20MCA132 OBJECT ORIENTED PROGRAMMING LAB

RECORD

VIVIN V. ABRAHAM
ROLL NO: 42
S2-REG-MCA

TABLE OF CONTENTS

Experiment No:1

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.

PROGRAM

```
class product{
	int pcode;
	String pname;
	float price;
	void getdat( int x,String y,float z){
	pcode=x;
	pname=y;
	price=z;
	}
	void showdata(){
	System.out.println("product code= "+pcode+ " product name= " +pname+" product price= "+price);
	}

}
public class produc{
public static void main(String[] args){
product p1=new product();
product p2=new product();
product p3=new product();
p1.getdat(101,"Soap",50);
p2.getdat(102,"brush",25);
p3.getdat(103,"broom",40);
p1.showdata();
p2.showdata();
p3.showdata();
if ((p1.price)<(p2.price)&&(p1.price)<(p3.price))
	{
		System.out.println("The cheapest product is "+p1.pname);
	}
	else if ((p2.price)<(p1.price)&&(p2.price)<(p3.price))
	{
		System.out.println("The cheapest product is "+p2.pname);
	}
	else
	{
		System.out.println("The cheapest product is "+p3.pname);
```

```
        }
}
}
```

OUTPUT

```
D:\java_lab>java produc
product code= 101 product name= Soap product price= 50.0
product code= 102 product name= brush product price= 25.0
product code= 103 product name= broom product price= 40.0
The cheapest product is brush

D:\java_lab>
```

RESULT

The program has been executed and output verified.

Experiment No:2

Read 2 matrices from the console and perform matrix addition.

PROGRAM

```
import java.util.Scanner;
class matadd
{
	public static void main(String[] args){
	int row,col,i,j;
	Scanner sc= new Scanner(System.in);
	System.out.println("Enter the number of rows");
	row = sc.nextInt();
	System.out.println("Enter the no. of columns");
	col = sc.nextInt();

	int mat1[][]=new int [row][col];
	int mat2[][]=new int [row][col];
	int mat3[][]=new int [row][col];

System.out.println("Enter the matrix");
for(i=0;i<row;i++)
{
	for(j=0;j<col;j++)
	{
		mat1[i][j]=sc.nextInt();
	}
	System.out.println();
}
System.out.println("Enter the matrix");
for(i=0;i<row;i++)
{
	for(j=0;j<col;j++)
	{
		mat2[i][j]=sc.nextInt();
	}
	System.out.println();
}
for(i=0;i<row;i++)
{
	for(j=0;j<col;j++)
	{
		mat3[i][j]=mat1[i][j]+mat2[i][j];
	}
}
System.out.println("The sum of the matrix is");
for(i=0;i<row;i++)
```

```
{
        for(j=0;j<col;j++)
        {
                System.out.println(mat3[i][j]);
        }
        System.out.println();
}
}
}
```

OUTPUT

```
D:\java_lab>java matadd
Enter the number of rows
2
Enter the no. of columns
2
Enter the matrix
2 3

3 4

Enter the matrix
5 7

4 6

The sum of the matrix is
7
10

7
10


D:\java_lab>javac matadd.java

D:\java_lab>
```

RESULT

The program has been executed and output verified.

Experiment No:3

Add complex numbers

PROGRAM

```
class comple
        {
        int real;
        int imaginary;
        void getdata(int x, int y)
        {
                real=x;
                imaginary=y;
        }
        void showdata()
        {
                System.out.println("complex number: "+real+ " + "+imaginary+"i" );
        }
}
public class complex{
public static void main(String[] args){
comple o1=new comple();
comple o2=new comple();
o1.getdata(1,2);
o2.getdata(3,4);
o1.showdata();
o2.showdata();
System.out.println("Sum : "+(o1.real+o2.real)+ " + " +(o1.imaginary+o2.imaginary)+"i");
}
}
```

OUTPUT

```
D:\java_lab>javac complex.java

D:\java_lab>java complex
complex number: 1 + 2i
complex number: 3 + 4i
Sum : 4 + 6i

D:\java_lab>
```

RESULT

The program has been executed and output verified.

Experiment No:4

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

PROGRAM

```
import java.util.Scanner;
class Sym1
{
	public static void main(String[] args)
	{
		int row,col,i,j;
		int x=0;
		Scanner ob= new Scanner(System.in);
		System.out.print("Enter the number of Rows of matrix:");
		row=ob.nextInt();
		System.out.print("Enter the number of columns of matrix:");
		col=ob.nextInt();
		if(col==row)
		{
			System.out.println("Since the No. of Row is eqaul to No.of Columuns \nMatrix can become Symmetric");
		}
		else
		{
			System.out.println("No. of Rows and Colum must be same");
			System.exit(0);//to exit the program

		}
		int mat[][]=new int[row][col];
		System.out.println("Enter the Matrix");
		for(i=0;i<row;i++)
		{
			for(j=0;j<col;j++)
			{
				mat[i][j]=ob.nextInt();
			}
			System.out.println();
		}
		for(i=0;i<row;i++)
		{
			for(j=0;j<row;j++)
			{
				if (mat[i][j]==mat[j][i])
				{
					x=1;
					//System.out.println("Given matrix are Symmetric");
					break;
				}
```

```
                        else
                        {
                                x=0;
                                //System.out.println("Given matrix are not a symmetric matrix");
                                break;
                        }
                }
        }
        if (x==1)
        {
                System.out.println("Given Matrix is Symmetric.");
        }
        else
        {
                System.out.println("Given Matrix is not Symmetric.");
        }
}
}
```

OUTPUT

```
D:\java_lab>javac Sym1.java

D:\java_lab>java Sym1
Enter the number of Rows of matrix:2
Enter the number of columns of matrix:2
Since the No. of Row is eqaul to No.of Columuns
Matrix can become Symmetric
Enter the Matrix
2
4

4
2

Given Matrix is Symmetric.

D:\java_lab>
```

RESULT

The program has been executed and output verified.

Experiment No:5

Create CPU with attribute price. Create inner class Processor (no. of cores, manufacturer) and static nested class RAM (memory, manufacturer). Create an object of CPU and print information of Processor and RAM.

PROGRAM

```
class CPU
{
double x=1000.00;
class Processor
{
        int y=10;
        String i="Microsoft";
        static class Ram
        {
                String j="5GB";
                String k="Intel";
        }
}
}
public class chr
{
public static void main(String[] args)
{
CPU Co = new CPU();
CPU.Processor Po = Co.new Processor();
CPU.Processor.Ram Ro= new CPU.Processor.Ram();
System.out.println(Co.x);
System.out.println(Po.y);
System.out.println(Po.i);
System.out.println(Ro.j);
System.out.println(Ro.k);
}
}
```

OUTPUT

RESULT

The program has been executed and output verified

Experiment No:6

Program to Sort strings

PROGRAM

```
class sortstring
{
	public static void main(String[] args)
	{
		String names[]={"Amal","Jyothi","College","of","Engineering"};
		String temp;
		int n = names.length;
		for(int i=0;i<n;i++)
		{
			for(int j=i+1;j<n;j++)
			{
				if(names[i].compareTo(names[j])>0)
				{
					temp=names[i];
					names[i]=names[j];
					names[j]=temp;
				}
			}
		}
		System.out.println("The sorted array of string is:");
		for(int i=0;i<n;i++)
		{
			System.out.println(names[i]);
		}
}
}
```

OUTPUT

```
D:\java_lab>javac sortstring.java

D:\java_lab>java sortstring
The sorted array of string is:
Amal
College
Engineering
Jyothi
of
```

RESULT

The program has been executed and output verified.

Experiment No:7

Search an element in an array.

PROGRAM

```
import java.util.*;
public class searche
{
	public static void main(String[] args)
	{
		int n,i,b,flag=0;
		Scanner s= new Scanner(System.in);
		System.out.println("Enter the number of elements");
		n=s.nextInt();
		int a[]=new int[n];
		System.out.println("Enter the elements of the array");
		for(i=0;i<n;i++)
		{
			a[i]=s.nextInt();
		}
		System.out.println("Enter the element to search");
		b=s.nextInt();
		for(i=0;i<n;i++)
		{
			if(a[i]==b)
			{
				flag=1;
				break;
			}
			else
			{
				flag=0;
			}
```

```
            }
            if(flag==1)
            {
                System.out.println("Element is found in position "+(i+1));

            }
            else
            {
                System.out.println("Element not found");
            }
        }
}
```

OUTPUT

```
D:\java_lab>javac searche.java

D:\java_lab>java searche
Enter the number of elements
3
Enter the elements of the array
6
6
3
Enter the element to search
1
Element not found
```

RESULT

The program has been executed and output verified.

Experiment No:8

Perform string manipulations

PROGRAM

```
public class examp2
{
	public static void main(String[] args)
	{
		String x="All Are Welcome";
		int a= x.length();
		System.out.println("The length of the string is "+a);
		System.out.println(x.toUpperCase());
		System.out.println(x.toLowerCase());
		System.out.println(x.indexOf("Are"));
		String y="College";
		String z="College";
		System.out.println(x+"to"+y);
		System.out.println(z.concat(y));
		System.out.println(x.substring(5,12));//print a data from particular region
		if(y.equals(z))//compare two strings are equal or not
		{
			System.out.println("Strings are equal");
		}
		else
		{
			System.out.println("not equal");
		}
		System.out.println(x.charAt(7));//character at a position
		System.out.print("The reverse of "+y+" is : ");
		for(int j=y.length()-1; j>=0;j--)//reverse of a string
		{
			System.out.print(y.charAt(j));
		}
		if(x.contains("How"))
		{
			System.out.println("\nGiven Element found in "+x);
		}
		else
		{
			System.out.println("\nElement not found");
		}
		System.out.println(x.replace("All","You"));//replace function
	}
}
```

OUTPUT

```
D:\java_lab>javac examp2.java

D:\java_lab>java examp2
The length of the string is 15
ALL ARE WELCOME
all are welcome
4
All Are WelcometoCollege
CollegeCollege
re Welc
Strings are equal

The reverse of College is : egelloC
Element not found
You Are Welcome

D:\java_lab>
```

RESULT

The program has been executed and output verified.

Experiment No:9

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

PROGRAM

```
import java.util.*;

public class Employyee

{

        public static void main(String[] args)

        {

                Scanner ab=new Scanner(System.in);

                Scanner ac=new Scanner(System.in);

                System.out.println("Enter the number of Employee");

                int n,i,flag=0;

                n=ab.nextInt();

                int eNo[]=new int[n];

                String eName[]= new String[n];

                float eSalary[]= new float[n];

                System.out.println("Enter the Employee informations");

                for(i=0;i<n;i++)

                {

                        System.out.println("Employee Number");

                        eNo[i]=ab.nextInt();

                        System.out.println("Employee Name");

                        eName[i]=ac.nextLine();

                        System.out.println("Employee salary");

                        eSalary[i]=ab.nextInt();

                }

                System.out.println("To search an Employee details please enter Employee Number");

                int d=ab.nextInt();

                for(i=0;i<n;i++)
```

```
		{
			if(eNo[i]==d)
			{
				flag=1;
				break;
			}
			else
			{
				flag=0;
			}
			}
		if(flag==1)
		{
			System.out.println("Employ_No "+eNo[i]+" Name "+eName[i]+" Salary "+eSalary[i]);
		}
		else
		{
			System.out.println("Not a valid Employee number");
		}
	}
}
```

OUTPUT

```
D:\java_lab>javac Employyee.java

D:\java_lab>java Employyee
Enter the number of Employee
3
Enter the Employee informations
Employee Number
101
Employee Name
Anu
Employee salary
300000
Employee Number
102
Employee Name
Jose
Employee salary
23000
Employee Number
103
Employee Name
Binu
Employee salary
12000
To search an Employee details please enter Employee Number
102
Employ_No 102 Name Jose Salary 23000.0

D:\java_lab>
```

RESULT

The program has been executed and output verified.

Experiment No:10

Area of different shapes using overloaded functions

PROGRAM

```
import java.util.*;

public class area

{

		static double circle(double r)

		{

			return 3.14*r*r;

		}

		static int rectangle(int a,int b)

		{

			return a*b;

		}

		static double triangle(double c, double d)

		{

			return 0.5*c*d;

		}

		static double cone(double e,double f)

		{
```

```
        return (3.14*e*e)+(3.14*e*f);
}
static double sphere(double r1)
{
        return 4*3.14*r1;
}
static double cylinder(double h,double r2)
{
        return (2*3.14*r2)+(r2+h);
}
public static void main(String[] args)
{
        Scanner o= new Scanner(System.in);

        System.out.println();
        System.out.println("1.CHECK AREA OF CIRCLE");
        System.out.println();
        System.out.println("**********************");
        System.out.print("Enter the Radius of Circle : ");
        double r=o.nextInt();
        double ab=circle(r);
        System.out.println("Area of Circle : "+ab);

        System.out.println();
        System.out.println("2.CHECK AREA OF RECTANGLE");
        System.out.println();
        System.out.println("**********************");
        System.out.println();
        System.out.print("Enter the Length of Rectangle : ");
        int a=o.nextInt();
        System.out.print("Enter the Breadth of Rectangle : ");
        int b=o.nextInt();
```

```
int ac=rectangle(a,b);
System.out.println("Area of Rectangle : "+ac);

System.out.println();
System.out.println("3.CHECK AREA OF TRIANGLE");
System.out.println();
System.out.println("**********************");
System.out.println();
System.out.print("Enter the Height of Triangle : ");
double c=o.nextInt();
System.out.print("Enter the Base of Triangle : ");
double d=o.nextInt();
double cd=triangle(c,d);
System.out.println("Area of Triangle : "+cd);

System.out.println();
System.out.println("4.CHECK AREA OF CONE");
System.out.println();
System.out.println("**********************");
System.out.println();
System.out.print("Enter the Base Radius of Cone : ");
double e=o.nextInt();
System.out.print("Enter the Slant Height of Cone : ");
double f=o.nextInt();
double ef=cone(e,f);
System.out.println("Area of Cone : "+ef);

System.out.println();
System.out.println("5.CHECK AREA OF SPHERE : ");
System.out.println();
System.out.println("**********************");
System.out.println();
```

```
                System.out.print("Enter the Radius of Sphere : ");
                double r1=o.nextInt();
                double ra=sphere(r1);
                System.out.println("Area of Sphere : "+ra);



                System.out.println();
                System.out.println("6.CHECK AREA OF CYLINDER : ");
                System.out.println();
                System.out.println("**********************");
                System.out.println();
                System.out.print("Enter the Height of Cylinder : ");
                double h=o.nextInt();
                System.out.print("Enter the Radius of Cylinder : ");
                double r2=o.nextInt();
                double rh=cylinder(h,r2);
                System.out.println("Area of Cylinder : "+rh);
        }

}
```

OUTPUT

```
D:\java_lab>javac area.java

D:\java_lab>java area

1.CHECK AREA OF CIRCLE

********************
Enter the Radius of Circle : 3
Area of Circle : 28.259999999999998

2.CHECK AREA OF RECTANGLE

********************

Enter the Length of Rectangle : 2
Enter the Breadth of Rectangle : 5
Area of Rectangle : 10

3.CHECK AREA OF TRIANGLE

********************

Enter the Height of Triangle : 4
Enter the Base of Triangle : 2
Area of Triangle : 4.0

4.CHECK AREA OF CONE

********************

Enter the Base Radius of Cone : 7
Enter the Slant Height of Cone : 4
Area of Cone : 241.78000000000003

5.CHECK AREA OF SPHERE :

********************

Enter the Radius of Sphere : 3
Area of Sphere : 37.68

6.CHECK AREA OF CYLINDER :

********************

Enter the Height of Cylinder : 6
Enter the Radius of Cylinder : 3
Area of Cylinder : 27.84

D:\java_lab>
```

RESULT

The program has been executed and output verified.

