PROGRAM-1

AIM:-

Read two matrices from user and perform matrix addition

ALGORITHM:-

CODE:-

```
import java.util.Scanner;
class matrix
{
	public static void main(String args[])
	{
		int m,n,i,j;
		Scanner in = new Scanner(System.in);
		System.out.println("Enter the number of rows in matrix:");
		m = in.nextInt();
		System.out.println("Enter the number of columns in matrix:");
		n = in.nextInt();
		int mat1[][] = new int[m][n];
		int mat2[][] = new int[m][n];
		int mat3[][] = new int[m][n];
		System.out.println("Enter the first matrix elements:");
		for (i=0 ;i<m ; i++)
			for (j=0; j<n; j++)
				mat1[i][j] = in.nextInt();
		System.out.println("Enter the second matrix elements:");
		for (i=0 ;i<m ; i++)
			for (j=0; j<n; j++)
				mat2[i][j] = in.nextInt();
		System.out.println("The matrix after addition:");
		for (i=0 ;i<m ; i++)
		{
			for (j=0; j<n; j++)
			{
				mat3[i][j] = mat1[i][j] + mat2[i][j];
				System.out.print(" "+ mat3[i][j]+ " ");
			}
			System.out.println();
		}
	}
}
```

RESULT:-

The program is done and result is obtained successfully.

OUTPUT:-

```
Enter the number of rows in matrix:
2
Enter the number of columns in matrix:
2
Enter the first matrix elements:
1 2 3 4
Enter the second matrix elements:
2 5 7 8
The matrix after addition:
3 7
10 12
```

PROGRAM-2

AIM:-

Add two complex numbers

ALGORITHM:-

CODE:-

```
import java.util.*;
public class Complex
{
	double real;
	double image;
	Complex(double r, double i)
	{
		real =r;
		image =i;
	}
	public static Complex sum(Complex c1, Complex c2)
	{
	Complex c = new Complex(0,0);
	c.real = c1.real + c2.real;
	c.image = c1.image + c2.image;
	return c;
	}
	public static void main(String args[])
	{
		Complex c1 = new Complex(5.3,2);
		Complex c2 = new Complex(7,3.1);
		Complex temp = sum(c1,c2);
		System.out.println("Complex number
		1:"+c1.real+"+"+c1.image+"i");
		System.out.println("Complex number 2:
		"+c2.real+"+"+c2.image+"i");
		System.out.println("Sum: "+temp.real+"+"+temp.image+"i");
	}
}
```

RESULT:-

The program is done and result is obtained successfully.

OUTPUT:-

```
Complex number 1: 5.3+2.0i
Complex number 2: 7.0+3.1i
Sum: 12.3+5.1i
```

PROGRAM-3

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.

ALGORITHM:-

CODE:-

```
class product
{
	String pcode,pname;
	int price;
	void read(String p,Stringn,int pi)
	{
		pcode = p;
		pname = n;
		price = pi;
	}
}
class product_details
{
	public static void main(String args[])
	{
		product p1 = new product();
		product p2 = new product();
		product p3 = new product();
		p1.read("e43","soap",100);
		p2.read("f44","brush",50);
		p3.read("k777","phone",25000);
		if(p1.price < p2.price && p1.price < p3.price)
		{
			System.out.println("The lowest price is: "+p1.price);
		}
		else if (p2.price< p3.price)
		{
			System.out.println("The lowest price is: "+p2.price);
		}
		else
		{
			System.out.println("The lowest price is: "+p3.price);
		}
	}
}
```

RESULT:-

The program is done and result is obtained successfully.

OUTPUT:-

The lowest price is: 50

PROGRAM-4

AIM:-

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.

ALGORITHM:-

CODE:-

```
class CPU
{
   int price;
   CPU(int p)
   {
      this.price = p;
   }
   class Processor
   {
      int cores;
      String manufacture;
      Processor(int n, String m)
     {
         this.cores = n;
         this.manufacture = m;
      }
      void display()
     {
         System.out.println("No of Cores : " + this.cores);
         System.out.println("Processor manufactures : " + this.manufacture);
      }
   }
   static class Ram
   {
      int memory;
      String manufacture;
```

```
        Ram(int n, String m)
        {
            this.memory = n;
            this.manufacture = m;
        }
        void display()
        {
            System.out.println("Memory Size : " + this.memory);
            System.out.println("Memory manufactures : " + this.manufacture);
        }
    }
    void display()
    {
        System.out.println("Price of CPU : " + this.price);
    }
    public static void main(String[] args)
    {
        CPU intel = new CPU(23000);
        CPU.Processor i_processor = intel.new Processor(4, "intel");
        CPU.Ram i_ram = new Ram(1024, "Asus");
        intel.display();
        i_processor.display();
        i_ram.display();
    }
}
```

RESULT:-

The program is done and result is obtained successfully.

OUTPUT:-

Price of CPU : 23000
No of Cores : 4
Processor manufactures : intel
Memory Size : 1024
Memory manufactures : Asus

PROGRAM-5

AIM:-

Program to Sort strings.

ALGORITHM:-

CODE:-

```
import java.util.Scanner;
class stringSort
{
	public static void main(String[] args)
	{
		int n;
		String words[]= new String[10],temp;
		Scanner in = new Scanner(System.in);
		System.out.println("Enter the number of strings:");
		n = in.nextInt();
		System.out.println("Enter the strings:");
		for (int i =0 ; i < n ; i++)
		{
		 words[i] = in.next();
		}
		for (int i = 0; i < n; i++)
		{
			for (int j = i + 1; j < n; j++)
			{
				if (words[i].compareTo(words[j]) > 0)
				{
					temp = words[i];
					words[i] = words[j];
					words[j] = temp;
				}
			}
```

```
            }
            System.out.println("\nThe strings in alphabetical order are: ");
            for (int i = 0; i < n; i++)
            {
                System.out.println(words[i]);
            }
        }
}
```

RESULT:-

The program is done and result is obtained successfully.

