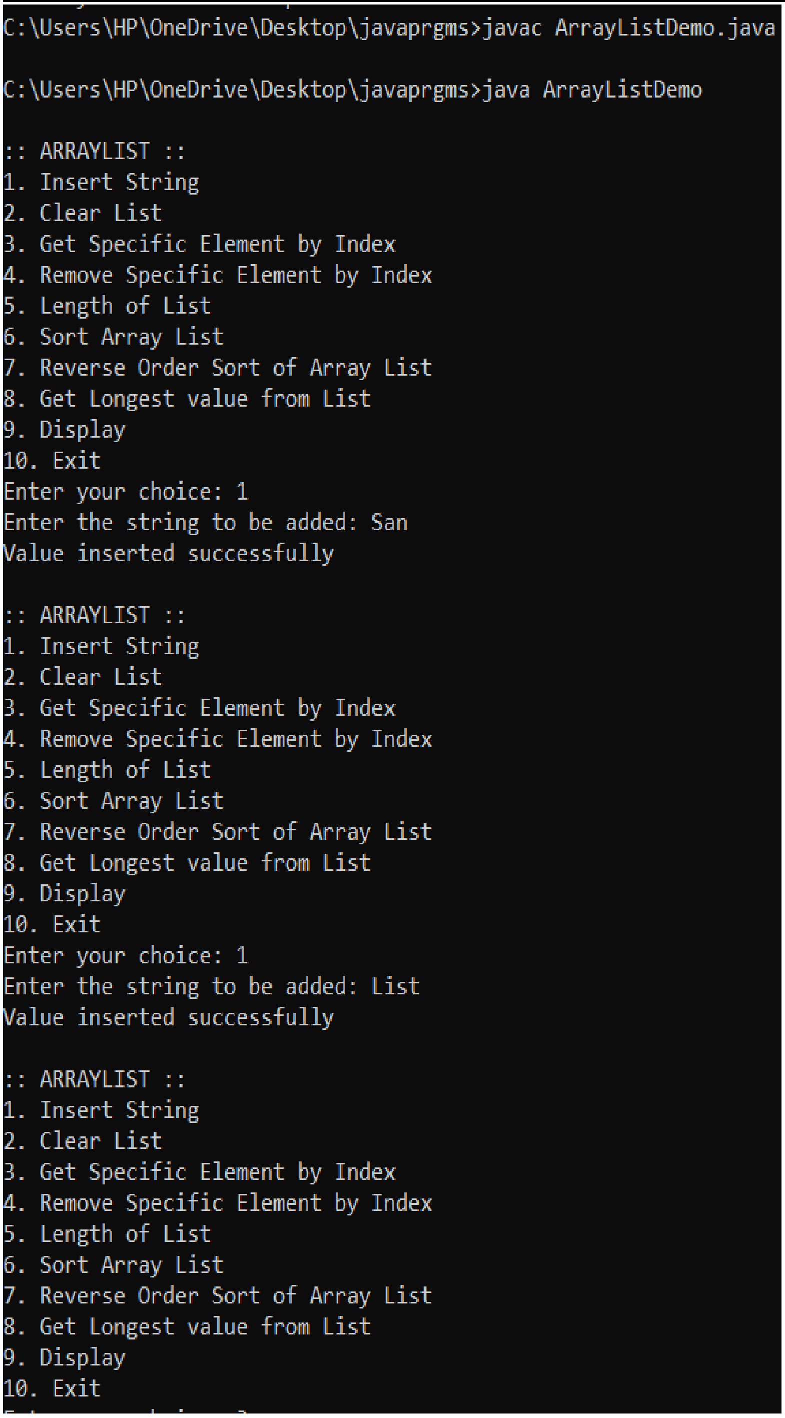
Procedure

```
import java.util.ArrayList;
import java.util.Comparator;
import java.util.Scanner;
import java.util.Collections;
public class ArrayListDemo
{
        public static void insert(ArrayList<String> list)
        {
                Scanner Snr = new Scanner(System.in);
                System.out.print("Enter the string to be added: ");
                String val = Snr.next();
                list.add(val);
                System.out.println("Value inserted successfully");
        }
        public static void deleteAll(ArrayList<String> list)
        {
                list.clear();
                System.out.println("ArrayList successfully cleared");
        }
        public static void find(ArrayList<String> list)
        {
                Scanner Snr = new Scanner(System.in);
                System.out.print("Enter the index to get: ");
                int val = Snr.nextInt();
                System.out.println(list.get(val));
        }
        public static void delete(ArrayList<String> list)
        {
                Scanner Snr = new Scanner(System.in);
                System.out.print("Enter the index to delete: ");
                int val = Snr.nextInt();
                list.remove(val);
                System.out.println("Value successfully removed");
        }
        public static void len(ArrayList<String> list)
        {
```

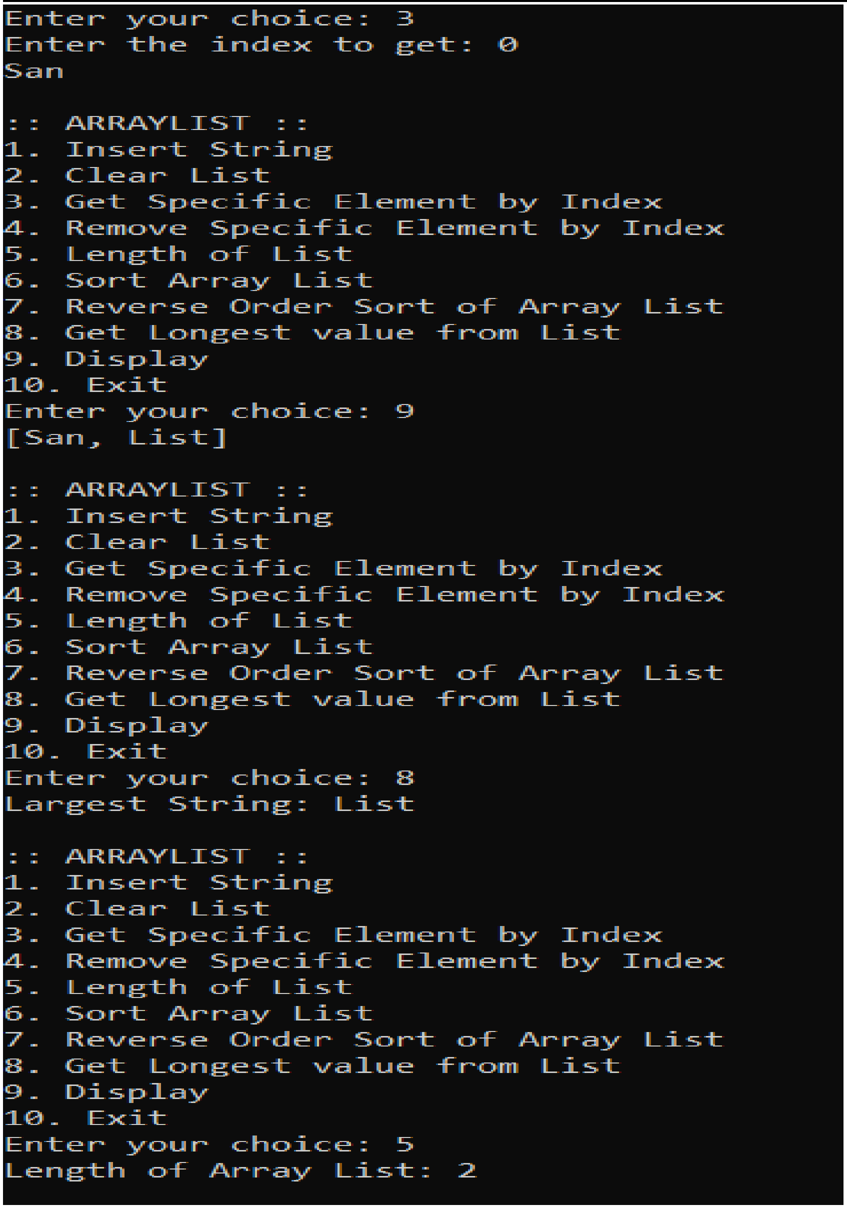
```
		System.out.println("Length of Array List: " + list.size());
	}
	public static void sortlist(ArrayList<String> list)
	{
		list.sort(Comparator.naturalOrder());
		System.out.println("Sorted Array List: " + list);
	}
	public static void sortRev(ArrayList<String> list)
	{
		list.sort(Comparator.reverseOrder());
		System.out.println("Reverse Order Sorted Array List: " + list);
	}
	public static void max(ArrayList<String> list)
	{
		String large = list.get(0);
		int len = large.length();
		for(int i=1; i<list.size(); i++)
		{
			String cand = list.get(i);
			if(len < cand.length())
			{
				large = list.get(i);
				len = large.length();
			}
		}
		System.out.println("Largest String: " + large);
	}
	public static void display(ArrayList<String> list)
	{
		System.out.println(list);
	}
	public static void main(String[] args)
	{
		ArrayList<String> list = new ArrayList<String>();
		Scanner Snr = new Scanner(System.in);
		int choice=0;
		while(choice!=10)
		{
			System.out.println("");
			System.out.println(":: ARRAYLIST ::");
			System.out.println("1. Insert String");
			System.out.println("2. Clear List");
			System.out.println("3. Get Specific Element by Index");
			System.out.println("4. Remove Specific Element by Index");
			System.out.println("5. Length of List");
			System.out.println("6. Sort Array List");
			System.out.println("7. Reverse Order Sort of Array List");
			System.out.println("8. Get Longest value from List");
```

```
                        System.out.println("9. Display");
                        System.out.println("10. Exit");
                        System.out.print("Enter your choice: ");
                        choice= Snr.nextInt();
                        switch(choice)
                        {
                                case 1:
                                        insert(list);
                                        break;
                                case 2:
                                        deleteAll(list);
                                        break;
                                case 3:
                                        find(list);
                                        break;
                                case 4:
                                        delete(list);
                                        break;
                                case 5:
                                        len(list);
                                        break;
                                case 6:
                                        sortlist(list);
                                        break;
                                case 7:
                                        sortRev(list);
                                        break;
                                case 8:
                                        max(list);
                                        break;
                                case 9:
                                        display(list);
                                        break;
                                case 10:
                                        System.exit(0);
                                        break;
                                default:
                                        System.out.println("Enter a valid choice!");
                }
        }
}
}
```