Experiment No:11

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.

PROGRAM

```
import java.util.*;

class Employees

{
```

```
	int empid;
	String name,address;
	double salary;
	public Employees(int empid, String name, String address, double salary)
	{
		this.empid = empid;
		this.name = name;
		this.address = address;
		this.salary = salary;
	}
}
public class Teacher extends Employees
{
	String subject,department;
	public Teacher(int empid, String name, String address, double salary,String department,String subject )
	{
		super(empid, name, address, salary);
		this.subject = subject;
		this.department = department;
	}
	void display()
	{
		System.out.println("Empid : "+this.empid+" \nName : "+this.name+" \nSalary : "+this.salary+" \nAddress : "+this.address+" \ndepartment : "+this.department+" \nSubjects : "+this.subject);
	}
	public static void main(String[] args)
	{
		Scanner sc=new Scanner(System.in);
		Scanner oc=new Scanner(System.in);2		int n;
		System.out.print("Enter number of Teachers : ");
		n=sc.nextInt();
		Teacher obj[]=new Teacher[n];
		for(int i=0;i<n;i++)
```

```
            {
                int j = i+1;
                System.out.print("Enter Empid of teacher "+j+" : ");
                int Empid = sc.nextInt();
                System.out.print("Enter Name of teacher "+j+" : ");
                String Name = oc.nextLine();
                System.out.print("Enter Salary of teacher "+j+" : ");
                double Salary = sc.nextDouble();
                System.out.print("Enter Address of teacher "+j+" : ");
                String Address = oc.nextLine();
                System.out.print("Enter department of teacher "+j+" : ");
                String department =oc.nextLine();
                System.out.print("Enter Subjects of teacher "+j+" : ");
                String Subjects =oc.nextLine();
                obj[i] = new Teacher(Empid, Name, Address, Salary, department, Subjects);
            }
            System.out.println("\n-------------------------------------------------------------------------\n");
            System.out.println("Teacher's List \n");
            for(int i=0;i<n;i++)
            {
                obj[i].display();
            }
        }
}
```

OUTPUT

```
D:\java_lab>javac Teacher.java

D:\java_lab>java Teacher
Enter number of Teachers : 2
Enter Empid of teacher 1 : 101
Enter Name of teacher 1 : Grace Joseph
Enter Salary of teacher 1 : 30000
Enter Address of teacher 1 : Idukki
Enter department of teacher 1 : MCA
Enter Subjects of teacher 1 : Virtualization and Containers
Enter Empid of teacher 2 : 104
Enter Name of teacher 2 : Lisha Varghese
Enter Salary of teacher 2 : 40000
Enter Address of teacher 2 : Kottayam
Enter department of teacher 2 : MCA
Enter Subjects of teacher 2 : Database Management System

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

Teacher's List

Empid : 101
Name : Grace Joseph
Salary : 30000.0
Address : Idukki
department : MCA
Subjects : Virtualization and Containers
Empid : 104
Name : Lisha Varghese
Salary : 40000.0
Address : Kottayam
department : MCA
Subjects : Database Management System

D:\java_lab>
```

RESULT

The program has been executed and output verified.

Experiment No:12

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.

PROGRAM

```
import java.util.Scanner;

class Person

{

	String name,gender,address;

	int age;

	public Person(String name, String gender, String address, int age)

	{

		super();

		this.name = name;

		this.gender = gender;

		this.address = address;

		this.age = age;

	}

}

class Employee extends Person

{

	int empid;

	String company_name,qualification;

	double salary;

	public Employee(String name, String gender, String address, int age, int empid, String company_name,

	String qualification, double salary)

	{

		super(name, gender, address, age);

		this.empid = empid;
```

```
		this.company_name = company_name;
		this.qualification = qualification;
		this.salary = salary;
	}
}
class Teacher extends Employee
{
	String subject,department;
	int teacherid;
	public Teacher(String name, String gender, String address, int age, int empid, String company_name,
	String qualification, double salary, String subject, String department, int teacherid)
	{
		super(name, gender, address, age, empid, company_name, qualification, salary);
		this.subject = subject;
		this.department = department;
		this.teacherid = teacherid;
	}
	void display()
	{
		System.out.println("....Personal details...");
		System.out.println("Name : "+this.name+"\nGender : "+this.gender+"\nAge :"+this.age);
		System.out.println("...Employee details....");
		System.out.println("Empid : "+this.empid +"\ncompany_name : "+this.company_name+"\nSalary : "+this.salary+" Address : "+this.address+"\nqualification : "+this.qualification);
		System.out.println("...Teacher's details...");
		System.out.println("Teacherid : "+this.teacherid+ "\ndepartment : "+this.department+"\nSubjects : "+this.subject);
	}
```

```
}
public class Main5
{
	public static void main(String[] args)
	{
		Scanner s=new Scanner(System.in);
		Scanner os=new Scanner(System.in);
		int n;
		System.out.println("Enter number of Teachers : ");
		n=s.nextInt();
		Teacher obj[]=new Teacher[n];
		for(int i=0;i<n;i++)
		{
			System.out.println("Enter the person name:");
			String nam1=os.nextLine();
			System.out.println("Enter the Gender: ");
			String gen1=os.nextLine();
			System.out.println("Enter the Address: ");
			String adr1=s.next();
			System.out.println("Enter the Age:");
			int age1=s.nextInt();
			System.out.println("Enter the Employee id: ");
			int id1=s.nextInt();
			System.out.println("Enter the Company name: ");
			String cname1=os.nextLine();
			System.out.println("Enter the Salary:");
			double sal1=s.nextDouble();
			System.out.println("Enter the Qualification:");
			String qu1=os.nextLine();
```

```
                System.out.println("Enter the Teacher id: ");
                int tid1=s.nextInt();
                System.out.println("Enter the Department:");
                String dept1=os.nextLine();
                System.out.println("Enter the Subject:");
                String sub1=os.nextLine();
                obj[i]=new
Teacher(nam1,gen1,adr1,age1,id1,cname1,qu1,sal1,sub1,dept1,tid1);
        }
        System.out.println("\n---------------------------------------------------------------------------
\n");
        for(int i=0;i<n;i++)
        {
                obj[i].display();
        }
    }
}
```

OUTPUT

```
D:\java_lab>java Main5
Enter number of Teachers :
2
Enter the person name:
Anna
Enter the Gender:
Female
Enter the Address:
Kannur
Enter the Age:
24
Enter the Employee id:
102
Enter the Company name:
DR Technologies
Enter the Salary:
34000
Enter the Qualification:
MCA
Enter the Teacher id:
103
Enter the Department:
CS
Enter the Subject:
Virtualization
Enter the person name:
John
Enter the Gender:
Male
Enter the Address:
Trivandrum
Enter the Age:
25
Enter the Employee id:
105
Enter the Company name:
AC Technologies
Enter the Salary:
45000
Enter the Qualification:
MTech
Enter the Teacher id:
109
Enter the Department:
Mechanical
Enter the Subject:
Motor Mechanic

--------------------------------------------------------------------------
```

```
....Personal details...
Name : Anna
Gender : Female
Age :24
...Employee details....
Empid : 102
company_name : DR Technologies
Salary : 34000.0 Address : Kannur
qualification : MCA
...Teacher's details...
Teacherid : 103
department : CS
Subjects : Virtualization
....Personal details...
Name : John
Gender : Male
Age :25
...Employee details....
Empid : 105
company_name : AC Technologies
Salary : 45000.0 Address : Trivandrum
qualification : MTech
...Teacher's details...
Teacherid : 109
department : Mechanical
Subjects : Motor Mechanic

D:\java_lab>
```

RESULT

The program has been executed and output verified.

Experiment No:13

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.

PROGRAM

```
import java.util.Scanner;

class Publisher

{

	String Pubname;

	Publisher()

	{

		Scanner s=new Scanner(System.in);

		System.out.println("Enter publisher name");

		Pubname=s.nextLine();

	}

}

class Book extends Publisher

{

	String title, author;

	int price;

	Book()

	{

		Scanner s=new Scanner(System.in);

		Scanner os=new Scanner(System.in);

		System.out.println("Enter Title of the book");

		title=os.nextLine();

		System.out.println("Enter Author's name");

		author=os.nextLine();

		System.out.println("Enter price");

		price=s.nextInt();

	}

}
```

```
class Literature extends Book
{
	Literature()
	{
		System.out.println("Literature Books");
	}
	void display()
	{
		System.out.println("Publisher name: "+Pubname);
		System.out.println("Title of the book: "+title);
		System.out.println("Author's name: "+author);
		System.out.println("Price: "+price);
	}
}
public class Fiction extends Literature
{
	Fiction()
	{
		System.out.println("Friction Books");
	}
	void display()
	{
		super.display();
	}
	public static void main(String[] args)
	{
		int n;
		Scanner s=new Scanner(System.in);
		System.out.println("Enter the No of literature book: ");
		int a=s.nextInt();
		Literature L[]=new Literature[a];
```

```
for(int i=0;i<a;i++)
{
        L[i]=new Literature();
}
System.out.println("Enter the No of Fiction book: ");
int b=s.nextInt();
Fiction F[]=new Fiction[b];
for(int i=0;i<b;i++)
{
        F[i]=new Fiction();
}
int no;
System.out.println("Enter your choice of book \n1-Literature\n2-Fiction");

no=s.nextInt();
int type =no;
switch (no)
{
        case 1:
        System.out.println(".....Details of literature books ");

        for(int i=0;i<a;i++)
                L[i].display();
        break;
        case 2:
        System.out.println(".....Details of fiction books");
        for(int i=0;i<b;i++)
                F[i].display();
        break;
        default:
        System.out.println("Wrong input");
```

```
            }
        }
}
```

OUTPUT

```
D:\java_lab>javac Fiction.java

D:\java_lab>java Fiction
Enter the No of literature book:
2
Enter publisher name
A D
Enter Title of the book
A small story
Enter Author's name
Anin Jain
Enter price
560
Literature Books
Enter publisher name
S H
Enter Title of the book
A Large duck
Enter Author's name
Wiliam hook
Enter price
450
Literature Books
Enter the No of Fiction book:
2
Enter publisher name
R T
Enter Title of the book
R W
Enter Author's name
Polynath
Enter price
560
Literature Books
Friction Books
Enter publisher name
Q J
Enter Title of the book
In the Moon
Enter Author's name
Richi jin
Enter price
340
```

```
Literature Books
Friction Books
Enter your choice of book
1-Literature
2-Fiction
1
.....Details of literature books
Publisher name: A D
Title of the book: A small story
Author's name: Anin Jain
Price: 560
Publisher name: S H
Title of the book: A Large duck
Author's name: Wiliam hook
Price: 450
```

RESULT

The program has been executed and output verified.

Experiment No:14

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

PROGRAM

```
interface student
{
	void stresullt();

}
interface sports
{
	void spresult();
}
class result implements student,sports
{
	public void spresult()
	{
		String hundred="First";
		String twohundred="Second";
		String fivehundred="First";
		String relay="Second";
		System.out.println("Sports Result");
		System.out.println("Hundred Meter:"+hundred);
		System.out.println("Two Hundred Meter:"+twohundred);
		System.out.println("Five Hundred Meter:"+fivehundred);
		System.out.println("Relay:"+relay);
	}
	public void stresullt()
	{
		int physics=30;
		int chemistry=40;
		int maths=45;
		int english=50;
		int computer=50;
		System.out.println("Marks");
		System.out.println("Physics:"+physics);
		System.out.println("Chemistry:"+chemistry);
		System.out.println("Mathematics:"+maths);
		System.out.println("English:"+english);
		System.out.println("Computer:"+computer);
	}
	public static void main(String[] args)
	{
		result r = new result();
		r.stresullt();
		r.spresult();

	}
```

```
}
```

OUTPUT

```
D:\java_lab>javac result.java

D:\java_lab>java result
Marks
Physics:30
Chemistry:40
Mathematics:45
English:50
Computer:50
Sports Result
Hundred Meter:First
Two Hundred Meter:Second
Five Hundred Meter:First
Relay:Second

D:\java_lab>javac Results.java
```

RESULT

The program has been executed and output verified.

Experment:15

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.

PROGRAM

```
import java.util.Scanner;

interface Shape

{

   void input();

   void area();

   void perimeter();

}

class Circle implements Shape

{


   int r = 0;

   double pi = 3.14, ar = 0,per=0;

   public void input()

   {        Scanner s = new Scanner(System.in);

             System.out.print("Enter radius of circle:");

           r= s.nextInt();

   }

   public void area()

   {

      ar = pi * r * r;

      System.out.println("Area of circle:"+ar);

   }

   public void  perimeter()

   {
```

```
            per = 2 * pi * r;
            System.out.println("Perimeter of circle:"+per);
    }

}
class Rectangle implements Shape
{
    int l = 0, b = 0;
    double ar,per;
    public void input()
    {    Scanner s = new Scanner(System.in);
          System.out.print("Enter length of rectangle:");
        l = s.nextInt();
         System.out.print("Enter breadth of rectangle:");
        b = s.nextInt();
    }
    public void area()
    {
        ar = l * b;
        System.out.println("Area of rectangle:"+ar);
    }
    public void perimeter()
    {
              per = 2 * (l + b);
              System.out.println("Perimeter of rectangle:"+per);
    }
}
public class shapesss
{
```

```
public static void main(String[] args)
{ int n;
      Scanner s = new Scanner(System.in);
    Rectangle obj1 = new Rectangle();
      Circle obj2 = new Circle();


      System.out.println("1.Area of circle");
      System.out.println("2.Perimeter of circle");
      System.out.println("3.Area of rectangle");
      System.out.println("4.Perimeter of rectangle");
      System.out.println("Enter your option:");
        n= s.nextInt();
      switch(n) {
     case 1:
       obj2.input();
          obj2.area();
     break;
     case 2:
       obj2.input();
          obj2.perimeter();
     break;
        case 3:
       obj1.input();
          obj1.area();
     break;
     case 4:
       obj1.input();
          obj1.perimeter();
     break;
```

```
        default:
         System.out.println("Invalid option");
          }


    }
}
```

OUTPUT

```
D:\java_lab>javac shapesss.java

D:\java_lab>java shapesss
1.Area of circle
2.Perimeter of circle
3.Area of rectangle
4.Perimeter of rectangle
Enter your option:
1
Enter radius of circle:4
Area of circle:50.24

D:\java_lab>java shapesss
1.Area of circle
2.Perimeter of circle
3.Area of rectangle
4.Perimeter of rectangle
Enter your option:
3
Enter length of rectangle:23
Enter breadth of rectangle:4
Area of rectangle:92.0

D:\java_lab>
```

RESULT

The program has been executed and output verified.