OUTPUT:-

```
Enter the number of strings:
3
Enter the strings:
mango
apple
orange

The strings in alphabetical order are:
apple
mango
orange
```

PROGRAM-6

AIM:-

Perform string manipulations

ALGORITHM:-

CODE:-

```
public class fun_string
{
 public static void main(String[] args)
{
//String Concatenation
String str1 = "Rock";
String str2 = "Star";

//Method 1 : Using concat

String str3 = str1.concat(str2);
System.out.println("String concantenation method1:"+str3);

//Method 2 : Using "+" operator

String str4 = str1 + str2;
System.out.println("String concantenation method2:"+str4);

//Length of a String
System.out.println("Length of String: " + str1.length());

//character at specific position

System.out.println("Character at position 3: " + str1.charAt(3));

//Compare to
System.out.println("Compare To 'Rock': " + str1.compareTo("Rom"));

//Compare to - Ignore case
System.out.println("Compare To 'Rock' - Case Ignored: " +
str1.compareToIgnoreCase("rock"));

//using endsWith
System.out.println("EndsWith character 'k': " + str1.endsWith("k"));

//Convert to lowerCase
System.out.println("Convert to LowerCase: " + str1.toLowerCase());

//Convert to UpperCase
System.out.println("Convert to UpperCase: " + str1.toUpperCase());

//equals
System.out.println("Equals -'Rock': " + str1.equals("rock"));

//equalsIgnoreIgnoreCase
System.out.println("Equals ignore case -'Rock' :" + str1.equalsIgnoreCase("rock"));
}
}
```

RESULT:-

The program is done and result is obtained successfully.

OUTPUT:-

String concantenation method1:RockStar
String concantenation method2:RockStar
Length of String: 4
Character at position 3: k
Compare To 'Rock': -10
Compare To 'Rock' - Case Ignored: 0
EndsWith character 'k': true
Convert to LowerCase: rock
Convert to UpperCase: ROCK
Equals -'Rock': false
Equals ignore case -'Rock'

PROGRAM-7

AIM:-

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

ALGORITHM:-

CODE:-

```
import java.util.Scanner;
public class employee
{
int eNo;
String name;
float salary;
public void getInput()
{
        Scanner in =  new Scanner(System.in);
        System.out.println("Enter the employee id:");
        eNo = in.nextInt();
        System.out.println("Enter the employee name:");
        name = in.next();
        System.out.println("Enter the salary:");
        salary = in.nextFloat();
}
public void display()
{
System.out.println("Employee id:"+eNo);
System.out.println("Employee name:"+name);
System.out.println("Salary:"+salary);
}
public static void main(String[] args)
{
System.out.println("Enter the number of employees:");
Scanner in =  new Scanner(System.in);
```

```
int n = in.nextInt();
employee e[]= new employee[n];
for (int i=0;i<n;i++)
{
        e[i] = new employee();
        e[i].getInput();
}
for(int i=0;i<n;i++)
{
int j=i+1;
System.out.println("\nEmployee No:"+j);
System.out.println("**************");
e[i].display();
}
}
}
```

RESULT:-

The program is done and result is obtained successfully.

OUTPUT:-

```
Enter the number of employees:
3
Enter the employee id:
101
Enter the employee name:
abhi
Enter the salary:
1000
```

Enter the employee id:
102
Enter the employee name:
anu
Enter the salary:
1500
Enter the employee id:
103
Enter the employee name:
arun
Enter the salary:
1200

Employee No:1
*****
Employee id:101
Employee name:abhi
Salary:1000.0

Employee No:2
*****
Employee id:102
Employee name:anu
Salary:1500.0

Employee No:3
*****
Employee id:103
Employee name:arun
Salary:1200.0

PROGRAM-8

AIM:-

Area of different shapes using overloaded functions

ALGORITHM:-

CODE:-

```
import java.util.*;
public class overload
{
  public double area(double x)
    {
      return(3.14*x*x);
    }
  public double area(double x,double y)
    {
      return(x*y);
     }
public static void main(String args[])
{
overload obj=new overload();
Scanner s=new Scanner(System.in);
System.out.println("Enter the values:");
double x = s.nextDouble();
double y = s.nextDouble();
System.out.println("Area of circle is:"+obj.area(x));
System.out.println("Area of rectangle is:"+obj.area(x,y));
  }
}
```

RESULT:-

The program is done and result is obtained successfully.

OUTPUT:-

```
Enter the values:
2
4
Area of circle is:12.56
Area of rectangle is:8.0
```

PROGRAM-9

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, Teacher id and also contain constructors and methods to display the data members. Use array of objects to display details of N teachers.

ALGORITHM:-

CODE:-

```
import java.util.*;
class Person
{
	public String name;
	public String gender;
	public String address;
	public int age;
	public Person(String n, String g, String add, int ag)
	{
		name=n;
		gender=g;
		address=add;
		age=ag;
	}
}
class Employee extends Person
{
	public String empid;
	public String company_name;
	public String qualification;
	public float salary;
	public Employee(String id, String c, String q, float sal, String n, String g, String add, int age)
	{
		super(n,g,add,age);
		empid=id;
		company_name=c;
```

```
            qualification=q;
            salary=sal;
        }
}

class Teacher extends Employee
{
public String subject;
public String department;
public String teacher_id;



public Teacher(String sub, String dept, String tid, String id, String c, String q,
float sal, String n, String g, String add, int ag)
{
        super(id, c, q, sal, n, g, add, ag);

        subject=sub;
        department=dept;
        teacher_id=tid;
}

public void display()
{
        System.out.println("Teacher:");
        System.out.println("Name: "+name);
        System.out.println("Gender: "+gender);
        System.out.println("Address: "+address);
```

```
        System.out.println("Age: "+age);
        System.out.println(" ");
        System.out.println("Employee id: "+empid);
        System.out.println("Company name: "+company_name);
        System.out.println("Qualification: "+qualification);
        System.out.println("Salary: "+salary);
        System.out.println(" ");
        System.out.println("Teacher id: "+teacher_id);
        System.out.println("Subject: "+subject);
        System.out.println("Department: "+department);
}
public static void main(String args[])
{
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter no. of teachers: ");
        int n = sc.nextInt();
        Teacher[] list = new Teacher[20];
        System.out.println("Enter teachers' data: \n\n");
        for (int i=0; i<n; i++)
        {
                System.out.println("Enter name: ");
                String name=sc.next();
                System.out.println("Enter gender: ");
                String gen=sc.next();
                System.out.println("Enter age: ");
                int ag=sc.nextInt();
                System.out.println("Enter address: ");
```

```
            String ad=sc.next();
            System.out.println("Enter employee id: ");
            String eid=sc.next();
            System.out.println("Enter company name: ");
            String cn=sc.next();
            System.out.println("Enter qualification: ");
            String qu=sc.next();
            System.out.println("Enter salary: ");
            float salary=sc.nextFloat();
            System.out.println("Enter teacher id: ");
            String tid=sc.next();
            System.out.println("Enter subject: ");
            String sub=sc.next();
            System.out.println("Enter department: ");
            String dep=sc.next();
            list[i]= new Teacher(sub,dep,tid,eid,cn,qu,salary,name,gen,ad,ag);
        }
        System.out.println("\n\n Displaying teacher details: \n");
        for(int i=0; i<n; i++){
        list[i].display();
        System.out.println("\n\n");
        }
}

}
```

RESULT:-

The program is done and result is obtained successfully.