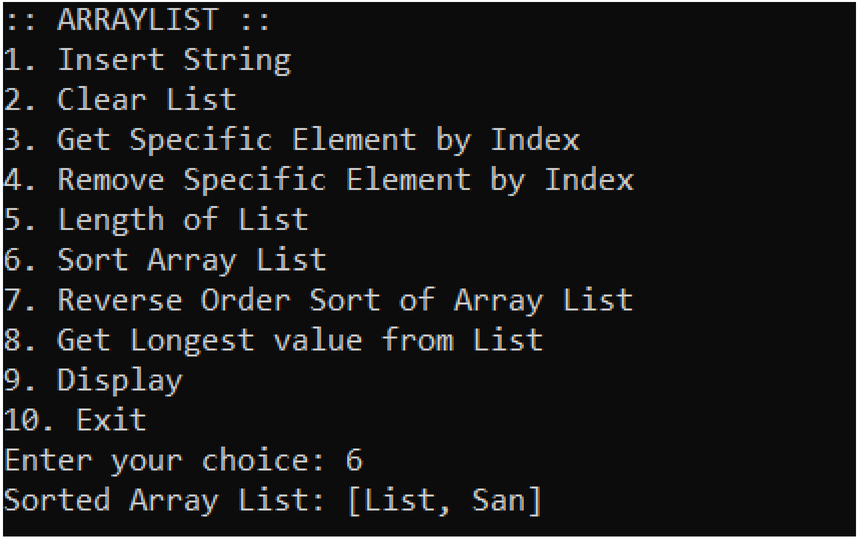
Output Screenshot:



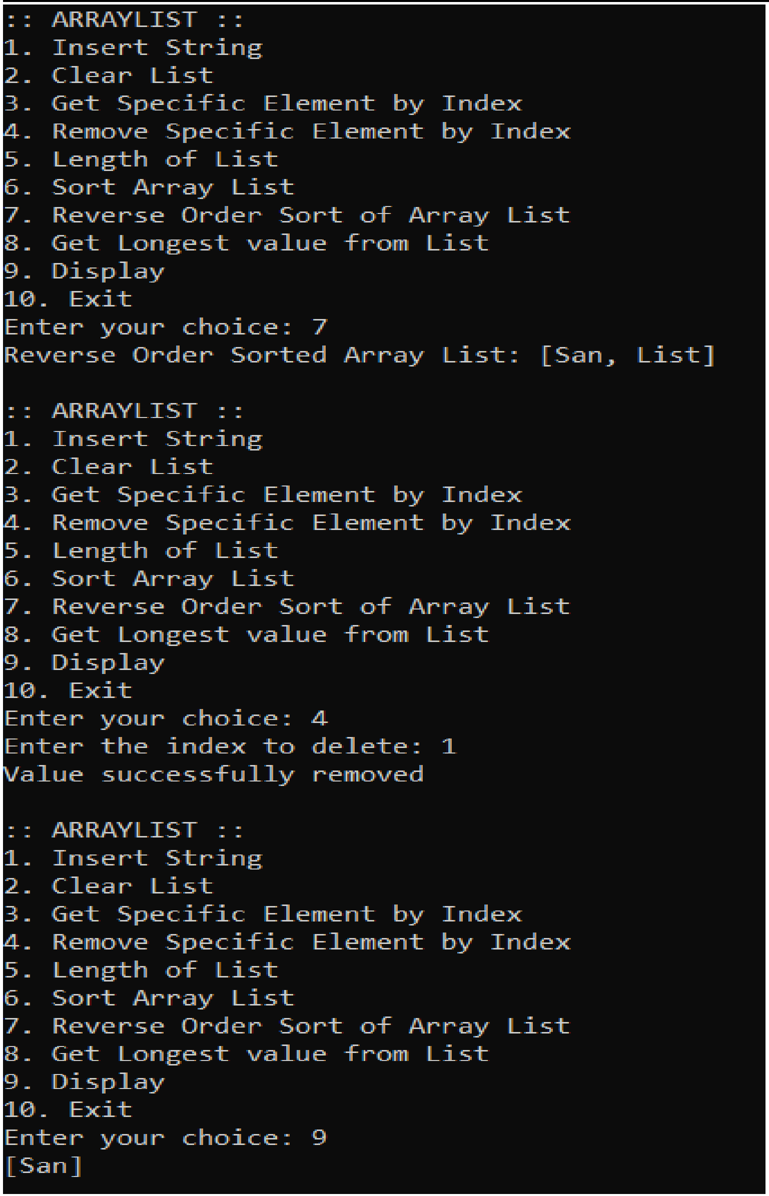
```
C:\Users\HP\OneDrive\Desktop\javaprgms>javac ArrayListDemo.java

C:\Users\HP\OneDrive\Desktop\javaprgms>java ArrayListDemo

:: ARRAYLIST ::
1. Insert String
2. Clear List
3. Get Specific Element by Index
4. Remove Specific Element by Index
5. Length of List
6. Sort Array List
7. Reverse Order Sort of Array List
8. Get Longest value from List
9. Display
10. Exit
Enter your choice: 1
Enter the string to be added: San
Value inserted successfully

:: ARRAYLIST ::
1. Insert String
2. Clear List
3. Get Specific Element by Index
4. Remove Specific Element by Index
5. Length of List
6. Sort Array List
7. Reverse Order Sort of Array List
8. Get Longest value from List
9. Display
10. Exit
Enter your choice: 1
Enter the string to be added: List
Value inserted successfully

:: ARRAYLIST ::
1. Insert String
2. Clear List
3. Get Specific Element by Index
4. Remove Specific Element by Index
5. Length of List
6. Sort Array List
7. Reverse Order Sort of Array List
8. Get Longest value from List
9. Display
10. Exit
```



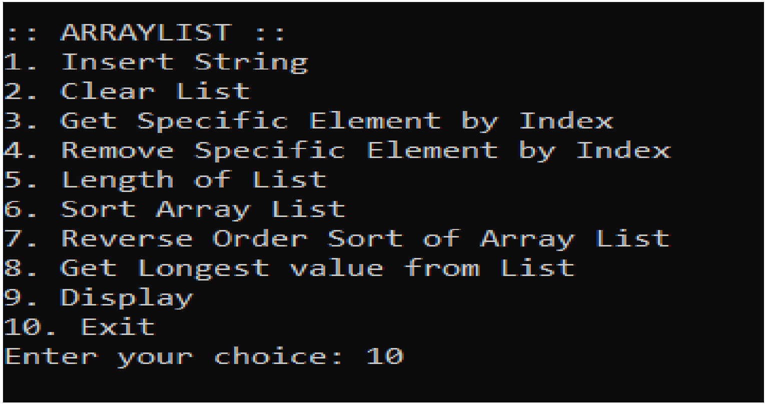
```
Enter your choice: 3
Enter the index to get: 0
San

:: ARRAYLIST ::
1. Insert String
2. Clear List
3. Get Specific Element by Index
4. Remove Specific Element by Index
5. Length of List
6. Sort Array List
7. Reverse Order Sort of Array List
8. Get Longest value from List
9. Display
10. Exit
Enter your choice: 9
[San, List]

:: ARRAYLIST ::
1. Insert String
2. Clear List
3. Get Specific Element by Index
4. Remove Specific Element by Index
5. Length of List
6. Sort Array List
7. Reverse Order Sort of Array List
8. Get Longest value from List
9. Display
10. Exit
Enter your choice: 8
Largest String: List

:: ARRAYLIST ::
1. Insert String
2. Clear List
3. Get Specific Element by Index
4. Remove Specific Element by Index
5. Length of List
6. Sort Array List
7. Reverse Order Sort of Array List
8. Get Longest value from List
9. Display
10. Exit
Enter your choice: 5
Length of Array List: 2
```



```
:: ARRAYLIST ::
1. Insert String
2. Clear List
3. Get Specific Element by Index
4. Remove Specific Element by Index
5. Length of List
6. Sort Array List
7. Reverse Order Sort of Array List
8. Get Longest value from List
9. Display
10. Exit
Enter your choice: 6
Sorted Array List: [List, San]
```



```
:: ARRAYLIST ::
1. Insert String
2. Clear List
3. Get Specific Element by Index
4. Remove Specific Element by Index
5. Length of List
6. Sort Array List
7. Reverse Order Sort of Array List
8. Get Longest value from List
9. Display
10. Exit
Enter your choice: 7
Reverse Order Sorted Array List: [San, List]

:: ARRAYLIST ::
1. Insert String
2. Clear List
3. Get Specific Element by Index
4. Remove Specific Element by Index
5. Length of List
6. Sort Array List
7. Reverse Order Sort of Array List
8. Get Longest value from List
9. Display
10. Exit
Enter your choice: 4
Enter the index to delete: 1
Value successfully removed

:: ARRAYLIST ::
1. Insert String
2. Clear List
3. Get Specific Element by Index
4. Remove Specific Element by Index
5. Length of List
6. Sort Array List
7. Reverse Order Sort of Array List
8. Get Longest value from List
9. Display
10. Exit
Enter your choice: 9
[San]
```



```
:: ARRAYLIST ::
1. Insert String
2. Clear List
3. Get Specific Element by Index
4. Remove Specific Element by Index
5. Length of List
6. Sort Array List
7. Reverse Order Sort of Array List
8. Get Longest value from List
9. Display
10. Exit
Enter your choice: 10
```

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

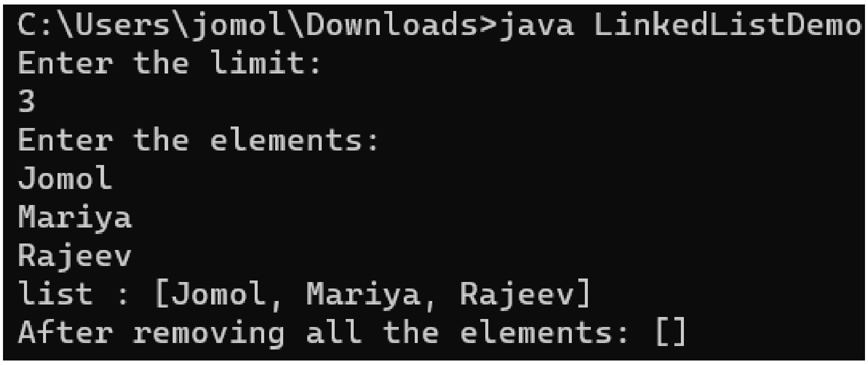
CO4

Implement packages, exception handling, multithreading and generic programming. Use java.util package and Collection framework.