Experiment No:16

7. Prepare bill with the given format using calculate method from interface.
Order No.

Product Id	Name	Quantity	unit price	Total
101	A	2	25	50
102	B	1	100	100
			Net. Amount	150

PROGRAM

```
interface bill
{
        int productdetails();
}
class product1 implements bill{
                int id = 101,quantity= 2,unit=25,total=0;
                String name="A";

        public int productdetails()
        {
        total = quantity * unit;
                System.out.println("Product Id :"+id);
                System.out.println("Name :"+name);
                System.out.println("Quantity :"+quantity);
                System.out.println("Unit price :"+unit);
                System.out.println("Total :"+total);
                return(total);

        }
}
class product2 implements bill{
```

```
        int id = 102,quantity= 1,unit=100,total=0;
                String name="B";

        public int productdetails()
        {
                total = quantity * unit;
                System.out.println("Product Id :"+id);
                System.out.println("Name :"+name);
                System.out.println("Quantity :"+quantity);
                System.out.println("Unit price :"+unit);
                System.out.println("Total :"+total);
                return(total);

        }
}
public class productbill
{
         public static void main(String[] args)
         {
                 product1 p1 = new product1();
                 product2 p2 = new product2();
                int t1= p1.productdetails();
                int t2= p2.productdetails();
                int t3=t1+t2;

        System.out.println("Net. Amount :"+t3);

         }
}
```

OUTPUT

```
D:\java_lab>javac productbill.java

D:\java_lab>java productbill
Product Id :101
Name :A
Quantity :2
Unit price :25
Total :50
Product Id :102
Name :B
Quantity :1
Unit price :100
Total :100
Net. Amount :150

D:\java_lab>
```

RESULT

The program has been executed and output verified.

Experiment No:17

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.

PROGRAM

```
//Area1.java

package Graphiccs;


interface Area1

{

	public void Rectangle();

	public void Triangle();

	public void Square();

	public void Circle();

	public void getRect();

	public void getTri();

	public void getSqr();

	public void getCrl();

}

//shapess.java

package Graphiccs;

import java.util.*;


public class shapess implements Area1

{

	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)
        {
            shapess o= new shapess();
            o.getRect();
            o.Rectangle();
            o.getTri();
            o.Triangle();
            o.getSqr();
            o.Square();
            o.getCrl();
            o.Circle();
        }
}
```

OUTPUT

```
D:\java_lab>javac -d . Area1.java

D:\java_lab>javac -d . shapess.java

D:\java_lab>java Graphiccs.shapess
Enter the length of the rectangle
4
Enter the breadth of the rectangle
5
Area of Rectangle is 20.0
Enter the height of the Triangle
2
Enter the base of the Triangle
6
Area of Triangle angle is 6.0
Enter the Side of the Square
3
Area of Square is 9.0
Enter the radius of the Circle
2
Area of Square is 12.56

D:\java_lab>
```

RESULT

The program has been executed and output verified.

Experiment No:18

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

PROGRAM

```
//operations.java

package Aarithmetic;


interface operations

{

        public void input();

        public void add();

        public void substract();

        public void multiply();

        public void division();


}

//basic.java

package Aarithmetic;

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();
}
```

}

OUTPUT

```
D:\java_lab>javac -d . operations.java

D:\java_lab>javac -d . basic.java

D:\java_lab>java Aarithmetic.basic
Enter two numbers
5
6
Sum is 11.0
Difference is -1.0
Product is 30.0
Quotient is 0.8333333333333334

D:\java_lab>
```

RESULT

The program has been executed and output verified.

Experiment No:19

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

PROGRAM

```
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 CheckLoginCredential {

 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

```
D:\java_lab>javac CheckLoginCredential.java

D:\java_lab>java CheckLoginCredential
Enter username :: vivin
Enter password :: 12343224
UsernameException: Username must be greater than 6 characters ???
        at CheckLoginCredential.main(CheckLoginCredential.java:31)
The finally statement is executed

D:\java_lab>
```

RESULT

The program has been executed and output verified.

Experiment No:20

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

PROGRAM

```
import java.util.Scanner;
import java.util.InputMismatchException;
public class TestDemo
{
	public static void main(String args[])
	{
		double total = 0, N, userInput;
		Scanner input = new Scanner(System.in);
		while (true)
		{
		 System.out.print("Enter how many numbers(N) 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

```
D:\java_lab>javac TestDemo.java

D:\java_lab>java TestDemo
Enter how many numbers(N) to calculate average:5
Enter number:3
Enter number:6
Enter number:3
Enter number:6
Enter number:32
Average: 10.0
```

RESULT

The program has been executed and output verified.

Experment:21

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

PROGRAM

```
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);
       System.out.println("Exiting from Thread A ...");
    }
}

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

 System.out.println("Enter the number of prime terms you want!");
    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++;
    }
}

//System.out.println("Exiting from Thread B ...");
```

```
}

public class Demonstration_111
{
   public static void main(String args[]) {
       ThreadA a = new ThreadA();
       ThreadB b = new ThreadB();
       a.start();
       b.start();
       System.out.println("... Multithreading is over ");
   }
}
```

OUTPUT

```
D:\java_lab>javac Demonstration_111.java

D:\java_lab>java Demonstration_111
... Multithreading is over
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
Exiting from Thread A ...
Enter the number of prime terms you want!
7
First 7 prime numbers are :-
2 3 5 7 11 13 17
D:\java_lab>
```

RESULT

The program has been executed and output verified.

Experiment No:22

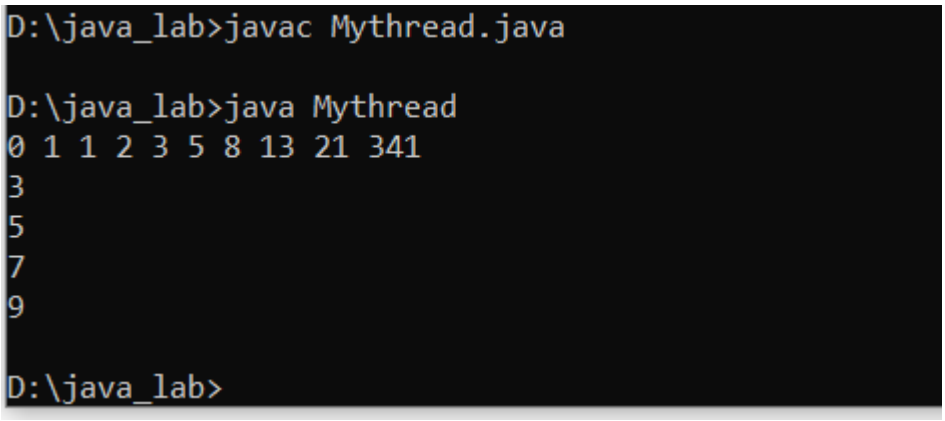
Define 2 classes; one for generating Fibonacci numbers and other for displaying even numbers in a given range. Implement using threads. (Runnable Interface)

PROGRAM

```
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(){
      for(int i=0;i<11;i++){
         if(i%2 == 1)
            System.out.println(i);
      }
   }
}


class Runnable1 implements Runnable{
   public void run(){
       int n1=0,n2=1,n3,i,count=10;
 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



RESULT

The program has been executed and output verified.

Experiment No:23

Producer/Consumer using ITC

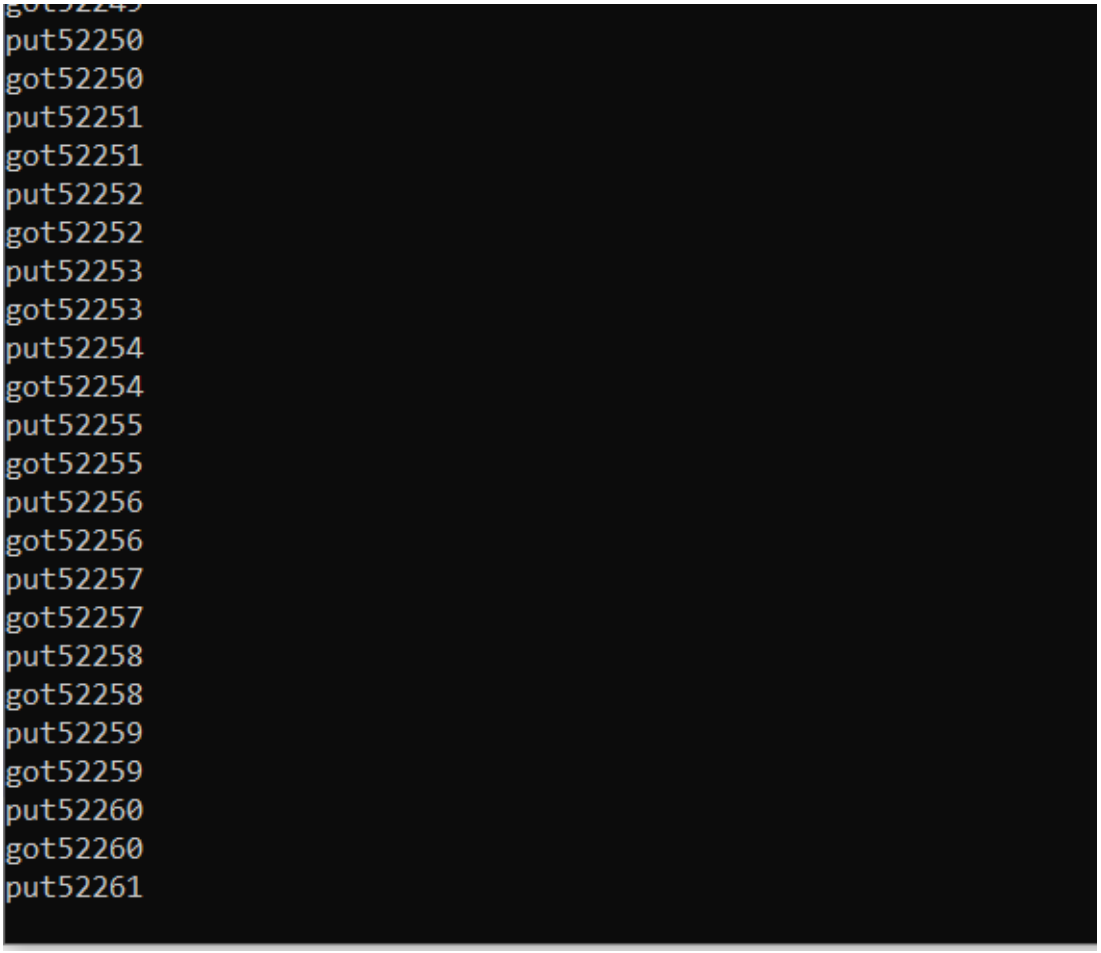
PROGRAM

```
import java.util.*;
class Q
{
	int n;
	boolean statusFlag=false;
	synchronized void put(int n)
	{
		try
			{
				while(statusFlag)
				{
					wait();
				}
			}
		catch(InterruptedException e){}
		this.n=n;
		System.out.println("Put :"+n);
		statusFlag=true;
		notify();
	}
	synchronized int get()
	{
		try{
			while(!statusFlag)
			{
				wait();
			}
```

```
            }
            catch(InterruptedException e){}
            statusFlag=false;
            System.out.println("Got :"+n);
            notify();
            return n;
        }
}
class Producer implements Runnable
{
        Q q;
        Producer(Q q)
        {
            this.q=q;
            new Thread(this, "Producer").start();
        }
        public void run()
        {
            int i=0;
            while(true)
            {
                q.put(i++);
            }
        }
}
class Consumer implements Runnable
{
        Q q;
        Consumer(Q q)
        {
```

```
		this.q=q;
		new Thread(this,"Consumer").start();
	}
	public void run()
	{
		while(true)
		{
			q.get();
		}
	}
}
public class D
{
	public static void main(String[] args)
	{
		Q q=new Q();
		Producer p=new Producer(q);
		Consumer c=new Consumer(q);
	}
}
```

OUTPUT



RESULT

The program has been executed and output verified.

Experiment No:24

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

PROGRAM

```
public class StackAsLinkedList {


   StackNode root;


   static class StackNode {
      int data;
      StackNode next;


      StackNode(int data) { this.data = data; }
   }


   public boolean isEmpty()
   {
      if (root == null) {
         return true;
      }
      else
         return false;
   }


   public void push(int data)
   {
      StackNode newNode = new StackNode(data);


      if (root == null) {
         root = newNode;
```

```
    }
    else {
        StackNode temp = root;
        root = newNode;
        newNode.next = temp;
    }
    System.out.println(data + " pushed to stack");
}

public int pop()
{
    int popped = Integer.MIN_VALUE;
    if (root == null) {
        System.out.println("Stack is Empty");
    }
    else {
        popped = root.data;
        root = root.next;
    }
    return popped;
}

public int peek()
{
    if (root == null) {
        System.out.println("Stack is empty");
        return Integer.MIN_VALUE;
    }
    else {
        return root.data;
```

```
    }
  }

  // Driver code
  public static void main(String[] args)
  {

    StackAsLinkedList sll = new StackAsLinkedList();

    sll.push(10);
    sll.push(20);
    sll.push(30);

    System.out.println(sll.pop()
                + " popped from stack");

    System.out.println("Top element is " + sll.peek());
  }
}
```

OUTPUT

```
D:\java_lab>javac StackAsLinkedList.java

D:\java_lab>java StackAsLinkedList
10 pushed to stack
20 pushed to stack
30 pushed to stack
30 popped from stack
Top element is 20

D:\java_lab>
```

RESULT

The program has been executed and output verified.

Experiment No:25

Using generic method perform Bubble sort.

PROGRAM

```
public class BubbleSort {
  static void bubbleSort(int[] arr) {
    int n = arr.length;
    int temp = 0;

    for(int i = 0; i < n; i++) {
      for(int j=1; j < (n-i); j++) {
        if(arr[j-1] > arr[j]) {
          temp = arr[j-1];
          arr[j-1] = arr[j];
          arr[j] = temp;
        }
      }
    }
  }
  public static void main(String[] args) {
    int arr[] = { 1, 6, -2, 6, -4, 8, 5, -7, -9, 4 };
    System.out.println("Array Before Bubble Sort");

    for(int i = 0; i < arr.length; i++) {
      System.out.print(arr[i] + " ");
    }
    System.out.println();
    bubbleSort(arr);
    System.out.println("Array After Bubble Sort");
```

```
    for(int i = 0; i < arr.length; i++) {
      System.out.print(arr[i] + " ");
    }
  }
}
```

OUTPUT

```
D:\java_lab>javac BubbleSort.java

D:\java_lab>java BubbleSort
Array Before Bubble Sort
1 6 -2 6 -4 8 5 -7 -9 4
Array After Bubble Sort
-9 -7 -4 -2 1 4 5 6 6 8
D:\java_lab>
```

RESULT

The program has been executed and output verified.

Experiment No:26

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

PROGRAM

```
import java.util.*;

public class arraylist {

   public static void main(String[] args)

   {

      ArrayList<String> list1=new ArrayList<String>();

      list1.add("jan");

      list1.add("feb");

      list1.add("march");

      list1.add("may");

      list1.add("jun");


      System.out.println(list1);


      list1.add("august");


      System.out.println(list1);

   }


}
```

OUTPUT

```
D:\java_lab>javac arraylist.java

D:\java_lab>java arraylist
[jan, feb, march, may, jun]
[jan, feb, march, may, jun, august]

D:\java_lab>
```

RESULT

The program has been executed and output verified.

Experiment No:27

Program to remove all the elements from a linked list

PROGRAM

```
import java.util.*;
 public class removelink
 {
 public static void main(String[] args)
{
   // create an empty linked list
    LinkedList<String> l_list = new LinkedList<String>();


  // use add() method to add values in the linked list
        l_list.add("violet");
        l_list.add("Green");
        l_list.add("Black");
        l_list.add("Pink");
        l_list.add("blue");
    // print the list
  System.out.println("The Original linked list: " + l_list);
 // Removing all the elements from the linked list
   l_list.clear();
   System.out.println("The New linked list: " + l_list);
 }
}
```

OUTPUT

```
D:\java_lab>javac removelink.java

D:\java_lab>java removelink
The Original linked list: [violet, Green, Black, Pink, blue]
The New linked list: []

D:\java_lab>
```

RESULT

The program has been executed and output verified.

Experiment No:28

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

PROGRAM

```
import java.util.*;
class PriorityQueue1
{
	public static void main(String args[])
	{
		PriorityQueue<String> queue=new PriorityQueue<String>();
		queue.add("Amit");
		queue.add("Vijay");
		queue.add("Karan");
		queue.add("Jai");
		queue.add("Rahul");
		System.out.println("head:"+queue.element());
		System.out.println("head:"+queue.peek());
		System.out.println("iterating the queue elements:");
		Iterator itr=queue.iterator();
		while(itr.hasNext())
		{
			System.out.println(itr.next());
		}
		queue.remove();
		queue.poll();
		System.out.println("after removing two elements:");
		Iterator<String> itr2=queue.iterator();
		while(itr2.hasNext())
		{
			System.out.println(itr2.next());
```

```
            }
        }
}
```

OUTPUT

```
D:\java_lab>javac PriorityQueue1.java

D:\java_lab>java PriorityQueue1
head:Amit
head:Amit
iterating the queue elements:
Amit
Jai
Karan
Vijay
Rahul
after removing two elements:
Karan
Rahul
Vijay

D:\java_lab>
```

RESULT

The program has been executed and output verified.

Experiment No:29

Program to demonstrate the addition and deletion of elements in deque

PROGRAM

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

      dq.add("Element 1 (Tail)");
      dq.addFirst("Element 2 (Head)");
      dq.addLast("Element 3 (Tail)");
      dq.push("Element 4 (Head)");
      dq.offer("Element 5 (Tail)");
      dq.offerFirst("Element 6 (Head)");

      System.out.println(dq + "\n");

      dq.removeFirst();
      dq.removeLast();
   }

}
```

OUTPUT

```
D:\java_lab>javac deque.java

D:\java_lab>java deque
[Element 6 (Head), Element 4 (Head), Element 2 (Head), Element 1 (Tail), Element 3 (Tail), Element 5 (Tail)]


D:\java_lab>
```

RESULT

The program has been executed and output verified.