OUTPUT:-

```
Enter no. of teachers:
2
Enter teachers' data:


Enter name:
Ram
Enter gender:
M
Enter age:
28
Enter qualification:
MCA
Enter salary:
70000
Enter teacher id:
455
Enter subject:
CS
Enter department:
CSE
Enter name:
Jiya
Enter gender:
F
Enter age:
35
Enter qualification:
Mtech
Enter salary:
60000
Enter teacher id:
989
Enter subject:
Electronics
```

Enter department:
EEE

Displaying teacher details:

Teacher:
Name: Ram
Gender: M
Address: jkl
Age: 28
Employee id: 123
Company name: abc
Qualification: MCA
Salary: 70000.0
Teacher id: 455
Subject: CS
Department: CSE

Teacher:
Name: Jiya
Gender: F
Address: pqr
Age: 35
Employee id: 554
Company name: gic
Qualification: Mtech
Salary: 60000.0
Teacher id: 989
Subject: Electronics
Department: EEE

PROGRAM-10

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.

ALGORITHM:-

CODE:-

```
import java.util.Scanner;
interface shape
{
        double pi = 3.14;
        double area(double x,double y);
        double perimeter(double x,double y);
}
class circle implements shape{
        public double area(double x,double y)
        {
                return(pi*x*x);
        }
        public double perimeter(double x,double y)
        {
        return(2*pi*x);
        }
}
class rect implements shape{
public double area(double x,double y)
{
return(x*y);
}
public double perimeter(double x,double y)
{
return(2*(x+y));
}
}
```

```
class interfacepgm
{
public static void main(String arg[])
{
Scanner scan = new Scanner(System.in);
while(true){
System.out.println("1:Area&Perimeter of circle");
System.out.println("2:Area &Perimeter of Rectangle");
System.out.println("3:Exit");
System.out.println("Enter your choice::");
int choice = scan.nextInt();

switch(choice){
case 1:
  shape obj=new circle();
  System.out.println("Area of circle is:"+obj.area(2.0,1.0));
  System.out.println("Perimeter of circle is:"+obj.perimeter(2.0,1.0));
break;
case 2:
  shape obj2=new rect();
  System.out.println("Area of recangle is:"+obj2.area(3.0,2.0));
  System.out.println("Perimeter of rectangle is:"+obj2.perimeter(3.0,2.0));
break;
case 3:
  System.out.println("Exiting the application");
  System.exit(0);
default: System.out.println("Incorrect input!!! Please re-enter choice from our
menu");
}
```

```
}
}
}
```

RESULT:-

The program is done and result is obtained successfully.

OUTPUT:-

```
1:Area&Perimeter of circle
2:Area &Perimeter of Rectangle
3:Exit
Enter your choice::
1
Area of circle is:12.56
Perimeter of circle is:12.56
1:Area&Perimeter of circle
2:Area &Perimeter of Rectangle
3:Exit
Enter your choice::
2
Area of recangle is:6.0
Perimeter of rectangle is:10.0
1:Area&Perimeter of circle
2:Area &Perimeter of Rectangle
3:Exit
Enter your choice::
3
Exiting the application
```

PROGRAM-11

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.

ALGORITHM:-

CODE:-

```
package Graphics;

public class Circle{
public void area(float r){
float area = 3.14f*r*r;
System.out.println("Area of circle is "+area);
}
}

package Graphics;

public class Rectangle{
public void area(int l, int b){
int area = l*b;
System.out.println("Area of rectangle is "+area);
}
}

package Graphics;

public class Square{
public void area(int s){
int area = s*s;
System.out.println("Area of square is "+area);
}
}

package Graphics;

public class Triangle{
public void area(float h, float b){
float area = 0.5f*b*h;
System.out.println("Area of triangle is "+area);
}
}

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

```
class Graphictest{
public static void main(String args[]){
Scanner sc = new Scanner(System.in);
Graphics.Rectangle r = new Graphics.Rectangle();
Graphics.Square s = new Graphics.Square();
Graphics.Circle c = new Graphics.Circle();
Graphics.Triangle t = new Graphics.Triangle();

int ch=1, flag, a, b;
float x,y;

do{
System.out.println("Select a shape: ");
System.out.println("1 - Rectangle ");
System.out.println("2 - Square ");
System.out.println("3 - Circle ");
System.out.println("4 - Triangle ");

flag=sc.nextInt();

switch(flag){

case 1: System.out.println("Enter length: ");
        a=sc.nextInt();
        System.out.println("Enter breadth: ");
        b=sc.nextInt();
        r.area(a,b);
        break;

case 2:System.out.println("Enter side: ");
        a=sc.nextInt();
        s.area(a);
        break;

case 3:System.out.println("Enter radius: ");
        x=sc.nextFloat();
        c.area(x);
        break;
```

```
case 4:System.out.println("Enter base: ");
        x=sc.nextFloat();
        System.out.println("Enter height: ");
        y=sc.nextFloat();
        t.area(x,y);
        break;

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

System.out.println("Press 0 to EXIT - any other to continue");
ch=sc.nextInt();

}while(ch!=0);
}

}
```

RESULT:-

The program is done and result is obtained successfully.

OUTPUT:-

```
Select a shape:
1 - Rectangle
2 - Square
3 - Circle
4 - Triangle
1
Enter length:
2
Enter breadth:
4
Area of rectangle is 8
Press 0 to EXIT - any other to continue
3
Select a shape:
1 - Rectangle
2 - Square
3 - Circle
4 - Triangle
```

2
Enter side:
3
Area of square is 9
Press 0 to EXIT - any other to continue

PROGRAM-12

AIM:-

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

ALGORITHM:-

CODE:-

```
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 Check {

 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("123456"))
    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");
 }
 }
}
```

RESULT:-

The program is done and result is obtained successfully.

OUTPUT:-

```
Enter username :: abcdef
Enter password :: 123456
Login Successful !!!
The finally statement is executed
```

PROGRAM-13

AIM:-

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

ALGORITHM:-

CODE:-