Procedure

```
import java.util.*;
public class LinkedListDemo{
   public static void main(String[] args)
   {
   Scanner sc = new Scanner(System.in);
   LinkedList<String> list=new LinkedList<String>();
   System.out.println("Enter the limit:");
   int limit = sc.nextInt();
   System.out.println("Enter the elements: ");
   for(int i=0;i<limit;i++)
      {
         list.add(sc.next());
      }
   System.out.println("list : "+list);
   list.clear();
   System.out.println("After removing all the elements: "+list);


   }
}
```

Output Screenshot



```
C:\Users\jomol\Downloads>java LinkedListDemo
Enter the limit:
3
Enter the elements:
Jomol
Mariya
Rajeev
list : [Jomol, Mariya, Rajeev]
After removing all the elements: []
```

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

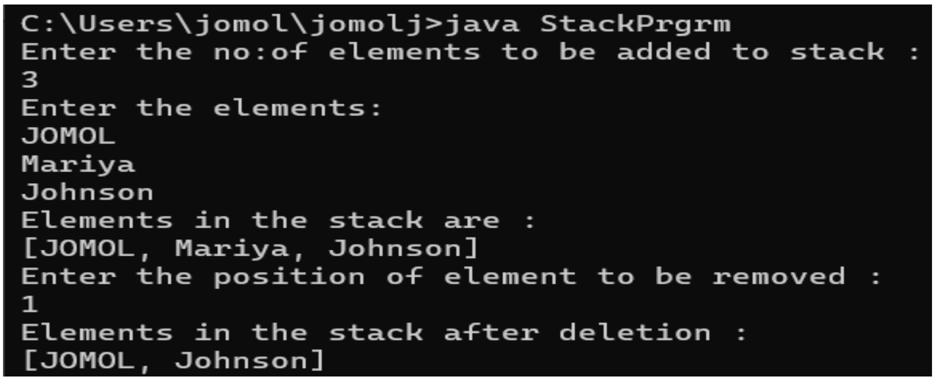
CO4

Implement packages, exception handling, multithreading and generic programming. Use java.util package and Collection framework.

Procedure

```
import java.util.*;
public class StackPrgrm
{
   public static void main(String args[])
   {
      Scanner sc = new Scanner(System.in);
      Stack<String> st = new Stack<String>();
      System.out.println("Enter the no:of elements to be added to stack : ");
      int n = sc.nextInt();
      System.out.println("Enter the elements: ");
      for(int i=0;i<n;i++)
      {
         st.add(sc.next());
      }
      System.out.println("Elements in the stack are : ");
      System.out.println(st);
      System.out.println("Enter the position of element to be removed : ");
      int k = sc.nextInt();
      String rem = st.remove(k);
      System.out.println("Elements in the stack after deletion : ");
      System.out.println(st);
   }
}
```

Output Screenshot



```
C:\Users\jomol\jomolj>java StackPrgrm
Enter the no:of elements to be added to stack :
3
Enter the elements:
JOMOL
Mariya
Johnson
Elements in the stack are :
[JOMOL, Mariya, Johnson]
Enter the position of element to be removed :
1
Elements in the stack after deletion :
[JOMOL, Johnson]
```

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

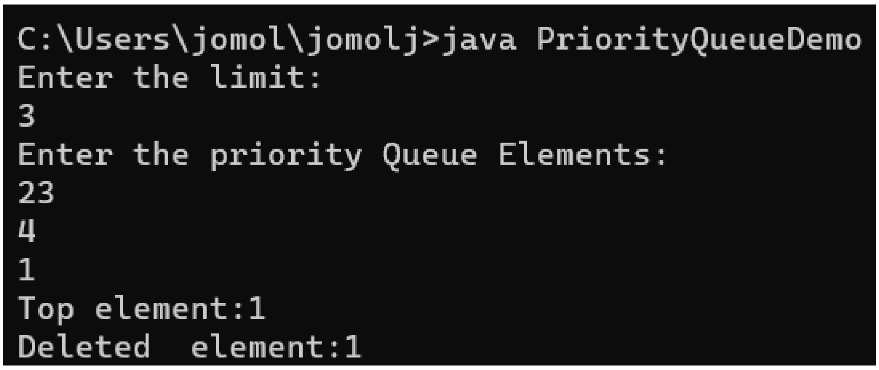
CO4

Implement packages, exception handling, multithreading and generic programming. Use java.util package and Collection framework.

Procedure

```
import java.util.*;
class PriorityQueueDemo{
   public static void main(String[] args)
   {
      Scanner sc =new Scanner(System.in);
      PriorityQueue<Integer> pq = new PriorityQueue<Integer>();
      System.out.println("Enter the limit:");
      int n = sc.nextInt();
      System.out.println("Enter the priority Queue Elements:");
      for(int i=0;i<n;i++)
      {
         pq.add(sc.nextInt());
      }
      System.out.println("Top element:" +pq.peek());
      System.out.println("Deleted  element:" +pq.poll());
   }
}
```

Output Screenshot



```
C:\Users\jomol\jomolj>java PriorityQueueDemo
Enter the limit:
3
Enter the priority Queue Elements:
23
4
1
Top element:1
Deleted  element:1
```

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

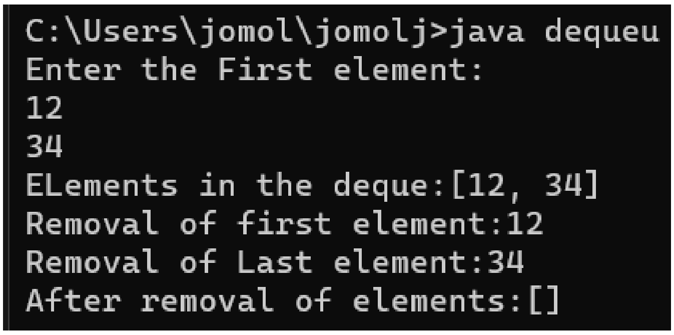
CO4

Implement packages, exception handling, multithreading and generic programming. Use java.util package and Collection framework.

Procedure

```
import java.util.*;
public class dequeu
{
   public static void main(String args[])
   {
      Deque<Integer> obj=new ArrayDeque<>();
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter the First element:");
      int f=sc.nextInt();
      obj.addFirst(f);
      int s=sc.nextInt();
      obj.addLast(s);
      System.out.println("ELements in the deque:"+obj);
      System.out.println("Removal of first element:"+obj.removeFirst());
      System.out.println("Removal of Last element:"+obj.removeLast());
      System.out.println("After removal of elements:"+obj);
   }
}
```

Output Screenshot



```
C:\Users\jomol\jomolj>java dequeu
Enter the First element:
12
34
ELements in the deque:[12, 34]
Removal of first element:12
Removal of Last element:34
After removal of elements:[]
```

Experiment No.: 29

Aim:

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

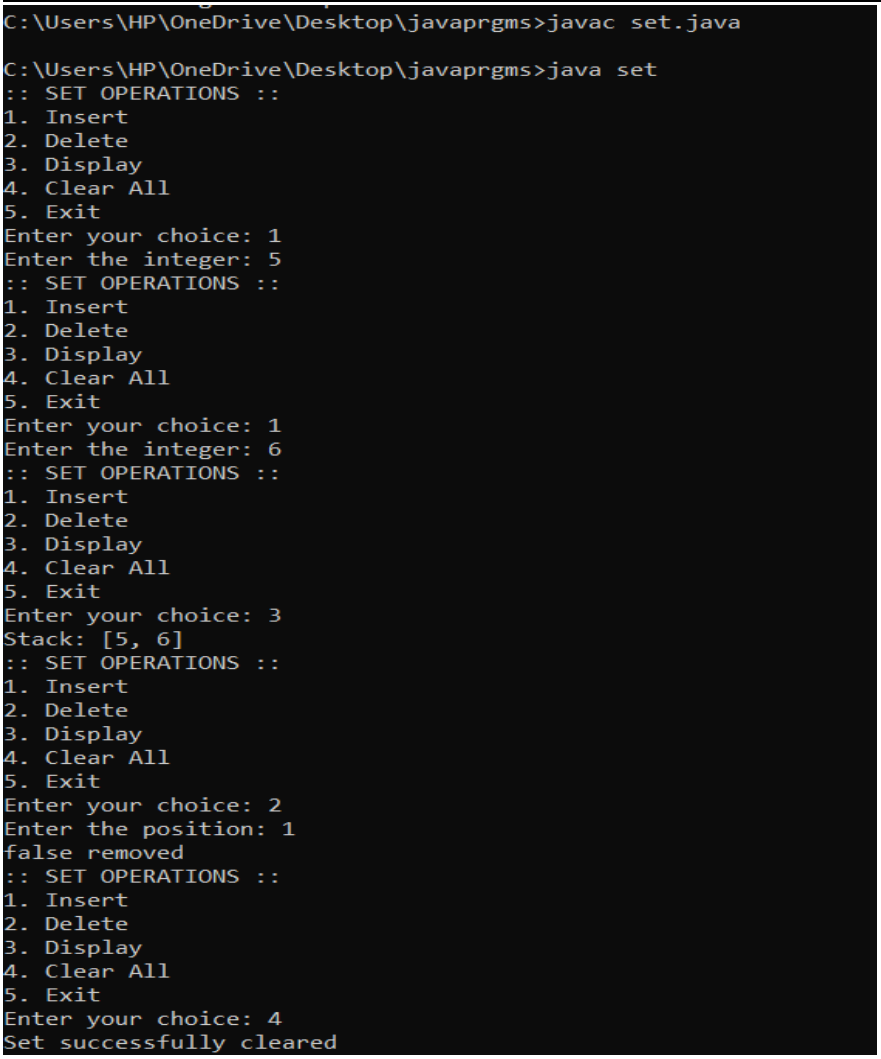
CO4:

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

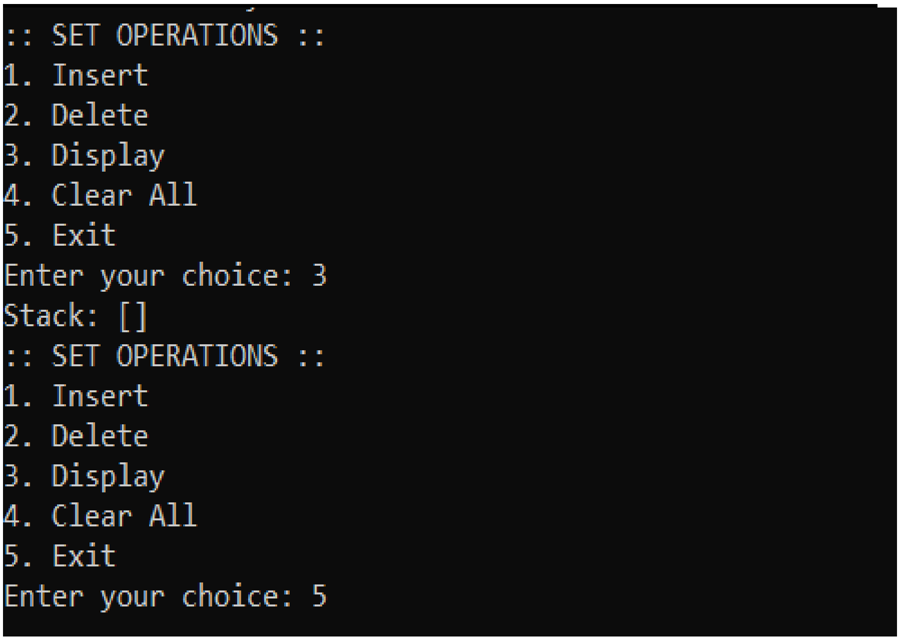
Procedure:

```
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

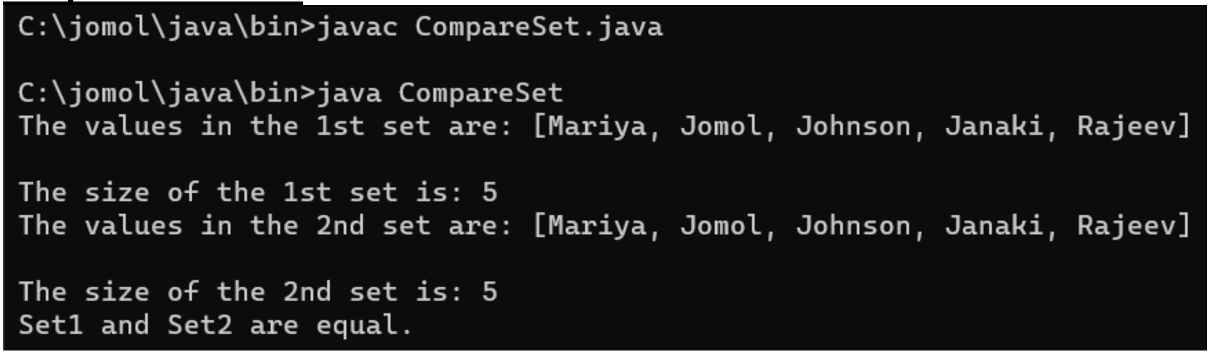
Aim: Write a Java program to compare two hash set.

CO4: Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

Procedure

```
import java.util.*;
public class CompareSet
{
   public static void main(String[] argv)
   {
      Set<String> set1 = new HashSet<String>();
      set1.add("Jomol");
      set1.add("Mariya");
      set1.add("Johnson");
      set1.add("Rajeev");
      set1.add("Janaki");
      System.out.println("The values in the 1st set are: "+ set1);
      System.out.println();
      System.out.println("The size of the 1st set is: "+ set1.size());
      Set<String> set2 = new HashSet<String>();
      set2.add("Jomol");
      set2.add("Rajeev");
      set2.add("Mariya");
      set2.add("Johnson");
      set2.add("Janaki");
      System.out.println("The values in the 2nd set are: "+ set2);
      System.out.println();
      System.out.println("The size of the 2nd set is: "+ set2.size());
      boolean result = set1.equals(set2);
      if(result) {
         System.out.println("Set1 and Set2 are equal.");
      }else {
         System.out.println("Set1 and Set2 are not equal.");
     }  } }
```

Output Screenshot



```
C:\jomol\java\bin>javac CompareSet.java

C:\jomol\java\bin>java CompareSet
The values in the 1st set are: [Mariya, Jomol, Johnson, Janaki, Rajeev]

The size of the 1st set is: 5
The values in the 2nd set are: [Mariya, Jomol, Johnson, Janaki, Rajeev]

The size of the 2nd set is: 5
Set1 and Set2 are equal.
```

Result

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

Experiment No.:31

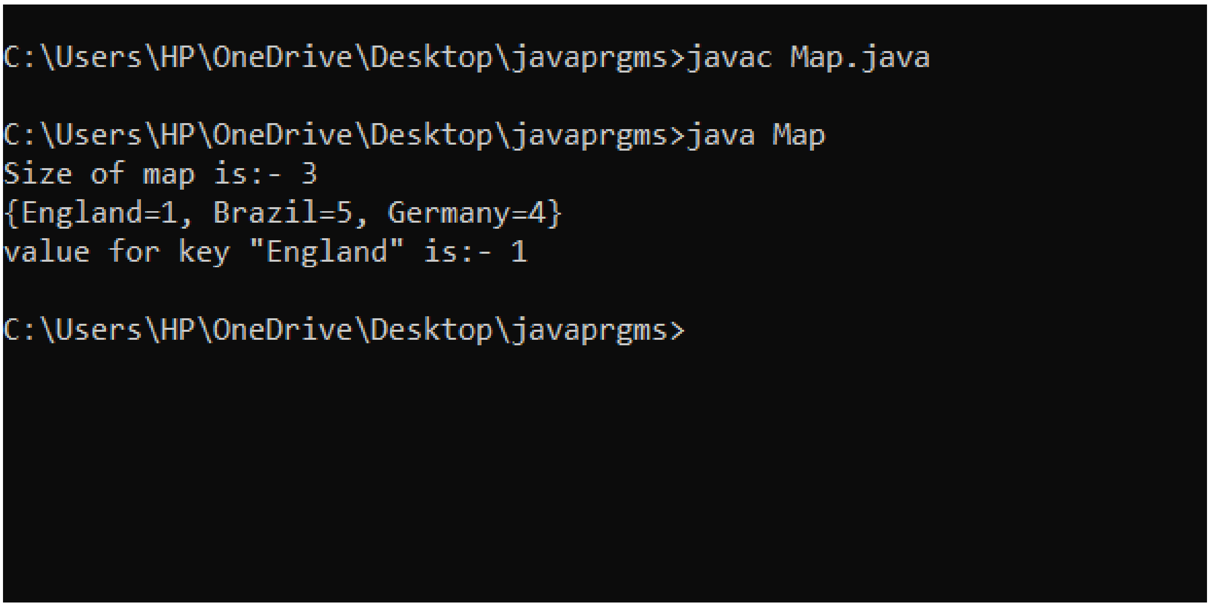
Aim: Program to demonstrate the working of Map interface by adding, changing and removing elements.

CO4: Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

Procedure

```
import java.util.*;
class map {
        public static void main(String args[])
        {
                Map<String, Integer> hm= new HashMap<String, Integer>();
                hm.put("a",20);
                hm.put("b",10);
                hm.put("c", 40);
                hm.put("d", 50);
                        System.out.println("map content:"+hm);
                        hm.put("a",40);
                   System.out.println("map content after changing value:"+hm);
                        hm.remove("c",50);
                        System.out.println("map content after removing value:"+hm);
        }
}
```

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.

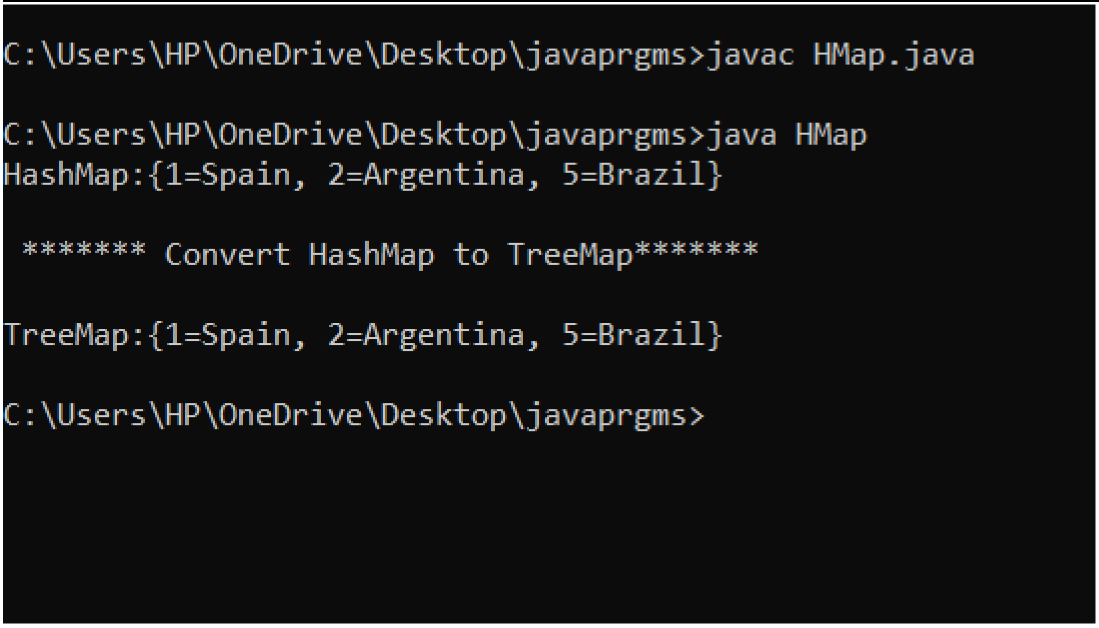
CO4:

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)
{
// TODO Auto-generated method stub
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:

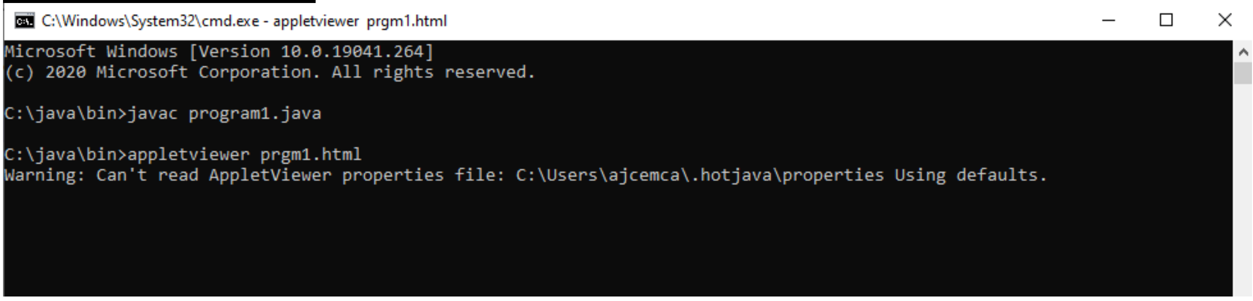
program1.java