Experiment No:30

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

PROGRAM

```
import java.util.*;

public class hashmap {

   public static void main(String[] args) {

      Map<String, Integer> hmap=new HashMap<String, Integer>();

      hmap.put("Anu ",new Integer(1));

      hmap.put("sinu",new Integer(2));

      hmap.put("Jinu",new Integer(3));


      for(Map.Entry<String,Integer> me : hmap.entrySet())

      {

         System.out.print(me.getKey() +" : ");

         System.out.println(me.getValue());

      }

   }


}
```

OUTPUT

```
D:\java_lab>javac hashmap.java

D:\java_lab>java hashmap
Jinu : 3
Anu  : 1
sinu : 2

D:\java_lab>
```

RESULT

The program has been executed and output verified.

Experiment No:31

Program to Convert HashMap to TreeMap

PROGRAM

```
import java.util.*;
import java.util.stream.*;
public class HT
{
  public static void main(String args[])
 {
    Map<String, String> map = new HashMap<>();
    map.put("1", "One");
    map.put("2", "Two");
    map.put("3", "Three");
    map.put("4", "Four");
    map.put("5", "Five");
    map.put("6", "Six");
    map.put("7", "Seven");
    map.put("8", "Eight");
    map.put("9", "Nine");
    System.out.println("HashMap = " + map);
    Map<String, String> treeMap = new TreeMap<>();
    treeMap.putAll(map);
    System.out.println("TreeMap (HashMap to TreeMap) " + treeMap);
  }
}
```

OUTPUT

```
D:\java_lab>javac HT.java

D:\java_lab>java HT
HashMap = {1=One, 2=Two, 3=Three, 4=Four, 5=Five, 6=Six, 7=Seven, 8=Eight, 9=Nine}
TreeMap (HashMap to TreeMap) {1=One, 2=Two, 3=Three, 4=Four, 5=Five, 6=Six, 7=Seven, 8=Eight, 9=Nine}

D:\java_lab>
```

RESULT

The program has been executed and output verified.

Experiment No:32

Program to draw Circle, Rectangle, Line in Applet

PROGRAM

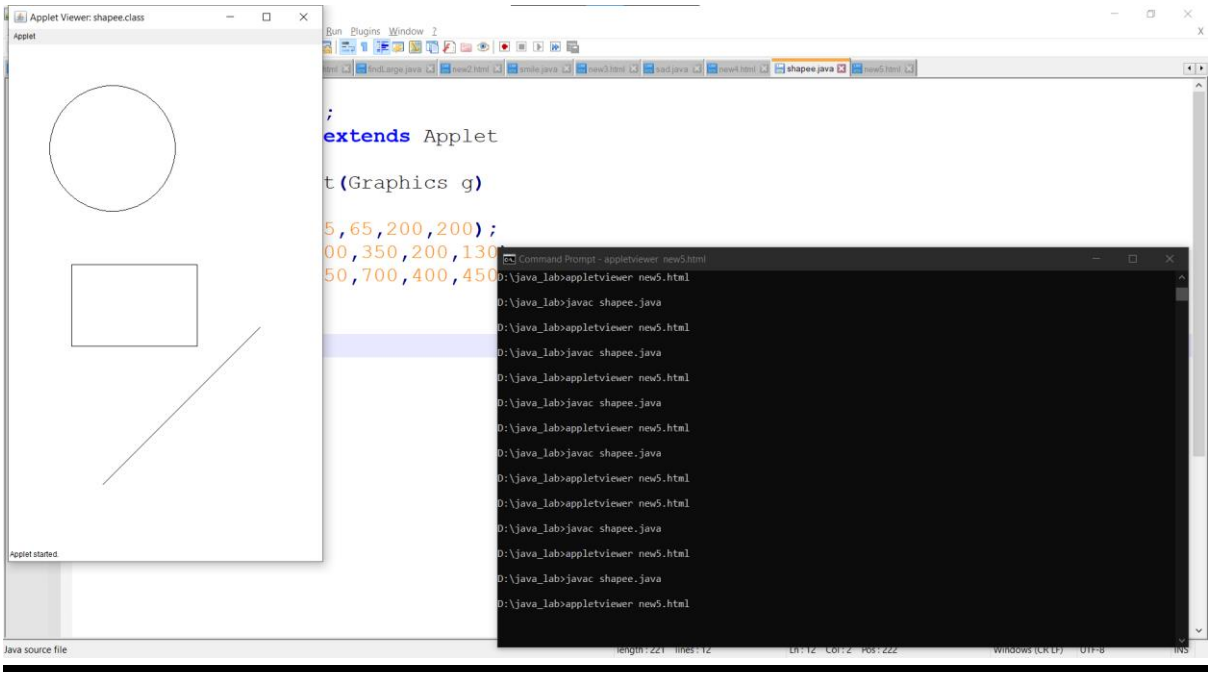
//shape.java

```
import java.awt.*;
import java.applet.*;
public class shapee extends Applet
{
	public void paint(Graphics g)
	{
		g.drawOval(65,65,200,200);
		g.drawRect(100,350,200,130);
		g.drawLine(150,700,400,450);

	}
}
```

//new5.html

```
<html>
	<head>
	</head>
	<body>
		<div align="center">
			<applet code="shapee.class" height="500" width="800">
			</applet>
		</div>
	</body>
</html>
```

OUTPUT



RESULT

The program has been executed and output verified.

Experiment No:33

Program to find maximum of three numbers using AWT.

PROGRAM

//findLarge.java

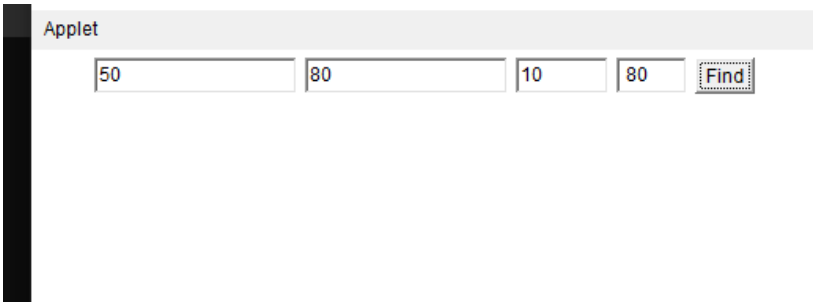
```
import java.awt.*;
import java.applet.*;
import java.awt.event.*;
public class findLarge extends Applet implements ActionListener
{
	TextField t1,t2,t3,t4;
	Button b1;
	public void init()
	{
		t1=new TextField(15);
		t1.setBounds(100,25,50,20);
		t2=new TextField(15);
		t2.setBounds(100,25,50,20);
		t3=new TextField(5);
		t3.setBounds(100,25,50,20);
		t4=new TextField("ANS");
		t4.setBounds(175,50,50,20);
		b1=new Button("Find");
		b1.setBounds(175,60,50,40);
		add(t1);
		add(t2);
		add(t3);
		add(t4);
		add(b1);
		b1.addActionListener(this);
```

```
	}

	public void actionPerformed(ActionEvent e)
	{
		int i,j,k;
		i=Integer.parseInt(t1.getText());
		j=Integer.parseInt(t2.getText());
		k=Integer.parseInt(t3.getText());
		if(i<j && k<j)
				t4.setText(" "+j);
		else if(i<k)
				t4.setText(" "+k);
		else
			t4.setText(" "+i);
	}
}
```

//new.html

```
<html>
	<head>
	</head>
	<body>
		<div align="center">
			<applet code="findLarge.class" height="500" width="800">
			</applet>
		</div>
	</body>
</html>
```

OUTPUT



RESULT

The program has been executed and output verified.

Experiment No:34

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

PROGRAM

//marks.java

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
public class marks extends Applet implements ActionListener {
public int per =0;
Label l1 = new Label("Enter Marks of English: ");
 Label l2 = new Label("Enter Marks of Malayalam: ");
 Label l3 = new Label("Enter Marks of Hindi: ");
 Label l4 = new Label("Enter Marks of Mathematics: ");
 Label l5 = new Label("Enter Marks of Physics: ");
 Label l6 = new Label("Total Percentage: ");

 TextField t1 = new TextField(10);
 TextField t2 = new TextField(10);
 TextField t3 = new TextField(10);
 TextField t4 = new TextField(10);
 TextField t5 = new TextField(10);
 TextField t6 = new TextField(10);

 Button b1 = new Button("CALCULATE PERCENTAGE");

 public marks()
 {
l1.setBounds(50, 100, 280, 20);
l2.setBounds(50, 150, 280, 20);
l3.setBounds(50, 200, 280, 20);
```

```
l4.setBounds(50, 250, 280, 20);
l5.setBounds(50, 300, 280, 20);
l6.setBounds(50, 350, 280, 20);

t1.setBounds(200, 100, 300, 20);
t2.setBounds(200, 150, 300, 20);
t3.setBounds(200, 200, 300, 20);
t4.setBounds(200, 250, 300, 20);
t5.setBounds(200, 300, 300, 20);
t6.setBounds(200, 350, 300, 20);

b1.setBounds(200,400, 200, 20);
GridLayout g1 = new GridLayout(20, 2, 5, 5);
setLayout(g1);
add(l1);
add(t1);
add(l2);
add(t2);
add(l3);
add(t3);
add(l4);
add(t4);
add(l5);
add(t5);
add(l6);
add(t6);
add(b1);
b1.addActionListener(this);
 }
@Override
public void actionPerformed(ActionEvent e) {
```

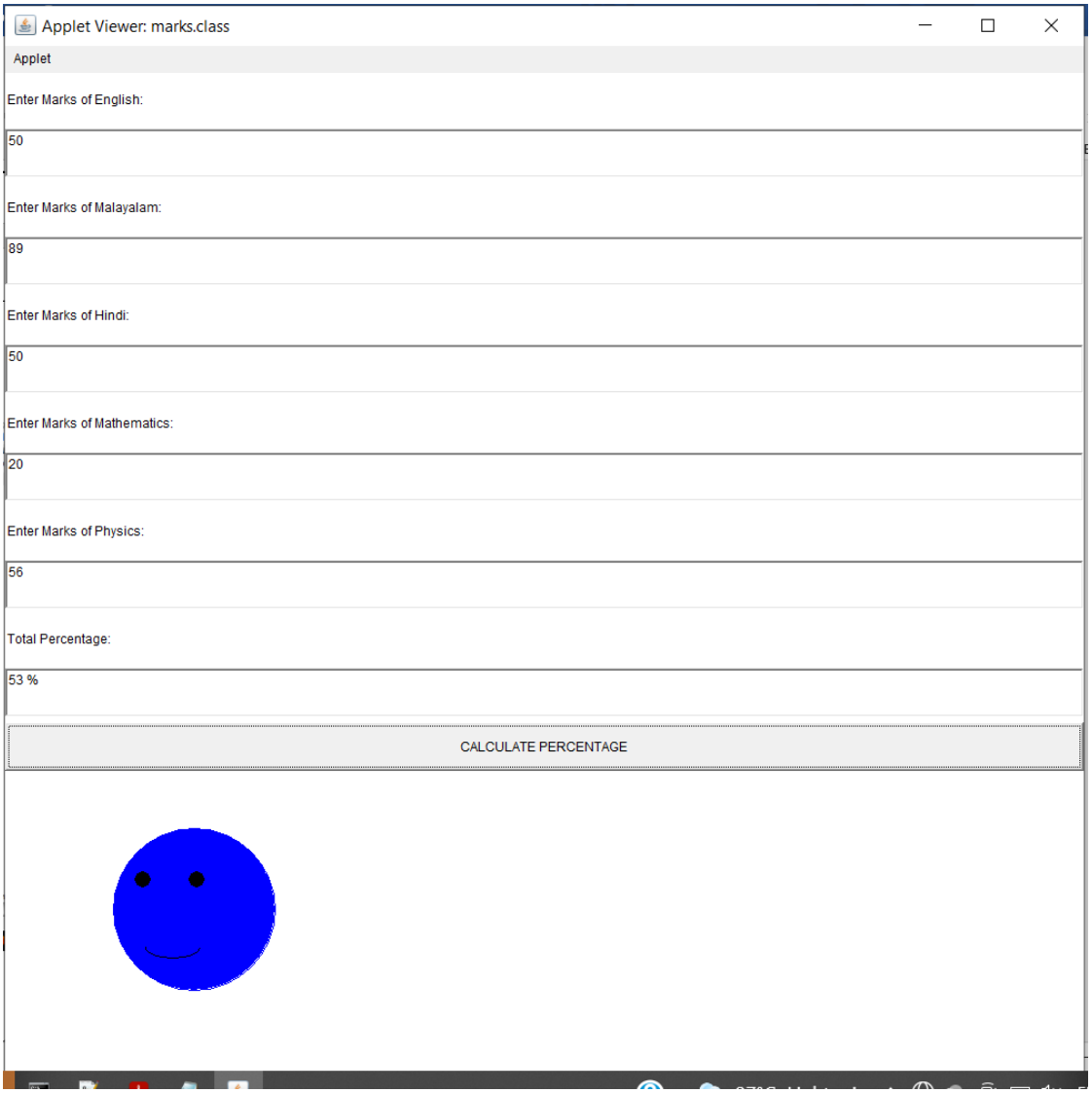
```
// TODO Auto-generated method stub
int m1 = Integer.parseInt(t1.getText());
 int m2= Integer.parseInt(t2.getText());
 int m3= Integer.parseInt(t3.getText());
 int m4= Integer.parseInt(t4.getText());
 int m5= Integer.parseInt(t5.getText());

 if(e.getSource()==b1)
 {
 int add=m1+m2+m3+m4+m5;
 per=add/5;
 t6.setText(String.valueOf(per)+" %");

 repaint();
 }

}
public void paint(Graphics g)
 {
if(per>=50)
{
g.setColor(Color.blue);
g.drawOval(100, 700, 150, 150);
g.fillOval(100, 700, 150, 150);
g.setColor(Color.BLACK);
g.fillOval(120, 740, 15, 15);
g.fillOval(170, 740, 15, 15);
g.drawArc(130, 800, 50, 20, 180, 180);
}
else if(per>0 && per<50)
{
```