```
import java.util.*;

class MyException extends Exception {
   public MyException(String value) {
      super(value);
   }
}

class Main {
   public static void main(String args[]) {
      int totalNums;
      int i;
      int temp, count = 0;
      int sum = 0;

      Scanner sc = new Scanner(System.in);

      System.out.println("Total numbers");
      totalNums = Integer.parseInt(sc.nextLine());
      for (i = 0; i < totalNums; i++) {
         try {
            temp = Integer.parseInt(sc.nextLine());
            if (temp > 0) {
               sum += temp;
               count += 1;
            } else {
               throw new MyException(Integer.toString(temp));
            }
         } catch (MyException ex) {
            System.out.print(ex.getMessage());
            System.out.println(" - Not a positive number");
         }
      }
      System.out.print("Count : ");
      System.out.println(count);
      System.out.print("sum: ");
      System.out.println(sum);
      System.out.print("Average : ");
      System.out.println(sum / count);

   }
}
```

RESULT:-

The program is done and result is obtained successfully.

OUTPUT:-

```
Total numbers
5
6
7
-9
-9 - Not a positive number
3
4
Count : 4
sum: 20
Average : 5
```

PROGRAM-14

AIM:-

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

ALGORITHM:-

CODE:-

```
import java.util.*;

class fibonacci implements Runnable {
   int l;

   fibonacci(int n) {
      l = n;
   }

   public void run() {
      int c;
      int a = 0, b = 1;
      System.out.print(a + " " + b);
      for (int i = 0; i <= l; i++) {
         c = a + b;
         System.out.print(" " + c);
         a = b;
         b = c;
      }
   }
}

class even implements Runnable {
   int l;

   even(int n) {
      l = n;
   }

   public void run() {
      for (int i = 0; i <= l; i++) {
         if (i % 2 == 0)
            System.out.print(i + "  ");
      }
      System.out.println("");

   }

}

class Main {
   public static void main(String args[]) {
      Scanner sc = new Scanner(System.in);
      System.out.println("Enter Limit :");
      int l = sc.nextInt();
```

```
            fibonacci f = new fibonacci(l);
            Thread T1 = new Thread(f);
            T1.start();
            even e = new even(l);
            Thread T2 = new Thread(e);
            T2.start();
        }
    }
```

RESULT:-

The program is done and result is obtained successfully.

OUTPUT:-

```
Enter Limit :
10
0  2  4  6  8  10
0 1 1 2 3 5 8 13 21 34 55 89 144
```

PROGRAM-15

AIM:-

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

ALGORITHM:-

CODE:-

```
import java.util.ArrayList;
import java.util.Collections;

class Main {
   public static void main(String[] args) {
      // string list
      ArrayList<String> data = new ArrayList<String>();
      // add
      data.add("A");
      data.add("B");
      data.add("C");
      data.add("D");
      // change
      data.set(1, "BB");
      System.out.println(data);

      // get
      System.out.println(data.get(0));
      System.out.println(data.get(1));

      // remove
      data.remove(0);
      System.out.println(data);

      // size
      System.out.println(data.size());

      // traversing
      for (String d : data) {
         System.out.println(d);
      }


      // sort
      Collections.sort(data);
      System.out.println(data);

      // clear
```

```
        data.clear();
        System.out.println(data);
    }
}
```

RESULT:-

The program is done and result is obtained successfully.

OUTPUT:-

```
[A, BB, C, D]
A
BB
[BB, C, D]
3
BB
C
D
[BB, C, D]
[]
```

PROGRAM-16

AIM:-

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

ALGORITHM:-

CODE:-

```
import java.util.*;


class stack_list {
   public static void main(String arg[]) {
      Stack<Integer> s = new Stack<Integer>();
      Scanner sc = new Scanner(System.in);
      int n;
      // Stack<integer> s = new Stack<integer>();
      do {
         System.out.println("\n\t\t\tSTACK LIST");
         System.out.println("\n\t\t=========================");
         System.out.println("\n\t 1.ADD");
         System.out.println("\n\t 2.SIZE");
         System.out.println("\n\t 3.PEEK");
         System.out.println("\n\t 4.POP");
         System.out.println("\n\t 5.SEARCH");
         System.out.println("\n\t 6.TRAVERSAL");
         System.out.println("\n\t 7.REMOVE");
         System.out.println("\n\t 8.EXIT");
         System.out.print("\n\t\t enter your choice = ");
         n = sc.nextInt();
         switch (n) {
            case 1:
               s.add(100);
               System.out.println("\n\tnew stack = " + s);
               break;
            case 2:
               System.out.println("\n\t the size of the stack = " + s.size());
               break;
            case 3:
               System.out.println("\n\t peek of the stack = " + s.peek());
               break;
            case 4:
               System.out.println("\n\t pop of the stack = " + s.pop());
               break;
            case 5:
               System.out.println("\n\t searching element of the stack = " +
s.search(400));
```

```
                break;
            case 6:
                for (Integer i : s)
                    System.out.println("\n\ttraversing each element in the stack = " +
i + " ");
                break;
            case 7:
                System.out.println("\n\t removing ana element in the stack = " +
s.remove(0));
                System.out.println("\n\t new stack = " + s);
                break;
            case 8:
                System.out.println("\n\t program exiting...!!!!!");
                break;
            default:
                System.out.println("\n\t try another menu.....");
          }

      } while (n != 8);
   }
}
```

RESULT:-

The program is done and result is obtained successfully.

OUTPUT:-

```
                    STACK LIST

          ==========================

      1.ADD

      2.SIZE

      3.PEEK

      4.POP

      5.SEARCH

      6.TRAVERSAL
```

7.REMOVE

8.EXIT

enter your choice = 1

new
stack = [100]

PROGRAM-17

AIM:-

Program to demonstrate the creation of queue object using the Priority Queue class

ALGORITHM:-

CODE:-

```
import java.util.PriorityQueue;
import java.util.Queue;

public class PriorityQueueDemo {

	public static void main(String[] args) {

		Queue<Integer> q = new PriorityQueue<>();
		System.out.println(q.peek());

		for (int i = 20; i <= 30; i++) {
			q.offer(i);
		}

		System.out.println(q);
		System.out.println(q.poll());
		System.out.println(q);
		System.out.println(q.remove());
		System.out.println(q);

	}

}
```

RESULT:-

The program is done and result is obtained successfully.

OUTPUT:-

```
null
[20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30]
20
[21, 23, 22, 27, 24, 25, 26, 30, 28, 29]
21
[22, 23, 25, 27, 24, 29, 26, 30, 28]
```

PROGRAM-18

AIM:-

Program to demonstrate the addition and deletion of elements in deque

ALGORITHM:-

CODE:-

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

		// add() method to insert
		dq.add("For");
		dq.addFirst("Geeks");
		dq.addLast("Geeks");

		System.out.println(dq);

		System.out.println(dq.pop());

		System.out.println(dq.poll());

		System.out.println(dq.pollFirst());

		System.out.println(dq.pollLast());
	}
}
```

RESULT:-

The program is done and result is obtained successfully.