```
import java.applet.Applet;
import java.awt.Graphics;
public class program1 extends Applet
{
public void paint(Graphics g)
{
g.drawString("Welcome",50,50);
g.drawLine(20,30,20,300);
g.drawRect(70,100,30,30);
}
}
```

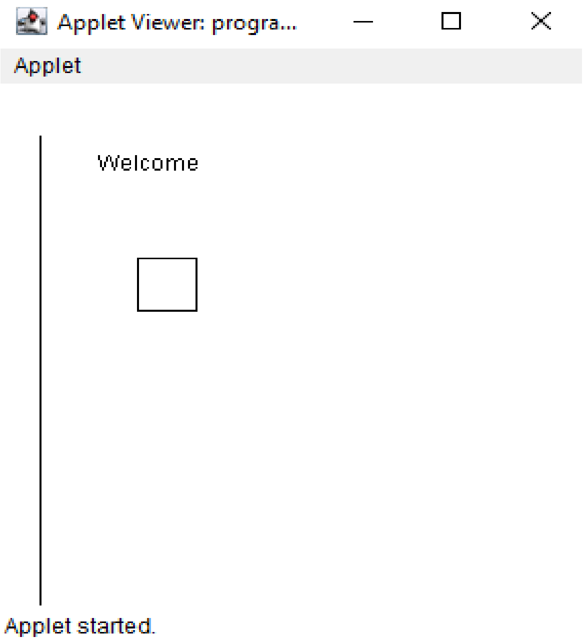
prgm1.html

```
<html>
<body>
<applet code="program1.class" width="300" height="300">
</applet>
</body>
</html>
```

Output Screenshot:



```
Microsoft Windows [Version 10.0.19041.264]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\java\bin>javac program1.java

C:\java\bin>appletviewer prgm1.html
Warning: Can't read AppletViewer properties file: C:\Users\ajcemca\.hotjava\properties Using defaults.
```



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.

CO5:

Design applications using files and network concepts

Procedure:

maxThree.java

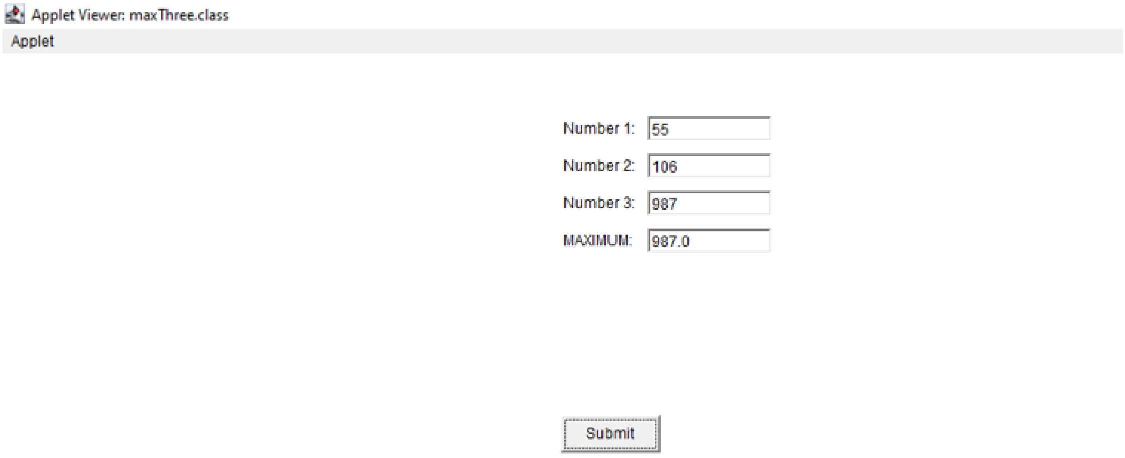
```
import java.awt.*;
import java.awt.Graphics;
import java.applet.*;
import java.awt.event.*;
public class maxThree extends Applet implements ActionListener
{
        Label l1,l2,l3,l4;
        TextField tf1,tf2,tf3,tf4;
        Button btn;
        public void init()
        {
        l1= new Label("Number 1:");
        tf1=new TextField();
        l2=new Label("Number 2:");
        tf2=new TextField();
        l3=new Label("Number 3:");
        tf3=new TextField();
        l4 = new Label("MAXIMUM:");
        tf4= new TextField();
        btn= new Button("Submit");
        setLayout(null);
      l1.setBounds(450,50,70,20);
      tf1.setBounds(520,50,100,20);
      l2.setBounds(450,80,70,20);
      tf2.setBounds(520,80,100,20);
      l3.setBounds(450,110,70,20);
      tf3.setBounds(520,110,100,20);
        l4.setBounds(450,140,70,20);
      tf4.setBounds(520,140,100,20);
      btn.setBounds(450,290,80,30);
        add(l1);
        add(l2);
        add(l3);
        add(l4);
        add(tf1);
        add(tf2);
        add(tf3);
        add(tf4);
        add(btn);
```

```
    btn.addActionListener(this);
      }
      public void actionPerformed(ActionEvent e)
      {
        if(e.getSource()==btn)
              {
              float n1,n2,n3,max;
        n1=Float.parseFloat(tf1.getText());
        n2=Float.parseFloat(tf2.getText());
        n3=Float.parseFloat(tf3.getText());
              if(n1>n2 && n1>n3)
              {
              max=n1;
              }
              else if(n2>n1 && n2>n3)
        {
              max=n2;
              }
              else
              {
              max=n3;
              }
        tf4.setText(String.valueOf(max));
              }
      }
      /*public static void main(String[] args)
      {
              maximum m = new maximum();
        m.setTitle("Maximum");
        m.setSize(150,150);
        m.setVisible(true);
      }*/
      }
```

maxThree.html

```
<html>
<head>
<title>Maximum of Three</title>
</head>
<body>
<applet code="maxThree.class" width="400" height="400">
</applet>
</body>
</html>
```

Output Screenshot:



Number 1: 55

Number 2: 106

Number 3: 987

MAXIMUM: 987.0

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 3 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:

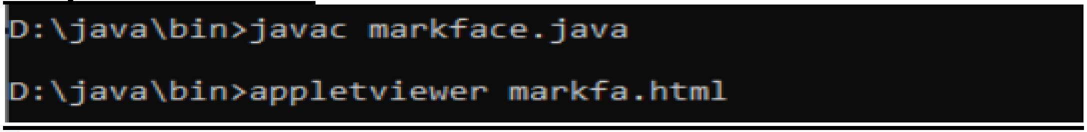
markface.java

```
import java.awt.*;
import java.applet.*;
import java.awt.event.*;
public class markface extends Applet implements ActionListener
{
        Label label1,label2,label3;
        TextField t1,t2,t3;
        Button btn;
        public void init()
        {
                label1=new Label("Mark1:");
                add(label1);
                t1=new TextField(20);
                add(t1);
                label2=new Label("Mark2:");
                add(label2);
                t2=new TextField(20);
                add(t2);
                label3=new Label("Mark3:");
                add(label3);
                t3=new TextField(20);
                add(t3);
                btn=new Button("Enter");
                add(btn);
                btn.addActionListener(this);
        }
        public void actionPerformed(ActionEvent e)
        {
                if(e.getSource()==btn)
                {
                        int mark1,mark2,mark3,total,percent;
                        mark1= Integer.parseInt(t1.getText());
                        mark2= Integer.parseInt(t2.getText());
                        mark3= Integer.parseInt(t3.getText());
                        total=mark1+mark2+mark3;
                        percent=((total*100)/300);
                        t3.setText(""+percent);
                        Graphics g=getGraphics();
                        int percentage= Integer.parseInt(t3.getText());
                        if(percentage>50)
                        {
                                g.drawOval(80,70,150,150);
                                g.fillOval(120,120,15,15);
                                g.fillOval(170,120,15,15);
                                g.drawArc(130,150,50,50,180,180);
                         }
                         else
                        {
                                g.drawOval(80,70,150,150);
                                g.fillOval(120,120,15,15);
                                g.fillOval(170,120,15,15);
                                g.drawArc(120,180,60,70,0,180);
                        }
                }
        }
}
```

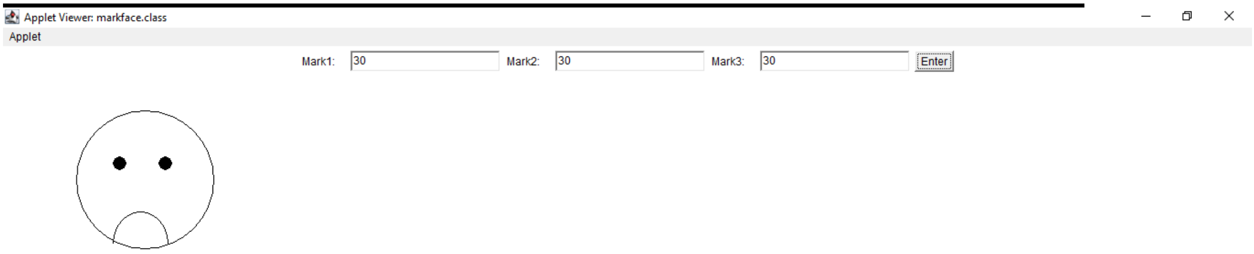
markfa.html

```
<html>
<body>
<applet code="markface.class" width="300" height="300">
</applet>
</body>
</html>
```

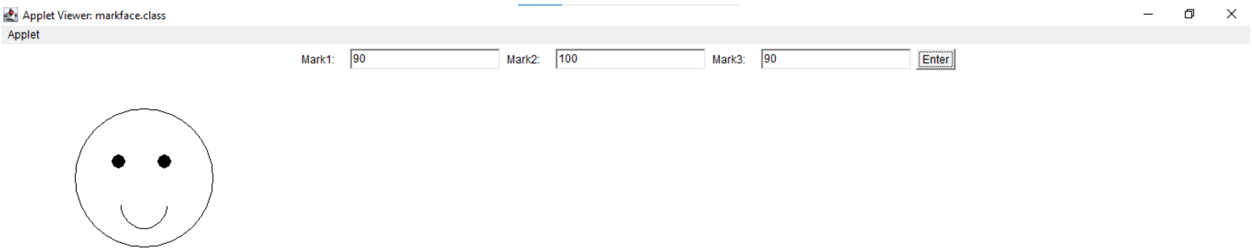
Output Screenshot:



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

D:\java\bin>appletviewer markfa.html
```



Mark1: 30 Mark2: 30 Mark3: 30



Mark1: 90 Mark2: 100 Mark3: 90

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:

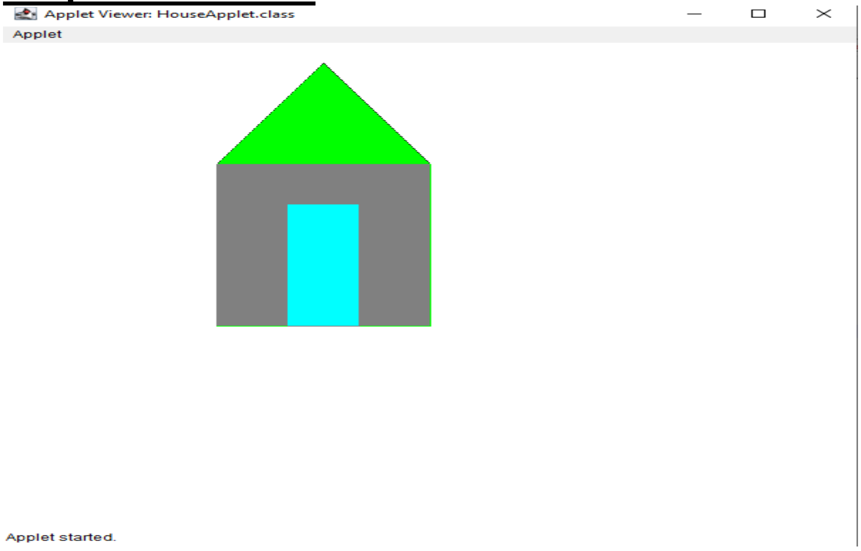
HouseApplet.java

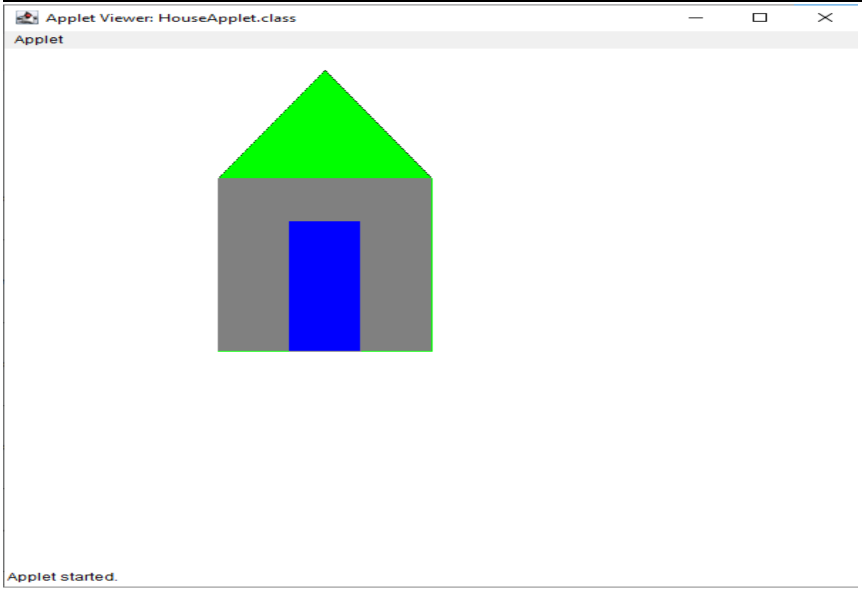
```
import java.awt.*;
import java.applet.*;
import java.awt.event.*;
public class HouseApplet extends Applet implements MouseListener
{
int a,b;
public void init()
{
addMouseListener( this);
}
public void paint(Graphics g)
{
int x[]={150,300,225};
int y[]={150,150,25};
g.drawPolygon(x,y,3);

g.setColor(Color.green);
g.fillPolygon(x,y,3);
g.drawRect(150,150,150,200);//House
g.setColor(Color.gray);
g.fillRect(150,150,150,200);
g.drawRect(200,200,50,150);//Door
g.setColor(Color.cyan);
g.fillRect(200,200,50,150);
if(a>200 && a<300 && b>200 && b<300)
{
g.setColor(Color.blue);
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="HouseApplet.class" width="600" height="600"></applet>
</body>
</html>
```

Output Screenshot:





<u>Result:</u>

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:

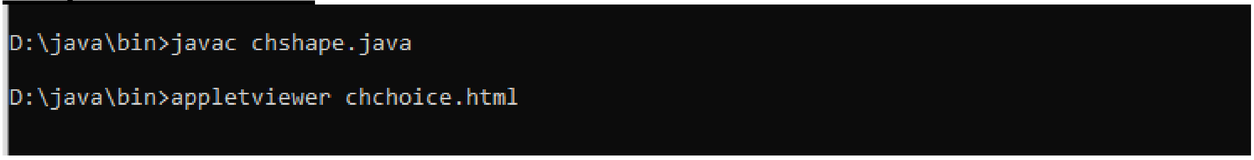
chshape.java

```
import java.awt.*;
import java.applet.*;
import java.awt.event.*;
public class chshape extends Applet implements ItemListener

{
        Choice ch=new Choice();
        int n;
        public void init()
        {

                ch.addItem("Choose the option");
                ch.addItem("Rectangle");
                ch.addItem("Triangle");
                ch.addItem("Square");
                ch.addItem("Circle");
                ch.addItemListener(this);
                add(ch);
        }
        public void paint(Graphics g)
        {
                if(n==0)
                {
                }
                if(n==1)
          {
                        g.drawRect(10,10,50,100);
          }
          if(n==2)
          {
                        int[] xPoints={150,50,250};
                        int[] zPoints={50,150,150};
                        g.drawPolygon(xPoints,zPoints,3);
                }
                if(n==3)
                {
                        g.drawRect(100,100,50,50);
                }
                if(n==4)
                {
                        g.drawOval(80,70,150,150);
                }
        }
        public void itemStateChanged(ItemEvent e)
        {
                n=ch.getSelectedIndex();
                repaint();
        }
}
```

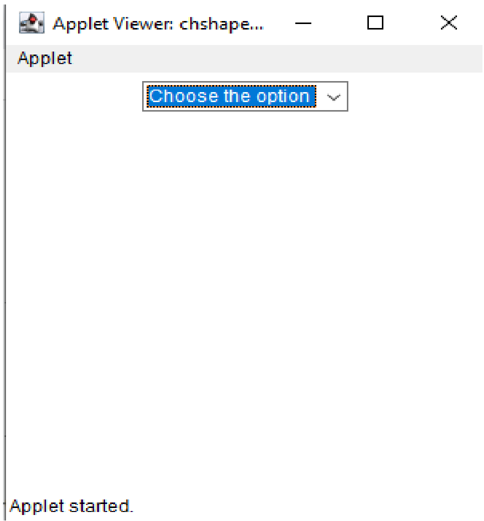
choice.html

```
<html>
<body>
<applet code="chshape.class" width="300" height="300">
</applet>
</body>
</html>
```

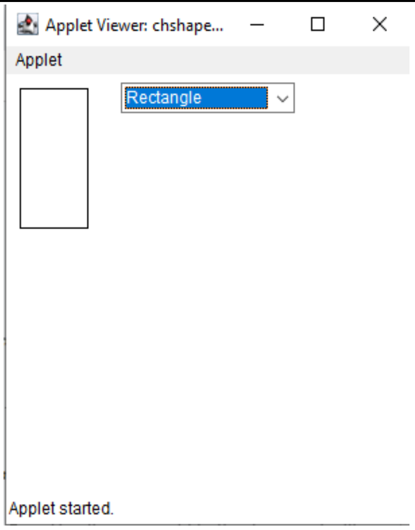
Output Screenshot:



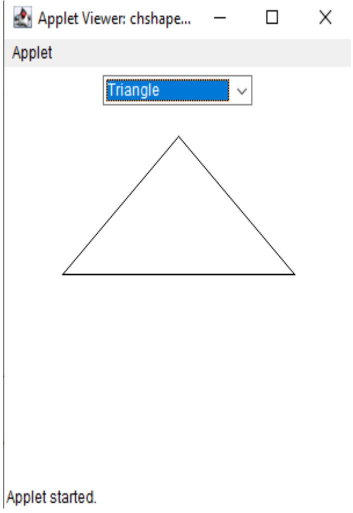
```
D:\java\bin>javac chshape.java

D:\java\bin>appletviewer chchoice.html
```



Applet started.



Applet started.



Applet started.

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.

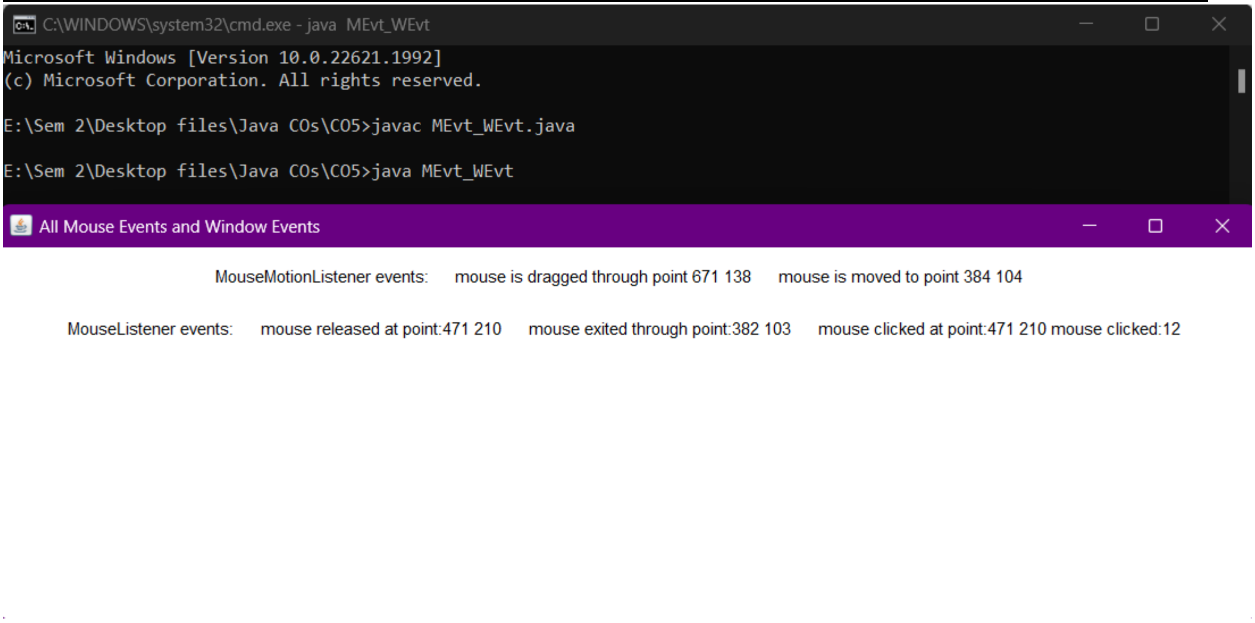
CO5:

Design applications using files and network concepts

Procedure:

```
import java.awt.*;
import java.awt.event.*;

public class MEvt_WEvt extends Frame implements MouseMotionListener, MouseListener
{
   static Label label1, label2, label3, label4, label5;

   public MEvt_WEvt()
{
      setTitle("All Mouse Events and Window Events");
      setSize(900, 300);
      setLayout(new FlowLayout());
      addMouseMotionListener(this);
      addMouseListener(this);

      Panel p = new Panel();
      Panel p1 = new Panel();

      Label l1 = new Label("MouseMotionListener events:");
      Label l2 = new Label("MouseListener events:");

      label1 = new Label("no event");
      label2 = new Label("no event");
      label3 = new Label("no event");
      label4 = new Label("no event");
      label5 = new Label("no event");

      p.add(l1);
      p.add(label1);
      p.add(label2);
      p1.add(l2);
      p1.add(label3);
      p1.add(label4);
      p1.add(label5);
      add(p);
      add(p1);
      addWindowListener(new WindowAdapter()
     {
         public void windowClosing(WindowEvent e)
        {
            System.exit(0);
        }
      });
      setVisible(true);
   }

   public static void main(String[] args)
   {
      new MEvt_WEvt();
   }
   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());
   }
}
```

<u>Output Screenshot:</u>



C:\WINDOWS\system32\cmd.exe - java MEvt_WEvt

```
Microsoft Windows [Version 10.0.22621.1992]
(c) Microsoft Corporation. All rights reserved.

E:\Sem 2\Desktop files\Java COs\CO5>javac MEvt_WEvt.java

E:\Sem 2\Desktop files\Java COs\CO5>java MEvt_WEvt
```

All Mouse Events and Window Events

```
MouseMotionListener events:   mouse is dragged through point 671 138   mouse is moved to point 384 104

MouseListener events:   mouse released at point:471 210   mouse exited through point:382 103   mouse clicked at point:471 210 mouse clicked:12
```

Result:

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

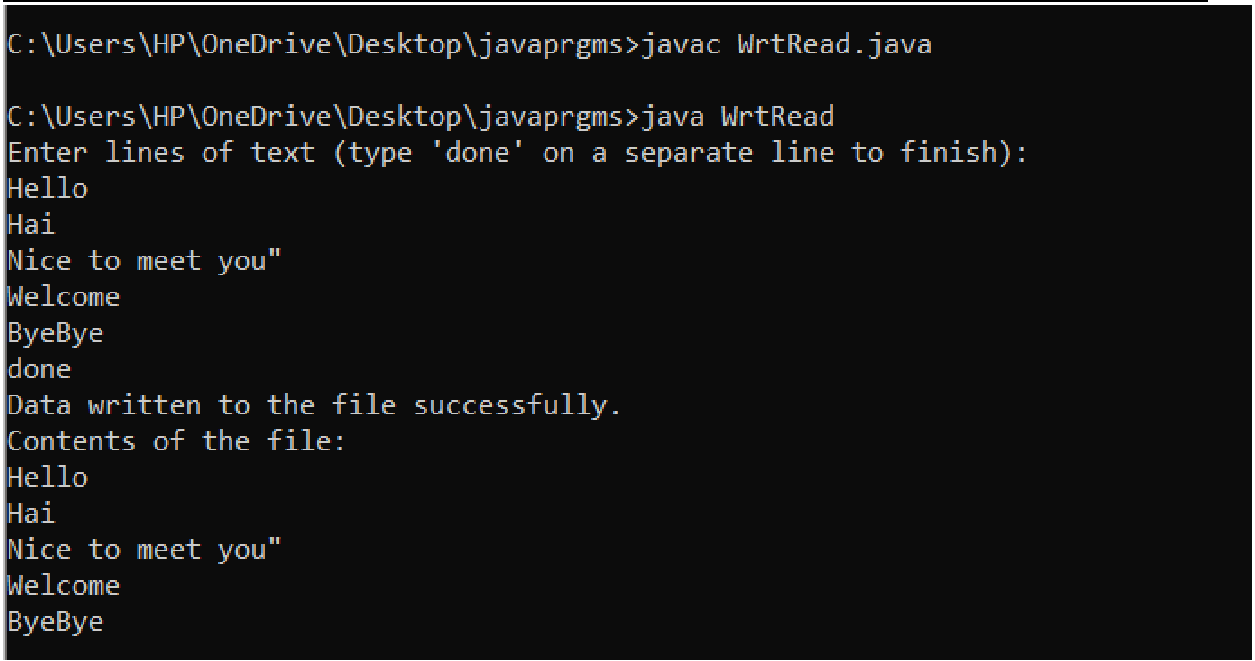
Experiment No.: 39

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

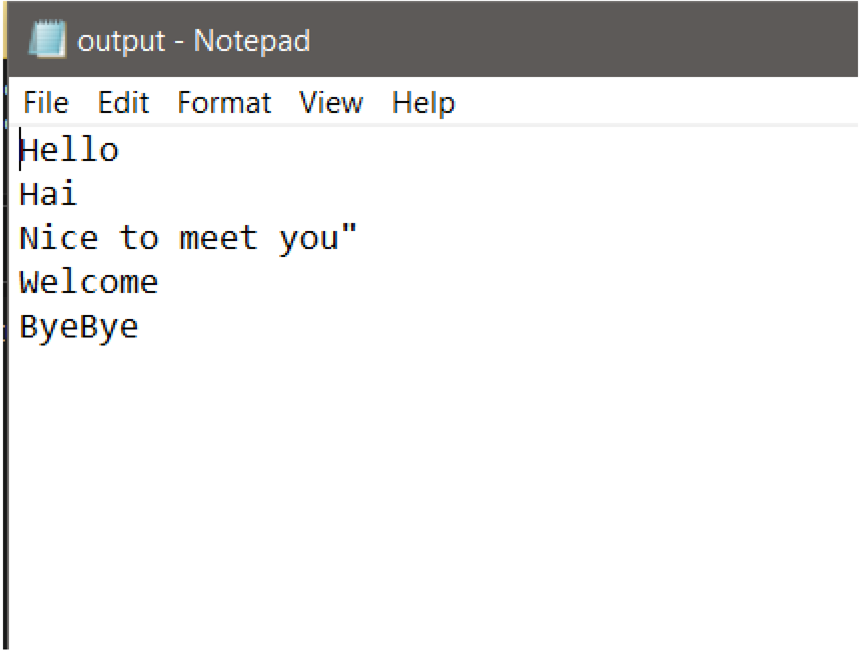
CO6:
Design applications using files and networking concepts.

Procedure:
```
import java.io.*;
import java.util.Scanner;

public class WrtRead
{
   public static void main(String[] args)
          {
       try
                   {
           FileWriter writer = new FileWriter("output.txt");
           Scanner scanner = new Scanner(System.in);
           System.out.println("Enter lines of text (type 'done' on a separate line to finish):");
           String inputLine;
           while (!(inputLine = scanner.nextLine()).equalsIgnoreCase("done"))
                             {
              writer.write(inputLine + "\n");
           }
           writer.close();
           System.out.println("Data written to the file successfully.");
           FileReader reader = new FileReader("output.txt");
           BufferedReader bufferedReader = new BufferedReader(reader);
           String line;
           System.out.println("Contents of the file:");
           while ((line = bufferedReader.readLine()) != null)
                             {
              System.out.println(line);
           }
           bufferedReader.close();
           scanner.close();
       }
                   catch (IOException e)
                   {
           e.printStackTrace();
       }
   }
}
```

Output Screenshot:



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

C:\Users\HP\OneDrive\Desktop\javaprgms>java WrtRead
Enter lines of text (type 'done' on a separate line to finish):
Hello
Hai
Nice to meet you"
Welcome
ByeBye
done
Data written to the file successfully.
Contents of the file:
Hello
Hai
Nice to meet you"
Welcome
ByeBye
```



```
Hello
Hai
Nice to meet you"
Welcome
ByeBye
```

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.

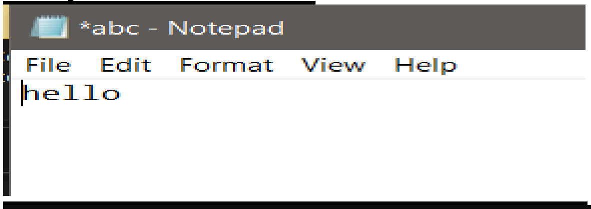
CO6:

Design applications using files and networking concepts.

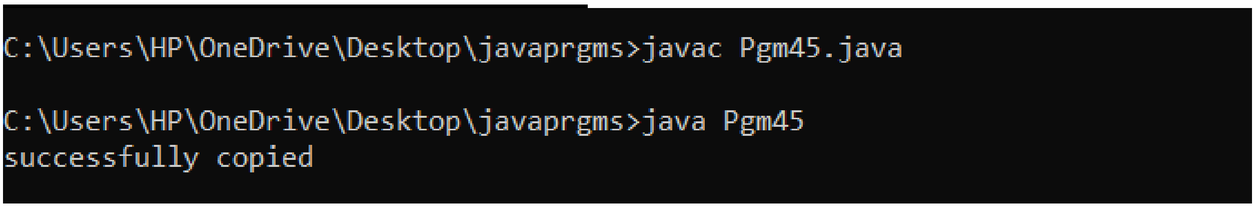
Procedure:

```
import java.io.*;
public class Pgm45
{
        public static void main(String[] args) throws Exception {
        {
                FileInputStream fileinput=new FileInputStream("abc.txt") ;
                FileOutputStream fileoutput=new FileOutputStream("2.txt");
                int i;
                while ((i=fileinput.read()) !=-1)
                {
                        fileoutput.write(i);
                }
                System.out.println("successfully copied");
        }
        }
}
```

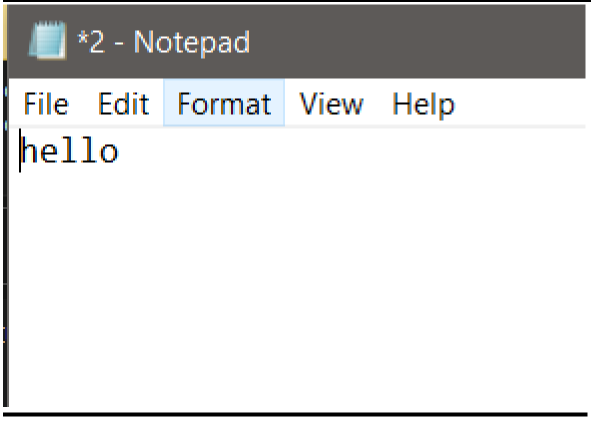
Output Screenshot:



```
hello
```



```
C:\Users\HP\OneDrive\Desktop\javaprgms>javac Pgm45.java

C:\Users\HP\OneDrive\Desktop\javaprgms>java Pgm45
successfully copied
```



```
hello
```

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.

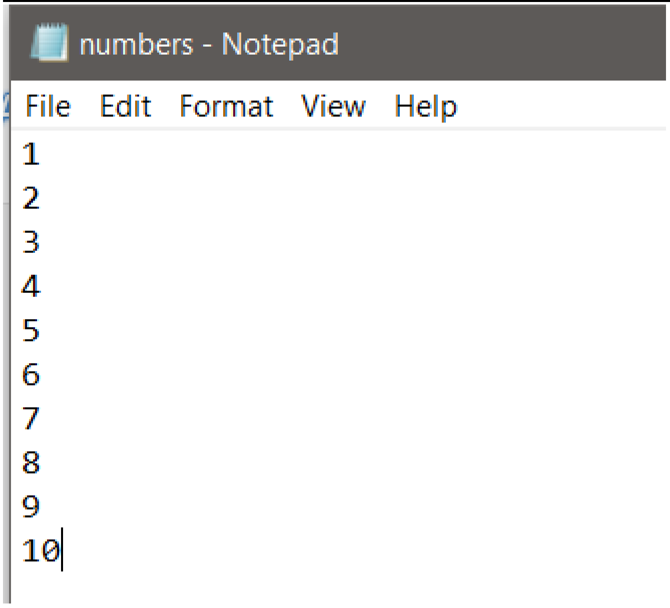
CO6:

Design applications using files and networking concepts.

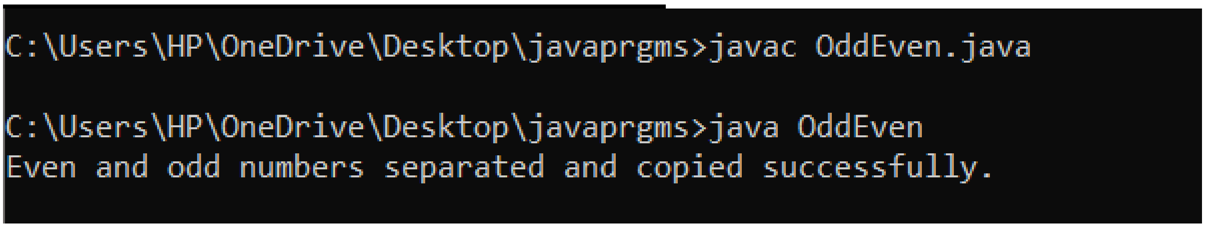
Procedure:

```
import java.io.*;
public class OddEven
{
   public static void main(String[] args)
         {
       try
                  {
          FileReader reader = new FileReader("numbers.txt");
          BufferedReader bufferedReader = new BufferedReader(reader);
          FileWriter evenWriter = new FileWriter("evens.txt");
          FileWriter oddWriter = new FileWriter("odds.txt");
          String line;
          while ((line = bufferedReader.readLine()) != null)
                            {
             int number = Integer.parseInt(line);
             if (number % 2 == 0)
                                     {
                evenWriter.write(number + "\n");
             } else
                                     {
                oddWriter.write(number + "\n");
             }
          }
          bufferedReader.close();
          evenWriter.close();
          oddWriter.close();

          System.out.println("Even and odd numbers separated and copied successfully.");
       }
                  catch (IOException | NumberFormatException e)
                  {
          e.printStackTrace();
       }
   }
}
```

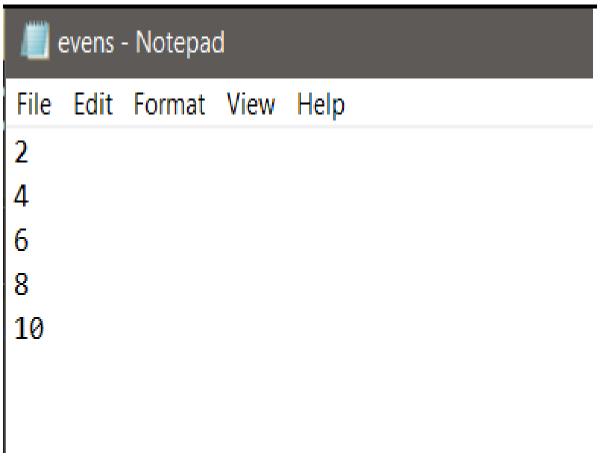
Output Screenshot:



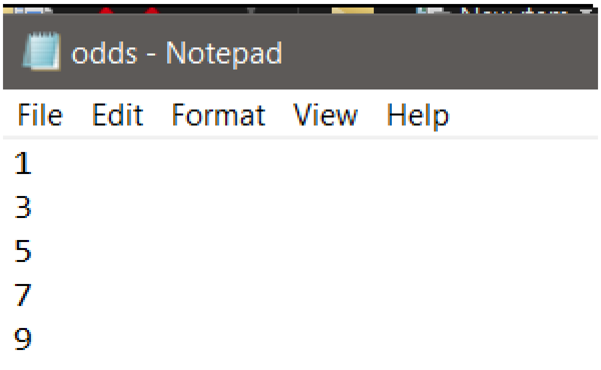
```
1
2
3
4
5
6
7
8
9
10
```



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

C:\Users\HP\OneDrive\Desktop\javaprgms>java OddEven
Even and odd numbers separated and copied successfully.
```



```
2
4
6
8
10
```



```
1
3
5
7
9
```

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:

Client.java

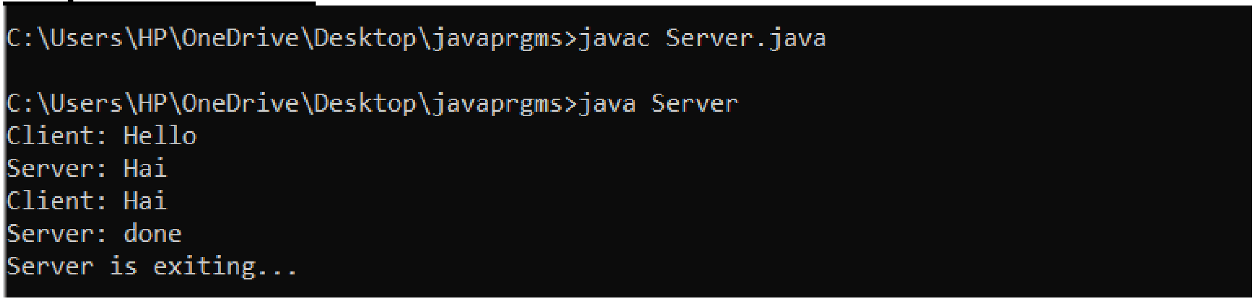
```
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.util.Scanner;

public class Client
{
   public static void main(String[] args)
  {
      try
      {
         DatagramSocket socket = new DatagramSocket();
         InetAddress serverAddress = InetAddress.getByName("localhost");
         int serverPort = 9876;
         Scanner scanner = new Scanner(System.in);
         while (true)
           {
            byte[] sendData;
            System.out.print("Client: ");
            String clientMessage = scanner.nextLine();
            sendData = clientMessage.getBytes();
            DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length,
serverAddress, serverPort);
            socket.send(sendPacket);

            if (clientMessage.equals("done"))
            {
               System.out.println("Client is exiting...");
               break;
            }
            byte[] receiveData = new byte[1024];
            DatagramPacket receivePacket = new DatagramPacket(receiveData,
receiveData.length);
            socket.receive(receivePacket);
         String serverMessage = new String(receivePacket.getData(), 0,
receivePacket.getLength());
            System.out.println("Server: " + serverMessage);

            if (serverMessage.equals("done"))
            {
               System.out.println("Server is exiting...");
               break;
            }
         }
         socket.close();
         scanner.close();
      }
      catch (Exception e)
     {
         e.printStackTrace();
      }
   }
}
```

Server.java

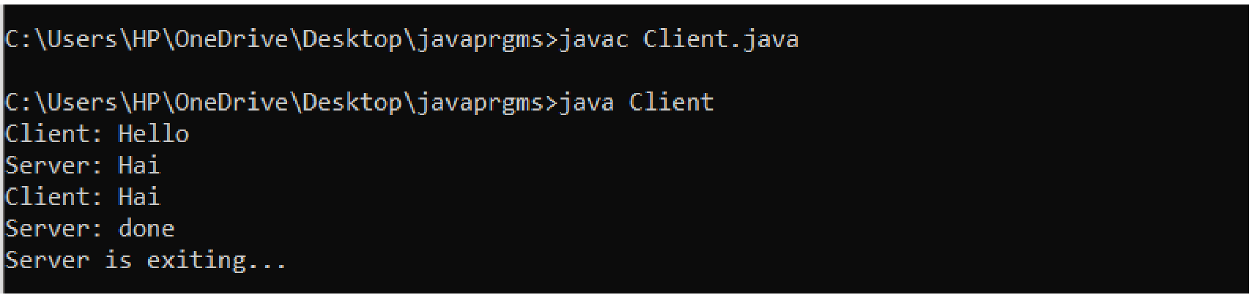
```
import java.net.DatagramPacket;
import java.net.DatagramSocket;
public class Server
{
   public static void main(String[] args)
         {
      try
                  {
         DatagramSocket socket = new DatagramSocket(9876);
         byte[] receiveData = new byte[1024];
         byte[] sendData;
         while (true)
                           {
            DatagramPacket receivePacket = new DatagramPacket(receiveData,
receiveData.length);
            socket.receive(receivePacket);
            String clientMessage = new String(receivePacket.getData(), 0,
receivePacket.getLength());
            System.out.println("Client: " + clientMessage);
            if (clientMessage.equals("done"))
                                    {
               System.out.println("Server is exiting...");
               break;
            }
            System.out.print("Server: ");
            String serverMessage = System.console().readLine();
            sendData = serverMessage.getBytes();
            DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length,
receivePacket.getAddress(), receivePacket.getPort());
            socket.send(sendPacket);
            if (serverMessage.equals("done"))
                                    {
               System.out.println("Server is exiting...");
```

```
                break;
            }
        }
        socket.close();
    }
                    catch (Exception e)
                    {
        e.printStackTrace();
    }
  }
}
```

Output Screenshot:



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

C:\Users\HP\OneDrive\Desktop\javaprgms>java Server
Client: Hello
Server: Hai
Client: Hai
Server: done
Server is exiting...
```



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

C:\Users\HP\OneDrive\Desktop\javaprgms>java Client
Client: Hello
Server: Hai
Client: Hai
Server: done
Server is exiting...
```

Result:

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