```
g.setColor(Color.blue);
g.drawOval(100, 700, 150, 150);
g.fillOval(100, 700, 150, 150);
 g.setColor(Color.BLACK);
 g.fillOval(120, 740, 15, 15);
 g.fillOval(170, 740, 15, 15);
g.drawArc(130,820,50,20,0,180);
}
 }
public static void main(String args[]) {
 new marks();
 }
}
```

//ni.html

```
<html>
        <head>
        </head>
        <body>
        <div align="center">
                <applet code="marks.class"width="1000"height="1000">
        </applet>
        </div>
</body>
</html>
```

OUTPUT


Applet Viewer: marks.class
Applet
Enter Marks of English:
50
Enter Marks of Malayalam:
89
Enter Marks of Hindi:
50
Enter Marks of Mathematics:
20
Enter Marks of Physics:
56
Total Percentage:
53 %
CALCULATE PERCENTAGE

Applet Viewer: marks.class

Applet

Enter Marks of English:

10

Enter Marks of Malayalam:

20

Enter Marks of Hindi:

30

Enter Marks of Mathematics:

23

Enter Marks of Physics:

20

Total Percentage:

20 %

CALCULATE PERCENTAGE

RESULT

The program has been executed and output verified.

Experiment No:35

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

PROGRAM

//house.java

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


public class house extends  Applet implements MouseListener,Runnable
{
        private Color textColor = Color.BLUE;
public void paint(Graphics g)
{ int [] x = {150, 300, 225};
int [] y = {150, 150, 25};
g.drawRect(150, 150, 150, 200); //House
g.drawRect(200, 200, 50, 150);
g.setColor(Color.blue);
g.setColor(textColor);
g.fillRect(200, 200, 50, 150); // Door
g.setColor(Color.black);
g.fillPolygon(x, y, 3); // Roof
}
public void init()
   {
      this.setSize(200,200);
      addMouseListener(this);
   }
```

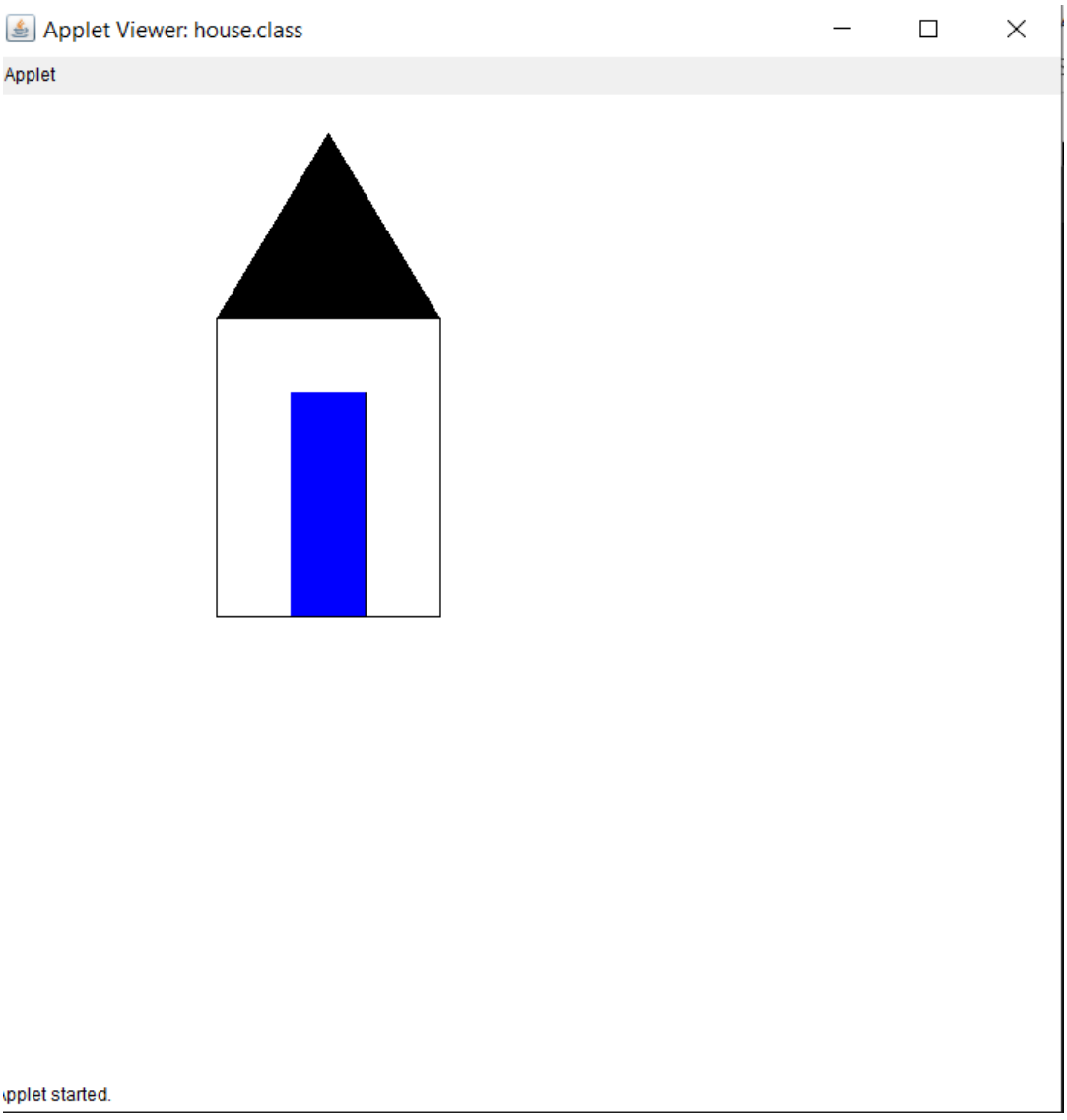
```
public void run()
{
   while(true)
   {
      repaint();
      try
      {
        Thread.sleep(17);
      }
      catch (InterruptedException e)
      {
         e.printStackTrace();
      }

   }
}
public void mouseClicked(MouseEvent e)
{
   int x=e.getX(),y=e.getY();
   if(x>=60 && x<=120 && y>=80 && y<=95)
      textColor=Color.BLUE;
   else
      textColor=Color.RED;
      repaint();
      System.out.println("Mouse Position: X= "+x+"Y"+y);
}
public void mousePressed(MouseEvent e){}
public void mouseReleased(MouseEvent e){}
public void mouseEntered(MouseEvent e){}
public void mouseExited(MouseEvent e){}
```

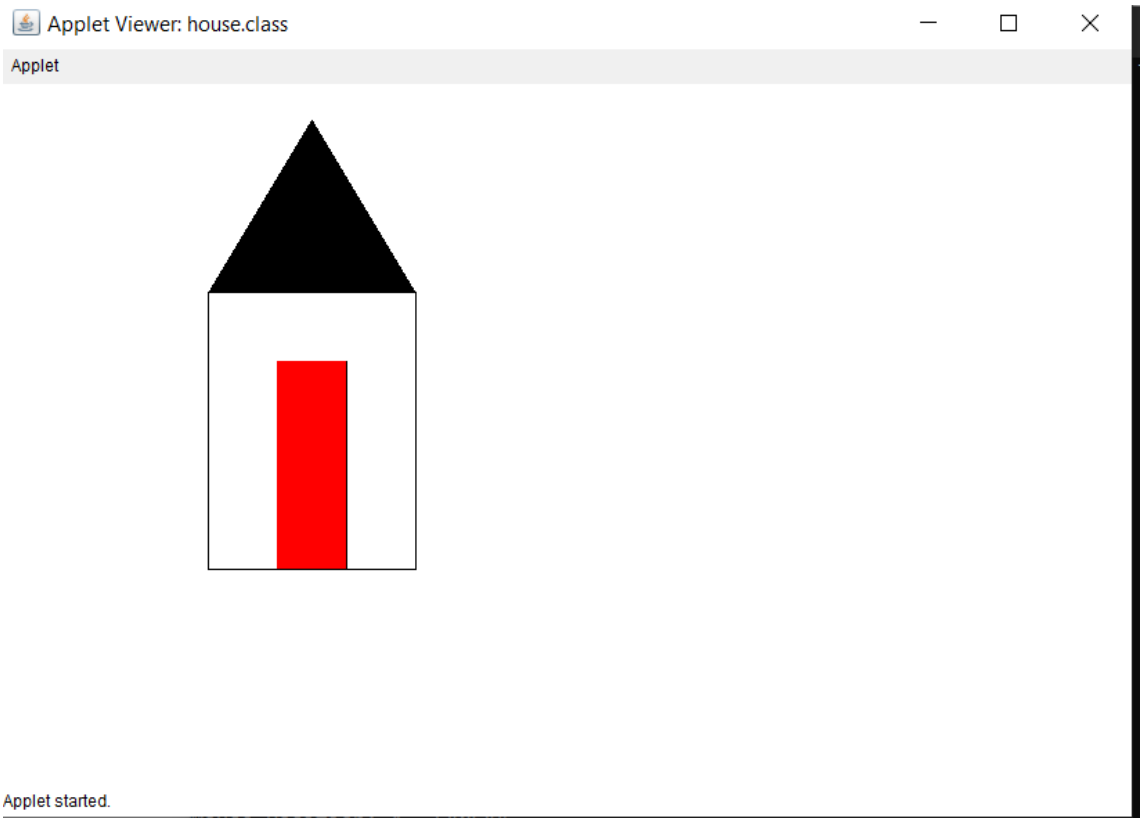
```
}
```

//house.html

```
<html>
	<head>
	</head>
<body>
	<div align="center">
		<applet code="house.class"width="800"height="500"></applet>
	</div>
</body>
</html>
```

OUTPUT





RESULT

The program has been executed and output verified.

Experiment No:36

Implement a simple calculator using AWT components.

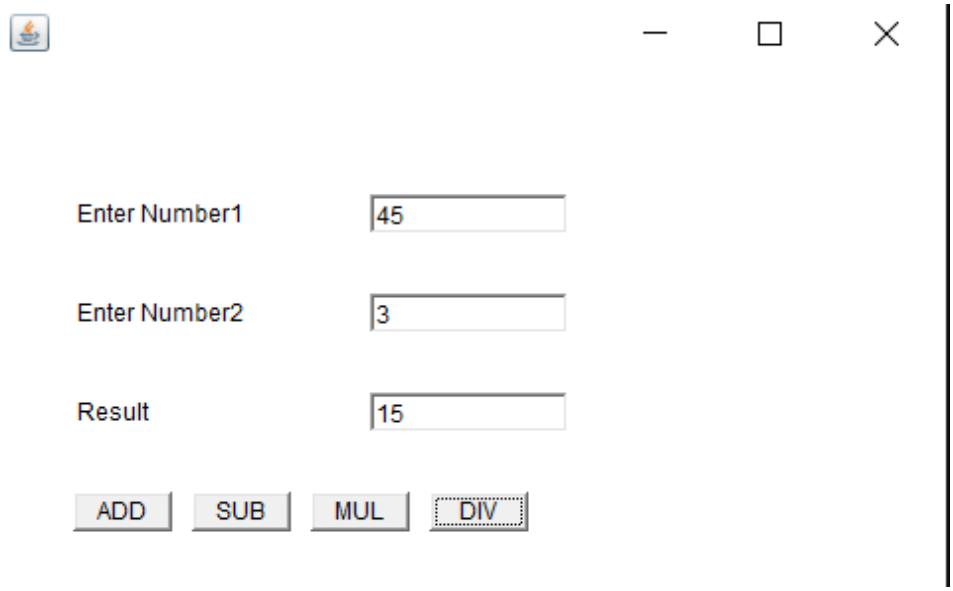
PROGRAM

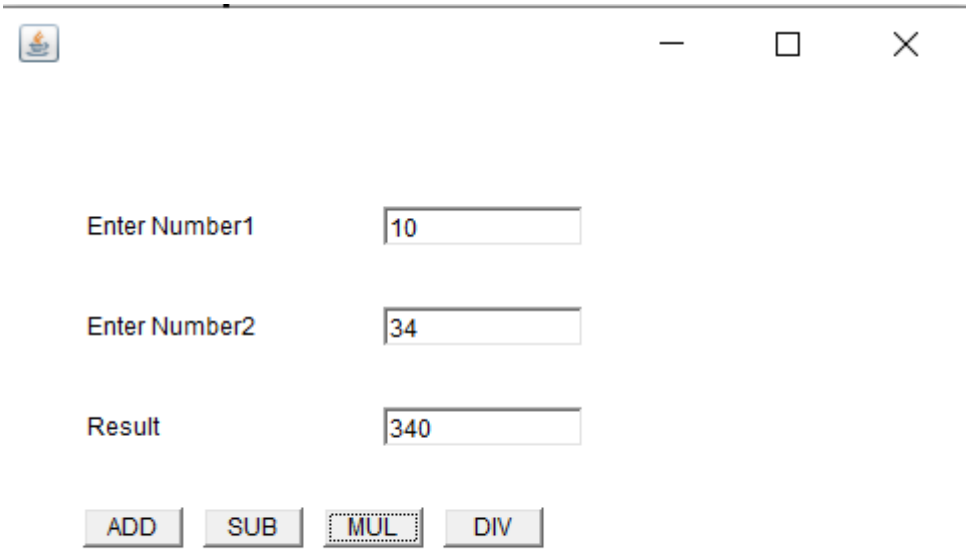
```
import java.awt.*;
import java.awt.event.*;
class calc implements ActionListener
{
	Frame f=new Frame();
	Label l1=new Label("Enter Number1");
	Label l2= new Label("Enter Number2");
	Label l3=new Label("Result");
	TextField t1=new TextField();
	TextField t2=new TextField();
	TextField t3=new TextField();
	Button b1=new Button("ADD");
	Button b2=new Button("SUB");
	Button b3=new Button("MUL");
	Button b4=new Button("DIV");
	calc()
	{
		l1.setBounds(50,100,100,20);
		l2.setBounds(50,150,100,20);
		l3.setBounds(50,200,100,20);
		t1.setBounds(200,100,100,20);
		t2.setBounds(200,150,100,20);
		t3.setBounds(200,200,100,20);
		b1.setBounds(50,250,50,20);
		b2.setBounds(110,250,50,20);
		b3.setBounds(170,250,50,20);
		b4.setBounds(230,250,50,20);
		f.add(l1);
```

```
        f.add(l2);
        f.add(l3);
        f.add(t1);
        f.add(t2);
        f.add(t3);
        f.add(b1);
        f.add(b2);
        f.add(b3);
        f.add(b4);
        b1.addActionListener(this);
        b2.addActionListener(this);
        b3.addActionListener(this);
        b4.addActionListener(this);
        f.setLayout(null);
        f.setVisible(true);
        f.setSize(500,500);
}
public void actionPerformed(ActionEvent e)
{
        int i=Integer.parseInt(t1.getText());
        int j=Integer.parseInt(t2.getText());
        if(e.getSource()==b1)
        {
                t3.setText(String.valueOf(i+j));
        }
        if(e.getSource()==b2)
        {
                t3.setText(String.valueOf(i-j));
        }
        if(e.getSource()==b3)
        {
```

```
                t3.setText(String.valueOf(i*j));
            }
            if(e.getSource()==b4)
            {
                t3.setText(String.valueOf(i/j));
            }
        }
        public static void main(String args[])
        {
            new calc();
        }
}
```

OUTPUT





RESULT

The program has been executed and output verified.