OUTPUT:-

```
[Geeks, For, Geeks]
Geeks
For
Geeks
null
```

PROGRAM-19

AIM:-

Write a Java program to compare two hash set

ALGORITHM:-

CODE:-

```
import java.util.HashSet;

class Main {
   public static void main(String[] args) {
      HashSet<String> marks = new HashSet<String>();
      marks.add("A");
      marks.add("B");
      marks.add("B"); // must  be unique
      System.out.println(marks);
      System.out.println(marks.contains("A"));
      //marks.remove("A");
      //System.out.println(marks.size());

     for(String i:marks){
         System.out.println(i);
     }

      marks.clear();
   }
}
```

RESULT:-

The program is done and result is obtained successfully.

OUTPUT:-

```
[A, B]
true
A
B
```

PROGRAM-20

AIM:-

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

ALGORITHM:-

CODE:-

```
import java.util.HashMap;

class Main {
   public static void main(String[] args) {
      HashMap<String,Integer> marks = new HashMap<String,Integer>();
      marks.put("A",123);
      marks.put("B",122);
      System.out.println(marks);
      System.out.println(marks.get("A"));
      //System.out.println(marks.remove("A"));
      //System.out.println(marks);
      System.out.println(marks.size());

      for(String i:marks.keySet()){
         System.out.println(i);
      }
      for(int i:marks.values()){
         System.out.println(i);
      }
      for(String i:marks.keySet()){
         System.out.println("key : "+i+ " value : "+marks.get(i));
      }
      marks.clear();
   }
}
```

RESULT:-

The program is done and result is obtained successfully.

OUTPUT:-

```
{A=123, B=122}
123
2
A
B
123
122
key : A value : 123
key : B value : 122
```

PROGRAM-21

AIM:-

Implement a simple calculator using AWT components.

CODE:-

```
import java.awt.*;
import java.awt.event.*;
class MyCalc extends WindowAdapter implements ActionListener{
  Frame f;
Label l1;
Button b1,b2,b3,b4,b5,b6,b7,b8,b9,b0;
Button badd,bsub,bmult,bdiv,bmod,bcalc,bclr,bpts,bneg,bback;
double xd;
double num1,num2,check;

MyCalc(){
  f= new Frame("MY CALCULATOR");
// INSTANTIATING COMPONENETS
l1=new Label();
l1.setBackground(Color.LIGHT_GRAY);
l1.setBounds(50,50,260,60);

b1=new Button("1");
  b1.setBounds(50,340,50,50);
b2=new Button("2");
  b2.setBounds(120,340,50,50);
b3=new Button("3");
  b3.setBounds(190,340,50,50);
b4=new Button("4");
  b4.setBounds(50,270,50,50);
b5=new Button("5");
  b5.setBounds(120,270,50,50);
b6=new Button("6");
  b6.setBounds(190,270,50,50);
b7=new Button("7");
  b7.setBounds(50,200,50,50);
b8=new Button("8");
```

```
  b8.setBounds(120,200,50,50);
b9=new Button("9");
  b9.setBounds(190,200,50,50);
b0=new Button("0");
  b0.setBounds(120,410,50,50);
bneg=new Button("+/-");
  bneg.setBounds(50,410,50,50);
bpts=new Button(".");
  bpts.setBounds(190,410,50,50);
bback=new Button("back");
 bback.setBounds(120,130,50,50);

badd=new Button("+");
  badd.setBounds(260,340,50,50);
bsub=new Button("-");
  bsub.setBounds(260,270,50,50);
bmult=new Button("*");
  bmult.setBounds(260,200,50,50);
bdiv=new Button("/");
  bdiv.setBounds(260,130,50,50);
bmod=new Button("%");
  bmod.setBounds(190,130,50,50);
bcalc=new Button("=");
  bcalc.setBounds(245,410,65,50);
bclr=new Button("CE");
  bclr.setBounds(50,130,65,50);

b1.addActionListener(this);
b2.addActionListener(this);
b3.addActionListener(this);
b4.addActionListener(this);
b5.addActionListener(this);
b6.addActionListener(this);
b7.addActionListener(this);
b8.addActionListener(this);
b9.addActionListener(this);
b0.addActionListener(this);

bpts.addActionListener(this);
bneg.addActionListener(this);
```

```
bback.addActionListener(this);

badd.addActionListener(this);
bsub.addActionListener(this);
bmult.addActionListener(this);
bdiv.addActionListener(this);
bmod.addActionListener(this);
bcalc.addActionListener(this);
bclr.addActionListener(this);

f.addWindowListener(this);
//ADDING TO FRAME
f.add(l1);
f.add(b1); f.add(b2); f.add(b3); f.add(b4); f.add(b5);f.add(b6); f.add(b7);
f.add(b8);f.add(b9);f.add(b0);

f.add(badd); f.add(bsub); f.add(bmod); f.add(bmult); f.add(bdiv);
f.add(bmod);f.add(bcalc);

f.add(bclr); f.add(bpts);f.add(bneg); f.add(bback);

f.setSize(360,500);
f.setLayout(null);
f.setVisible(true);
}
                //FOR CLOSING THE WINDOW
public void windowClosing(WindowEvent e) {
  f.dispose();
}

public void actionPerformed(ActionEvent e){
  String z,zt;
                  //NUMBER BUTTON
if(e.getSource()==b1){
 zt=l1.getText();
  z=zt+"1";
  l1.setText(z);
}
if(e.getSource()==b2){
zt=l1.getText();
```

```
z=zt+"2";
l1.setText(z);
}
if(e.getSource()==b3){
  zt=l1.getText();
  z=zt+"3";
  l1.setText(z);
}
if(e.getSource()==b4){
  zt=l1.getText();
  z=zt+"4";
  l1.setText(z);
}
if(e.getSource()==b5){
  zt=l1.getText();
  z=zt+"5";
  l1.setText(z);
}
if(e.getSource()==b6){
  zt=l1.getText();
  z=zt+"6";
  l1.setText(z);
}
if(e.getSource()==b7){
  zt=l1.getText();
  z=zt+"7";
  l1.setText(z);
}
if(e.getSource()==b8){
  zt=l1.getText();
  z=zt+"8";
  l1.setText(z);
}
if(e.getSource()==b9){
  zt=l1.getText();
  z=zt+"9";
  l1.setText(z);
}
if(e.getSource()==b0){
  zt=l1.getText();
```

```
  z=zt+"0";
  l1.setText(z);
}

if(e.getSource()==bpts){  //ADD DECIMAL PTS
  zt=l1.getText();
  z=zt+".";
  l1.setText(z);
}
if(e.getSource()==bneg){ //FOR NEGATIVE
  zt=l1.getText();
  z="-"+zt;
  l1.setText(z);
}

if(e.getSource()==bback){  // FOR  BACKSPACE
  zt=l1.getText();
  try{
    z=zt.substring(0, zt.length()-1);
    }catch(StringIndexOutOfBoundsException f){return;}
  l1.setText(z);
}
              //AIRTHMETIC BUTTON
if(e.getSource()==badd){                  //FOR ADDITION
  try{
    num1=Double.parseDouble(l1.getText());
    }catch(NumberFormatException f){
      l1.setText("Invalid Format");
      return;
    }
  z="";
  l1.setText(z);
  check=1;
}
if(e.getSource()==bsub){                  //FOR SUBTRACTION
  try{
    num1=Double.parseDouble(l1.getText());
    }catch(NumberFormatException f){
      l1.setText("Invalid Format");
      return;
```

```
    }
  z="";
  l1.setText(z);
  check=2;
}
if(e.getSource()==bmult){                //FOR MULTIPLICATION
  try{
    num1=Double.parseDouble(l1.getText());
    }catch(NumberFormatException f){
      l1.setText("Invalid Format");
      return;
    }
  z="";
  l1.setText(z);
  check=3;
}
if(e.getSource()==bdiv){                 //FOR DIVISION
  try{
    num1=Double.parseDouble(l1.getText());
    }catch(NumberFormatException f){
      l1.setText("Invalid Format");
      return;
    }
  z="";
  l1.setText(z);
  check=4;
}
if(e.getSource()==bmod){                 //FOR MOD/REMAINDER
  try{
    num1=Double.parseDouble(l1.getText());
    }catch(NumberFormatException f){
      l1.setText("Invalid Format");
      return;
    }
  z="";
  l1.setText(z);
  check=5;
}
                    //RESULT BUTTON
if(e.getSource()==bcalc){
```

```
  try{
    num2=Double.parseDouble(l1.getText());
    }catch(Exception f){
      l1.setText("ENTER NUMBER FIRST ");
      return;
    }
  if(check==1)
    xd =num1+num2;
  if(check==2)
    xd =num1-num2;
  if(check==3)
    xd =num1*num2;
  if(check==4)
    xd =num1/num2;
  if(check==5)
    xd =num1%num2;
  l1.setText(String.valueOf(xd));
}
                    //FOR CLEARING THE LABEL and Memory
if(e.getSource()==bclr){
  num1=0;
  num2=0;
  check=0;
  xd=0;
   z="";
   l1.setText(z);
    }

}
//MAIN METHOD where objects of MyCalc is instantaiated
 public static void main(String args[]){
       new MyCalc();
    }
}
```