Experiment No:37

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.

PROGRAM

//figchoice.java

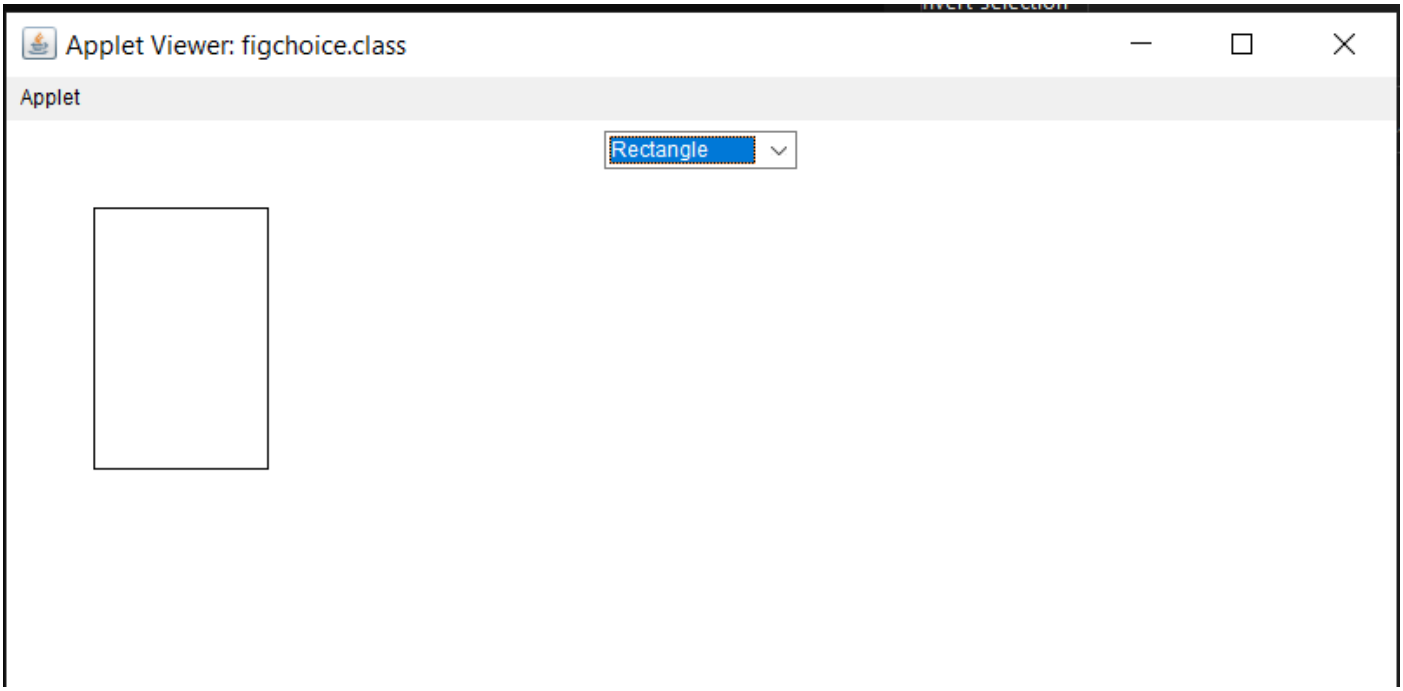
```
import java.applet.*;
import java.awt.*;
import java.awt.Graphics;
import java.awt.event.*;
public class figchoice extends Applet implements ItemListener {
 Choice ch;
 int x1[]= {50,120,220,20};
int y1[]= {50,120,20,20};
int n=4;
 int Selection;
 public void init()
 {
 ch = new Choice();
 ch.addItem("Select a Shape");
 ch.addItem("Rectangle");
 ch.addItem("Triangle");
 ch.addItem("Square");
 ch.addItem("Circle");
 add(ch);
 ch.addItemListener(this);
 }
public void itemStateChanged (ItemEvent e)
 {
```

```
 Selection = ch.getSelectedIndex();
 repaint();
 }
public void paint(Graphics g)
{
 super.paint(g);
 if (Selection == 1)
 {    g.drawRect(50,50,100,150);     }
 if (Selection == 2)
 {    g.drawPolygon(x1,y1,n);    }
 if (Selection == 3)
 {    g.drawRect(50,50,100,100);           }
 if (Selection == 4)
 {
 g.drawOval(70,30,100,100);
 } } }
```

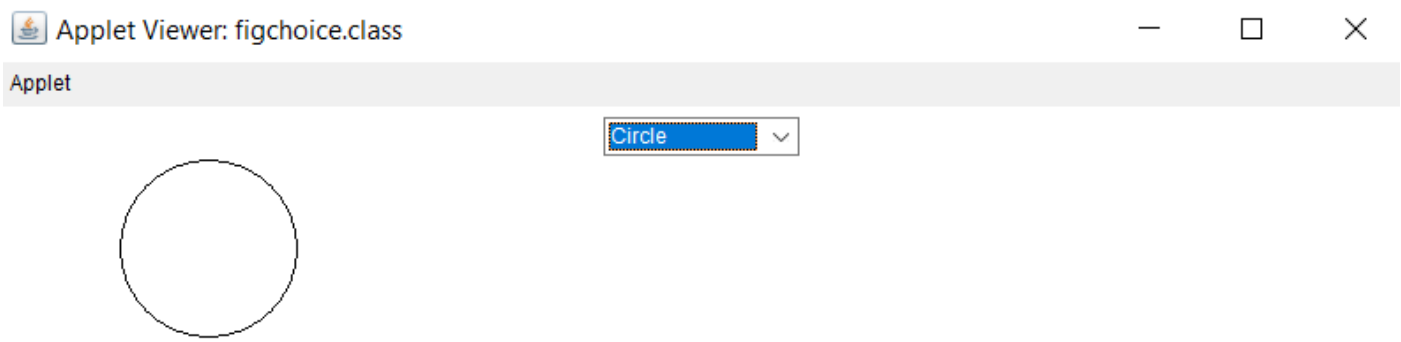
//fig.html

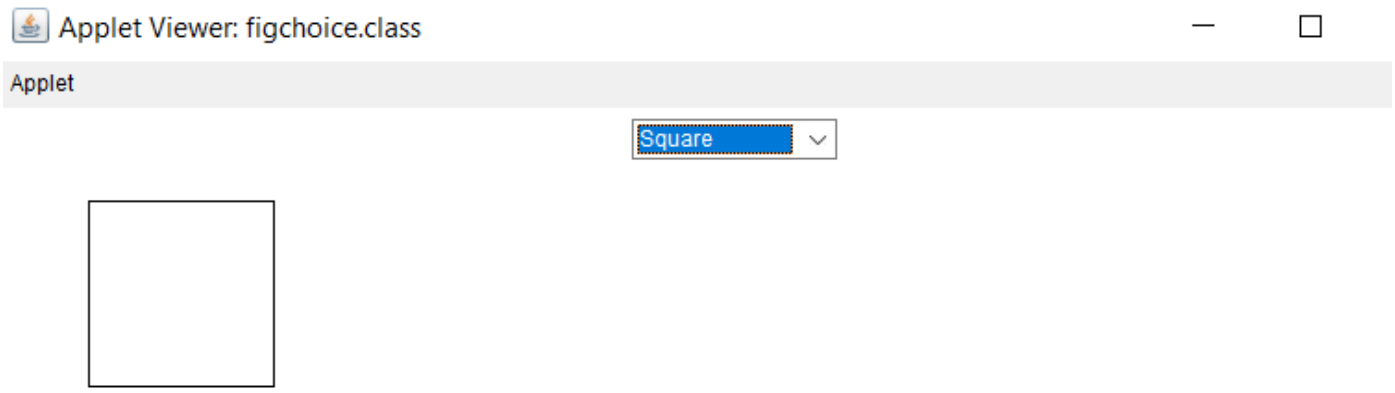
```
<html><head>
</head>
<body>
<div align="center">
<applet code="figchoice.class"width="800"height="500">
</applet>
</div>
</body>
</html>
```

OUTPUT


Applet Viewer: figchoice.class
Applet
Rectangle


Applet Viewer: figchoice.class
Applet
Triangle





RESULT

The program has been executed and output verified.

Experiment No:38

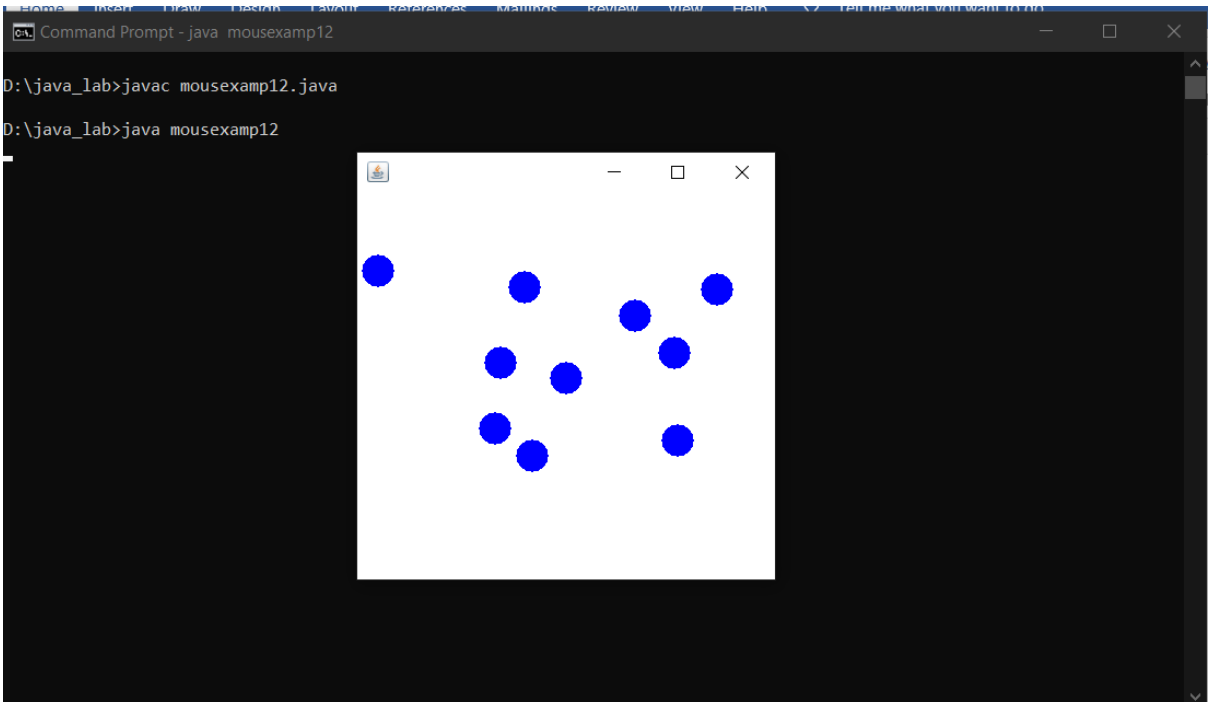
Develop a program to handle all mouse events

PROGRAM

```
import java.awt.*;
import java.awt.event.*;
public class mousexamp12 extends Frame implements MouseListener
{
        mousexamp12()
        {
        addMouseListener(this);
        setSize(400,400);
        setLayout(null);
        setVisible(true);
        }
        public void mouseClicked(MouseEvent e)
        {
        Graphics g=getGraphics();
        g.setColor(Color.blue);
        g.fillOval(e.getX(),e.getY(),30,30);
        }
public void mouseEntered(MouseEvent e)
{
}
public void mouseExited(MouseEvent e)
{
}
public void mousePressed(MouseEvent e)
{
}
public void mouseReleased(MouseEvent e){
}
```

```
public static void main(String args[])
{
new mousexamp12();
}
}
```

OUTPUT



RESULT

The program has been executed and output verified.

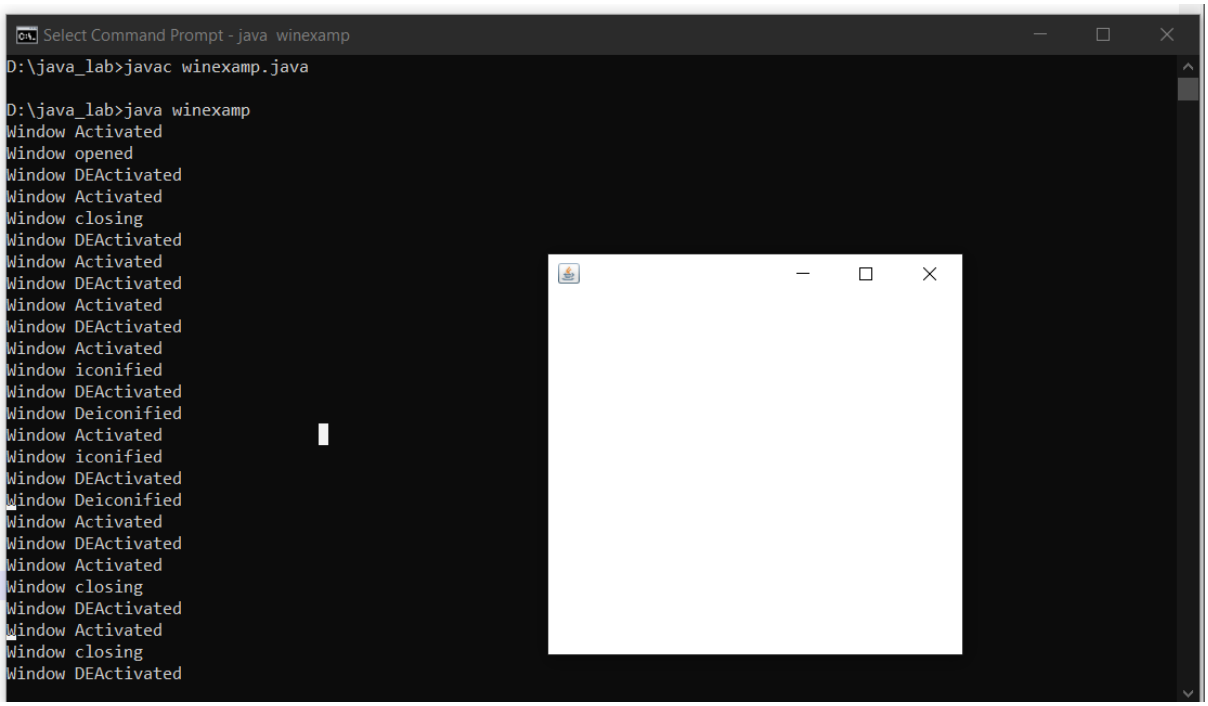
Experiment No:39

Develop a program to handle all window events

PROGRAM

```
import java.awt.*;
import java.awt.event.WindowEvent;
import java.awt.event.WindowListener;
public class winexamp extends Frame implements WindowListener
{
winexamp()
{
addWindowListener(this);
setSize(400,400);
setLayout(null);
setVisible(true);
}
public static void main(String args[])
{
new winexamp();
}
public void windowActivated(WindowEvent arg0)
{
System.out.println("Window Activated");
}
public void windowClosed(WindowEvent args0)
{
System.out.println("Window closed");
}
public void windowClosing(WindowEvent arg0)
{
System.out.println("Window closing");
}
public void windowDeactivated(WindowEvent arg0)
{
System.out.println("Window DEActivated");
}
public void windowDeiconified(WindowEvent arg0)
{
System.out.println("Window Deiconified");
}
public void windowIconified(WindowEvent arg0)
{
System.out.println("Window iconified");
}
public void windowOpened(WindowEvent arg0)
{
System.out.println("Window opened");
}
}
```

OUTPUT



RESULT

The program has been executed and output verified.