RESULT:-

The program is done and result is obtained successfully.

OUTPUT:-

PROGRAM-22

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.

CODE:-

```
import java.awt.Dimension;
import java.awt.EventQueue;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.GridBagConstraints;
import java.awt.GridBagLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.ButtonGroup;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JRadioButton;
import javax.swing.UIManager;
import javax.swing.UnsupportedLookAndFeelException;

public class DrawStuff extends JFrame {

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

   public DrawStuff() {
      EventQueue.invokeLater(new Runnable() {
         @Override
         public void run() {
            try {

UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName())
;
            } catch (ClassNotFoundException | InstantiationException |
IllegalAccessException | UnsupportedLookAndFeelException ex) {
                ex.printStackTrace();
            }
```

```
            JFrame frame = new JFrame("Testing");
            frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
            frame.add(new ControlPane());
            frame.pack();
            frame.setLocationRelativeTo(null);
            frame.setVisible(true);
        }
    });
}

public class ControlPane extends JPanel {

    private JRadioButton circle;
    private JRadioButton square;

    private DrawPane drawPane;

    public ControlPane() {
        setLayout(new GridBagLayout());

        ButtonGroup bg = new ButtonGroup();
        circle = new JRadioButton("Circle");
        square = new JRadioButton("Square");

        bg.add(circle);
        bg.add(square);

        GridBagConstraints gbc = new GridBagConstraints();
        gbc.gridwidth = GridBagConstraints.REMAINDER;
        gbc.weightx = 1;

        JPanel shape = new JPanel();
        shape.add(circle);
        shape.add(square);
        add(shape, gbc);

        JButton draw = new JButton("Draw");
        draw.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                if (circle.isSelected()) {
                    drawPane.setDrawableShape(DrawableShape.CIRCLE);
                } else if (square.isSelected()) {
```