Experiment No:40

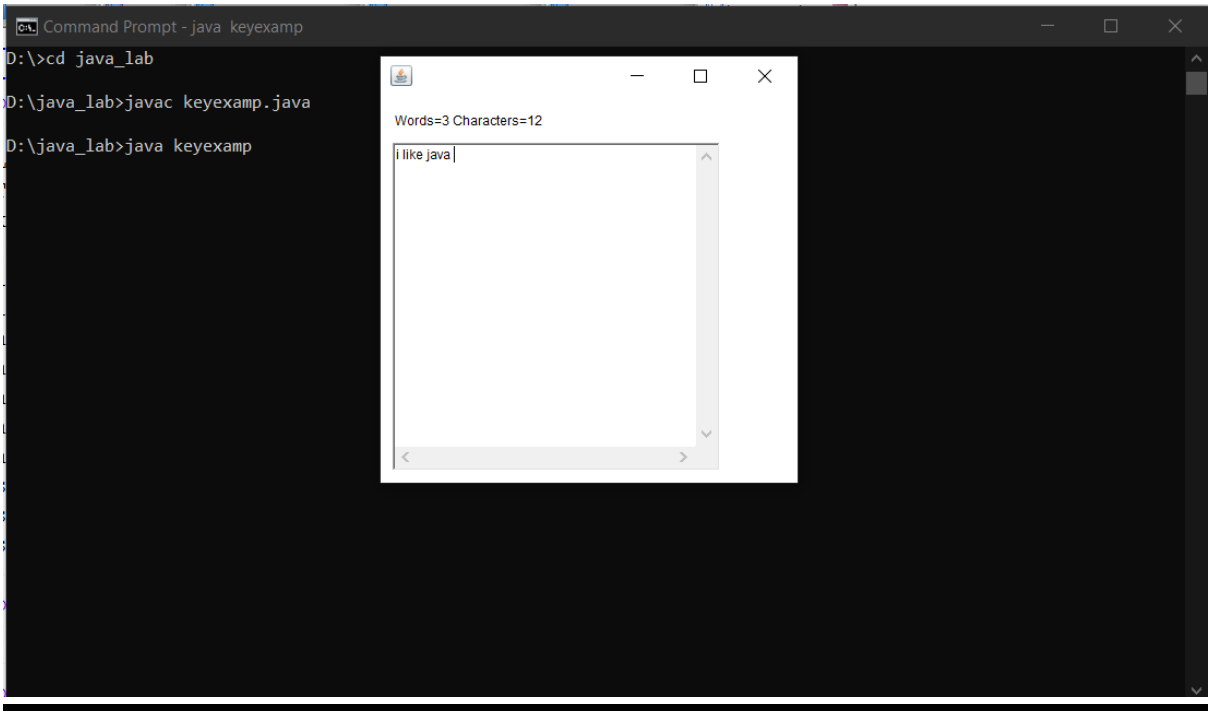
Develop a program to handle Key events.

PROGRAM

```
import java.awt.*;
import java.awt.event.*;
public class keyexamp extends Frame implements KeyListener
{
Label l;
TextArea a;
keyexamp()
{
l=new Label();
l.setBounds(20,50,200,20);
a=new TextArea();
a.setBounds(20,80,300,300);
a.addKeyListener(this);
add(l);
add(a);
setSize(400,400);
setLayout(null);
setVisible(true);
}
public void keyPressed(KeyEvent e)
{
}
public void keyReleased(KeyEvent e)
{
String t=a.getText();
String w[]=t.split("\\s");
l.setText("Words="+w.length+" Characters="+t.length());
}
```

```
public void keyTyped(KeyEvent e)
{}
public static void main(String args[])
{
new keyexamp();
}
}
```

OUTPUT



RESULT

The program has been executed and output verified.

Experiment:41

Program to list the sub directories and files in a given directory and also search for a file name.

PROGRAM

```
import java.io.File;
import java.util.*;
import java.io.*;
public class p1 {
 public static final String RED="\033[0;31m";
 public static final String RESET="\033[0m";
 static void RecursivePrint(File[] arr, int index, int level, String search
for) {
 // exit condition
 if (index == arr.length)
 return;
 // space for internbal level
 for (int i = 0; i < level; i++)
 System.out.print("\t");
 if(arr[index].getName().toLowerCase().contains(searchfor))
 System.out.print(RED);
 else
 System.out.print(RESET);
 // for files
 if (arr[index].isFile())
 System.out.println(arr[index].getName());
 else if (arr[index].isDirectory()) {
 System.out.println("[" + arr[index].getName() + "]");
 RecursivePrint(arr[index].listFiles(), 0, level + 1, searchfor);
 }
 RecursivePrint(arr, ++index, level, searchfor);
 }
 public static void main(String[] args) {
 Scanner scan = new Scanner(System.in);
 System.out.println("Enter the directory path");
```

```
        String maindirpath = scan.nextLine();
        System.out.println("Enter the file/directory name to search");
        String searchfor = scan.nextLine();
        File maindir = new File(maindirpath);
        if (maindir.exists() && maindir.isDirectory()) {
        File arr[] = maindir.listFiles();

       System.out.println("################################################
       ##
       ###");
        System.out.println("Files from main directory" + maindir);

       System.out.println("################################################
       ##
       ###");
       RecursivePrint(arr, 0, 0, searchfor.toLowerCase()); // array,index
       ,level,search
        }
        }
}
```

OUTPUT

```
D:\java_lab>javac p1.java

D:\java_lab>java p1
Enter the directory path
D:\java_lab\neww
Enter the file/directory name to search
ArrayListex.java
################################################
Files from main directoryD:\java_lab\neww
################################################
←[0mArrayListex.class
←[0;31mArrayListex.java

D:\java_lab>
```

RESULT

The program has been executed and output verified.

Experiment No:42

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

PROGRAM

```
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.*;
import java.util.*;
import java.io.File;
class read {
 public static void main(String[] args) {
 String var = "";
 Scanner scan = new Scanner(System.in);
 System.out.println("Enter the text to create file : type exit to stop"
);
 while (!var.endsWith("exit\n"))
 var = var + scan.nextLine()+"\n";
 try {
 File file = new File("output.txt");
 FileWriter fw = new FileWriter(file);
 fw.write(var);
 fw.close();
 System.out.println("Reading File content");
 FileReader fr = new FileReader("output.txt");
 String str = "";
 int i;
 while ((i = fr.read()) != -1) {
 // Storing every character in the string
 str += (char) i;
 }
 System.out.println(str);
 fr.close();
 } catch (IOException e) {
```

```
        System.out.println("There are some exception");
        }
        }
}
```

OUTPUT

```
D:\java_lab>javac read.java

D:\java_lab>java read
Enter the text to create file : type exit to stop
hai friends
exit
Reading File content
hai friends
exit


D:\java_lab>
```

RESULT

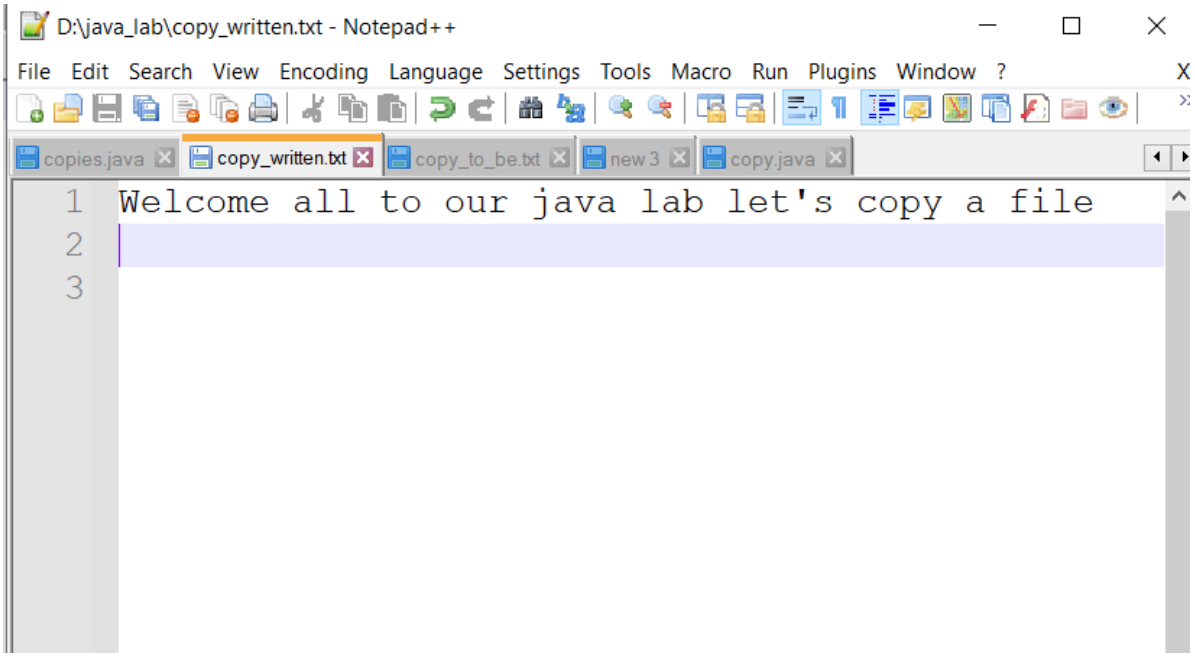
The program has been executed and output verified.

Experiment No:43

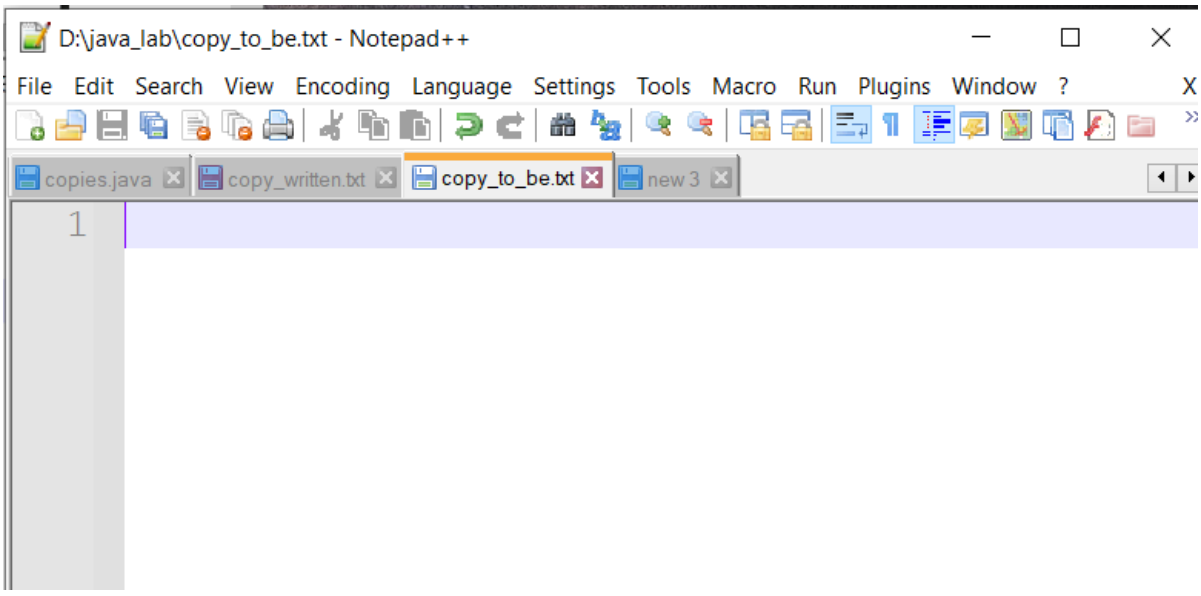
Write a program to copy one file to another

Pre-requisite

Create a text file with content where the java program is running for reading



And have another file to copy



PROGRAM

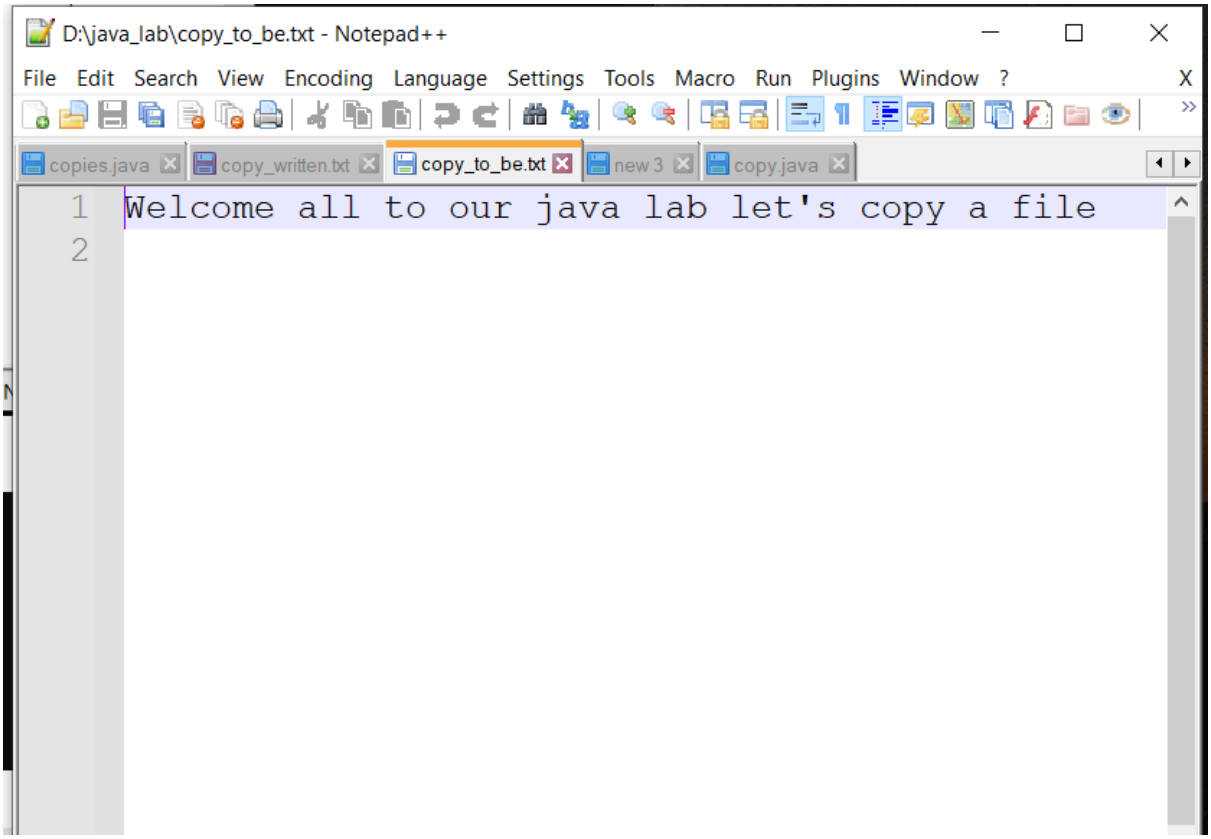
```
import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;
```

```
import java.io.*;
import java.util.*;
import java.io.File;
public class copy {
 public static void main(String[] args) {
 Scanner scan=new Scanner(System.in);
 System.out.println("Enter the source File Name");
 String source=scan.nextLine();
 try {
 FileReader fr=new FileReader(source);
 String str = "";
 int i;
 System.out.println("Reading from file "+source);
 while ((i = fr.read()) != -1) {
 // Storing every character in the string
 str += (char) i;
 }
 System.out.println(str);
 System.out.println("\nEnter the filename to copy");
 String destination=scan.nextLine();
 File file=new File(destination);
 FileWriter fw = new FileWriter(file);
 fw.write(str);
 fr.close();
 fw.close();
 System.out.println("Copied from "+source+" to "+destination+ " Successfully..!");
 } catch (Exception e) {
 //TODO: handle exception
 System.out.println("Exception Occured");
 }
 }
```

}

OUTPUT

```
D:\java_lab>java copy
Enter the source File Name
copy_written.txt
Reading from file copy_written.txt
Welcome all to our java lab let's copy a file


Enter the filename to copy
copy_to_be.txt
Copied from copy_written.txt to copy_to_be.txt Successfully..!

D:\java_lab>
```



RESULT

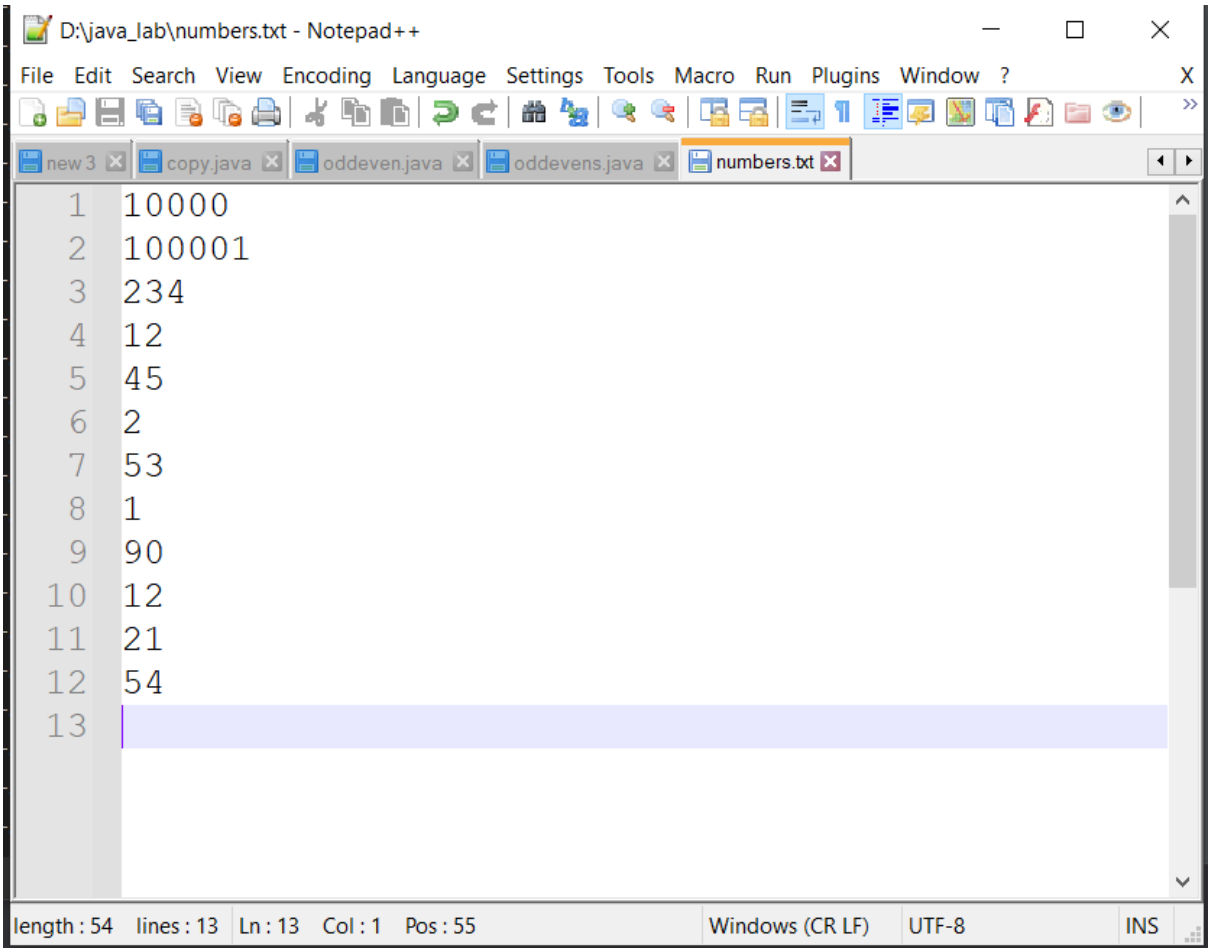
The program has been executed and output verified.

Experiment No:44

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

Pre-requisite

Create a text file with content of numbers where the java program is running for reading numbers



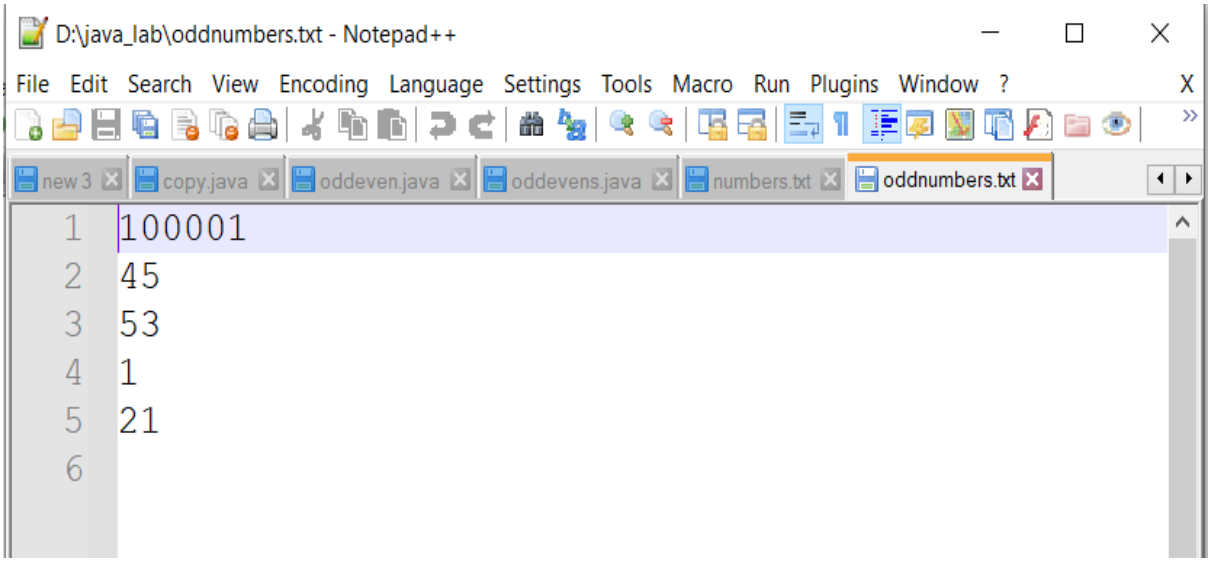
PROGRAM

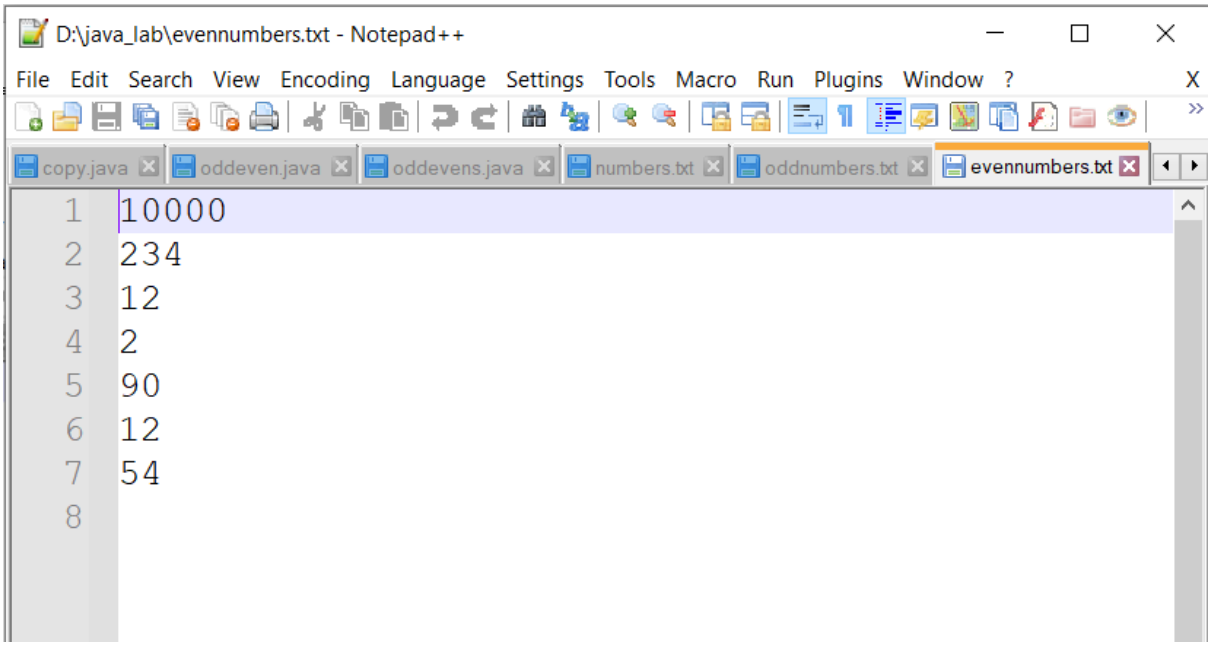
```
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.*;
import java.util.*;
import java.io.File;
public class oddeven {
 public static void main(String[] args) {
```

```
try {
FileReader fr = new FileReader("numbers.txt");
BufferedReader br = new BufferedReader(fr);
File file1 = new File("oddnumbers.txt");
FileWriter fw1 = new FileWriter(file1);
File file2 = new File("evennumbers.txt");
FileWriter fw2 = new FileWriter(file2);
String num;
while ((num = br.readLine()) != null) {
if (Integer.parseInt(num) % 2 == 0) {
fw2.write(num + "\n");
} else {
fw1.write(num + "\n");
}
}
fw1.close();
fw2.close();
} catch (Exception e) {
// TODO: handle exception
System.out.println("Error");
}
}
}
```

OUTPUT

```
D:\java_lab>javac oddeven.java

D:\java_lab>java oddeven

D:\java_lab>
```





RESULT

The program has been executed and output verified.

Experiment No: 45

Client server communication using Socket – TCP/IP

PROGRAM

Server

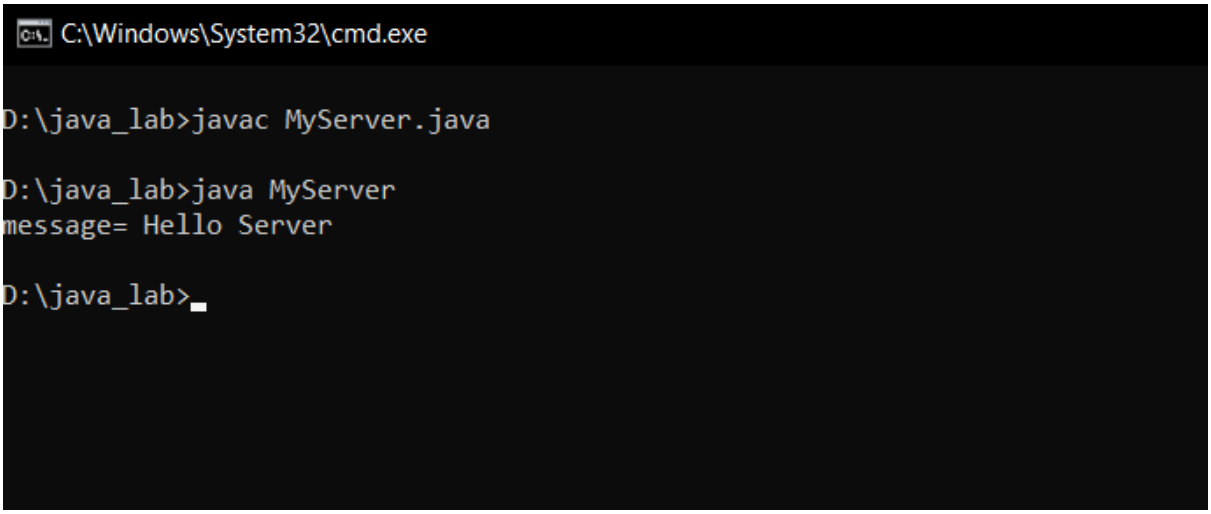
```
import java.io.*;
import java.net.*;
public class MyServer {
public static void main(String[] args) {
try{
ServerSocket ss=new ServerSocket(6666);
Socket s=ss.accept(); //establishes connection
DataInputStream dis=new DataInputStream(s.getInputStream());
String str=(String)dis.readUTF();
System.out.println("message= "+str);
ss.close();
}catch(Exception e) { System.out.println(e);}
}
}
```

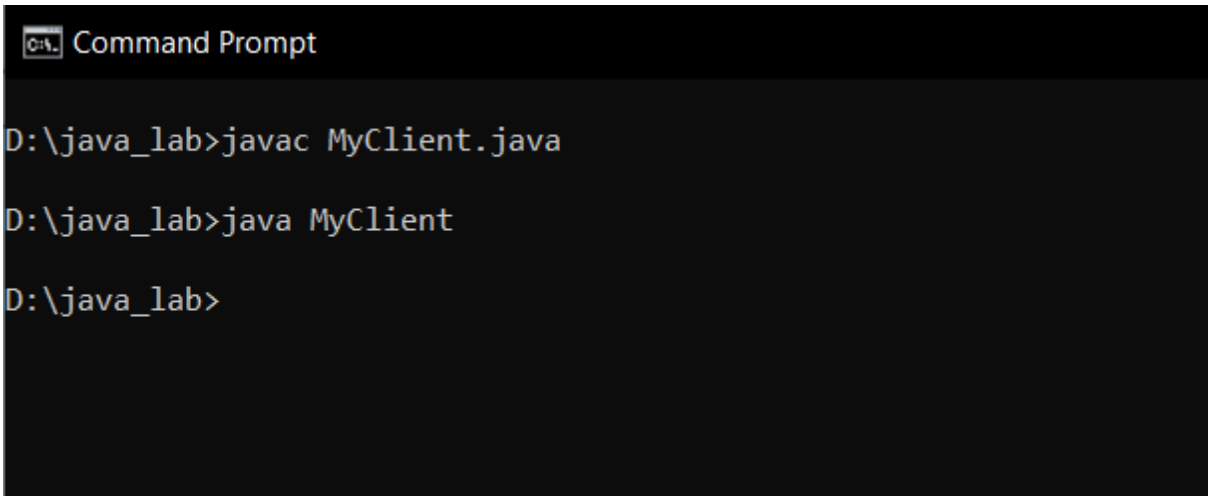
Client

```
import java.io.*;
import java.net.*;
public class MyClient {
public static void main(String[] args) {
try{
Socket s=new Socket("localhost",6666);
DataOutputStream dout=new DataOutputStream(s.getOutputStream());
dout.writeUTF("Hello Server"); // Writes a string to the underlying output stream using modified UTF-8 encoding
dout.flush();
```

```
dout.close();
s.close();
}catch(Exception e){System.out.println(e);}
}
}
```

OUTPUT





RESULT

The program has been executed and output verified.