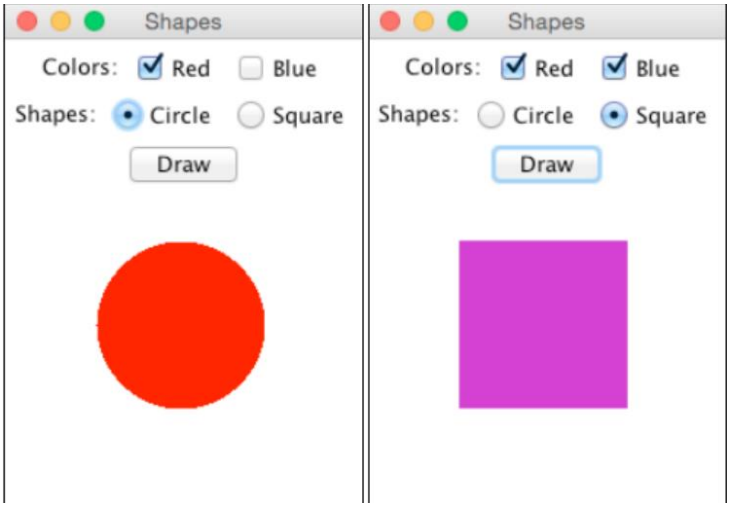
```
                drawPane.setDrawableShape(DrawableShape.SQUARE);
            }
        }
    });

    gbc.gridwidth = GridBagConstraints.REMAINDER;
    add(draw, gbc);

    drawPane = new DrawPane();

    gbc.weightx = 1;
    gbc.weighty = 1;
    gbc.fill = gbc.BOTH;
    add(drawPane, gbc);
  }

}

public enum DrawableShape {
  CIRCLE,
  SQUARE
}

public class DrawPane extends JPanel {

  private DrawableShape drawableShape;

  public DrawPane() {
  }

  public void setDrawableShape(DrawableShape drawableShape) {
    this.drawableShape = drawableShape;
    repaint();
  }

  public DrawableShape getDrawableShape() {
    return drawableShape;
  }

  @Override
  public Dimension getPreferredSize() {
    return new Dimension(200, 200);
  }
```

```
    @Override
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
        Graphics2D g2d = (Graphics2D) g.create();
        DrawableShape shape = getDrawableShape();
        if (shape != null) {
            int width = getWidth() - 20;
            int height = getHeight() - 20;
            int size = Math.min(width, height);

            int x = (getWidth() - size) / 2;
            int y = (getHeight() - size) / 2;
            if (shape == DrawableShape.CIRCLE) {
                g2d.fillOval(x, y, size, size);
            } else if (shape == DrawableShape.SQUARE) {
                g2d.fillRect(x, y, size, size);
            }
        }
        g2d.dispose();
    }

  }
}
```

RESULT:-

The program is done and result is obtained successfully.

OUTPUT:-


Shapes
Colors: Red Blue
Shapes: Circle Square
Draw
Shapes
Colors: Red Blue
Shapes: Circle Square
Draw

PROGRAM-23

AIM:-

Develop a program to handle all mouse events and window events

CODE:-

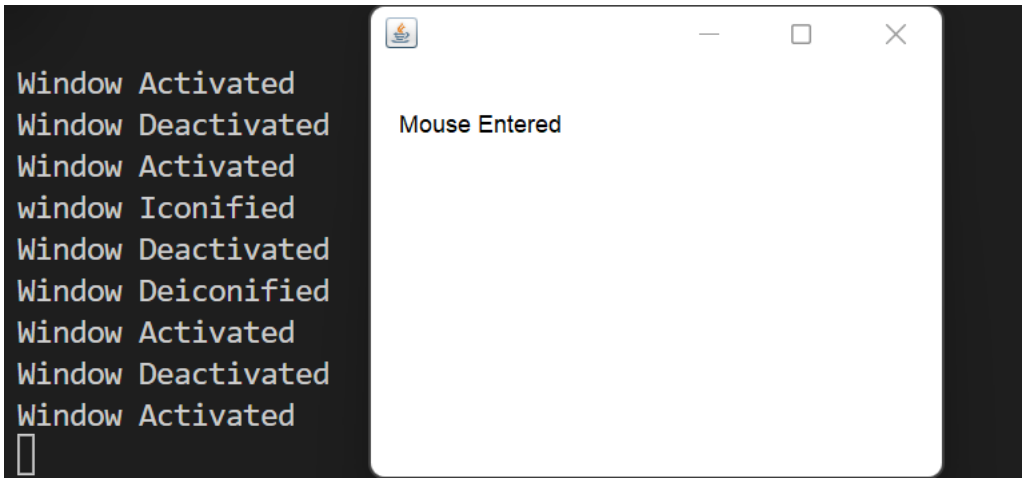
```
import javax.swing.*;
 import java.awt.event.*;
 import java.awt.*;
 class WindowListenerExample extends JFrame implements WindowListener
 {
       WindowListenerExample()
       {
         addWindowListener(this);
       }
       public void windowClosing(WindowEvent e)
       {
         System.out.println("Window Closing");
         dispose();
         System.exit(0);
       }
       public void windowOpened(WindowEvent e)
         { System.out.println("Window Open"); }
       public void windowClosed(WindowEvent e)
         { System.out.println("Window Closed");}
       public void windowActivated(WindowEvent e)
         { System.out.println("Window Activated"); }
       public void windowDeactivated(WindowEvent e)
         { System.out.println("Window Deactivated"); }
       public void windowIconified(WindowEvent e)
         { System.out.println("window Iconified"); }
       public void windowDeiconified(WindowEvent e)
         { System.out.println("Window Deiconified"); }
     }
     public class MouseListenerExample extends Frame implements
MouseListener{
       Label l;
       MouseListenerExample(){
          addMouseListener(this);
```

```
        l=new Label();
        l.setBounds(20,50,100,20);
        add(l);
        setSize(300,300);
        setLayout(null);
        setVisible(true);
    }
    public void mouseClicked(MouseEvent e) {
        l.setText("Mouse Clicked");
    }
    public void mouseEntered(MouseEvent e) {
        l.setText("Mouse Entered");
    }
    public void mouseExited(MouseEvent e) {
        l.setText("Mouse Exited");
    }
    public void mousePressed(MouseEvent e) {
        l.setText("Mouse Pressed");
    }
    public void mouseReleased(MouseEvent e) {
        l.setText("Mouse Released");
    }
}
 class Example
{
    public static void main(String[] args)
    {
     WindowListenerExample frame = new WindowListenerExample();
     new MouseListenerExample();
     frame.setTitle("Window Listener Java Example");
     frame.setBounds(100,200,200,200);
     frame.setVisible(true);
    }
}
```

RESULT:-

The program is done and result is obtained successfully.

OUTPUT:-



PROGRAM-24

AIM:-

Develop a program to handle Key events.

CODE:-

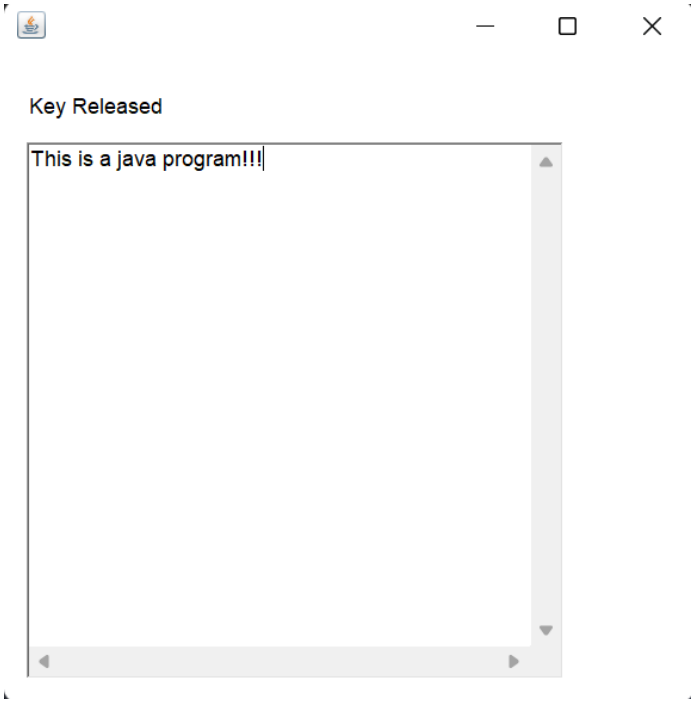
```
import java.awt.*;
import java.awt.event.*;
// class which inherits Frame class and implements KeyListener interface
public class KeyListenerExample extends Frame implements KeyListener {

// creating object of Label class   and TextArea class
 Label l;
    TextArea area;
// class constructor
    KeyListenerExample() {
          // creating the label
        l = new Label();
// setting the location of the label in frame
        l.setBounds (20, 50, 100, 20);
// creating the text area
        area = new TextArea();
// setting the location of text area
        area.setBounds (20, 80, 300, 300);
// adding the KeyListener to the text area
        area.addKeyListener(this);
// adding the label and text area to the frame
        add(l);
add(area);
// setting the size, layout and visibility of frame
        setSize (400, 400);
        setLayout (null);
        setVisible (true);
    }
// overriding the keyPressed() method of KeyListener interface where we set
the text of the label when key is pressed
    public void keyPressed (KeyEvent e) {
        l.setText ("Key Pressed");
```

```
    }
// overriding the keyReleased() method of KeyListener interface where we se
t the text of the label when key is released
    public void keyReleased (KeyEvent e) {
        l.setText ("Key Released");
    }
// overriding the keyTyped() method of KeyListener interface where we set t
he text of the label when a key is typed
    public void keyTyped (KeyEvent e) {
        l.setText ("Key Typed");
    }
  // main method
    public static void main(String[] args) {
        new KeyListenerExample();
    }
}
```

RESULT:-

The program is done and result is obtained successfully.

OUTPUT:-



PROGRAM-25

AIM:-

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

CODE:-

```
import java.io.FileWriter;
import java.io.IOException;
import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;

class Main {
   static String data="";
   public static void main(String[] arg) {
      // Read
      try {
         File dataFile = new File("DATA.txt");
         Scanner dataRead = new Scanner(dataFile);
         while (dataRead.hasNextLine()) {
            data += dataRead.nextLine();
            data +="\n";
         }
         dataRead.close();
      } catch (FileNotFoundException ex) {
         System.out.println("An error occured !");
         ex.printStackTrace();
      }
      // Write
      try {
         FileWriter dataWriter = new FileWriter("DATA2.txt");
         dataWriter.write(data);
         dataWriter.close();
      } catch (IOException ex) {
         System.out.println("An error occured !");
         ex.printStackTrace();
      }

   }
```

```
}
```

RESULT:-

The program is done and result is obtained successfully.

OUTPUT:-

```
Hiii
Hello
World
```

PROGRAM-26

AIM:-

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

CODE:-

```
import java.io.FileWriter;
import java.io.IOException;
import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;

class Main {
   static String data = "";
   static File dataFile = new File("DATA.txt");

   public static void main(String[] arg) {
      try {
         FileWriter oddFile = new FileWriter("odd.txt", true);
         FileWriter evenFile = new FileWriter("even.txt", true);

         Scanner dataRead = new Scanner(dataFile);
         while (dataRead.hasNextLine()) {
            data += dataRead.nextLine();
         }
         dataRead.close();
         String values[] = data.split(" ");
         int valuesInt[] = new int[values.length + 1];
         int count = 0;
         for (String i : values) {
            valuesInt[count++] = Integer.parseInt(i);
            //System.out.println(Integer.parseInt(i));
            if (Integer.parseInt(i) % 2 == 0) {
               evenFile.write(i+" ");

            } else {
               oddFile.write(i+" ");
            }
         }
```

```
            oddFile.close();
            evenFile.close();
        } catch (IOException ex) {
            System.out.println("An error occured !");
            ex.printStackTrace();
        } catch (Exception ex) {
            System.out.print("An error occured : ");
            System.out.println(ex.getMessage());
        }

    }
}
```

RESULT:-

The program is done and result is obtained successfully.

OUTPUT:-

```
Even: 2 4
Odd: 1 3 5
```

PROGRAM-27

AIM:-

Client server communication using Socket – TCP/IP

CODE:-

Client.java

```
// A Java program for a Client
import java.net.*;
import java.io.*;

public class Client
{
	// initialize socket and input output streams
	private Socket socket		= null;
	private DataInputStream input = null;
	private DataOutputStream out	= null;

	// constructor to put ip address and port
	public Client(String address, int port)
	{
		// establish a connection
		try
		{
			socket = new Socket(address, port);
			System.out.println("Connected");

			// takes input from terminal
			input = new DataInputStream(System.in);

			// sends output to the socket
			out = new DataOutputStream(socket.getOutputStream());
		}
		catch(UnknownHostException u)
		{
			System.out.println(u);
		}
		catch(IOException i)
```

```
            {
                System.out.println(i);
            }

            // string to read message from input
            String line = "";

            // keep reading until "Over" is input
            while (!line.equals("Over"))
            {
                try
                {
                    line = input.readLine();
                    out.writeUTF(line);
                }
                catch(IOException i)
                {
                    System.out.println(i);
                }
            }

            // close the connection
            try
            {
                input.close();
                out.close();
                socket.close();
            }
            catch(IOException i)
            {
                System.out.println(i);
            }
        }

        public static void main(String args[])
        {
            Client client = new Client("127.0.0.1", 5000);
        }
}
```

Server.java

```
// A Java program for a Server
import java.net.*;
import java.io.*;

public class Server
{
	//initialize socket and input stream
	private Socket		socket = null;
	private ServerSocket server = null;
	private DataInputStream in	= null;

	// constructor with port
	public Server(int port)
	{
		// starts server and waits for a connection
		try
		{
			server = new ServerSocket(port);
			System.out.println("Server started");

			System.out.println("Waiting for a client ...");

			socket = server.accept();
			System.out.println("Client accepted");

			// takes input from the client socket
			in = new DataInputStream(
				new BufferedInputStream(socket.getInputStream()));

			String line = "";

			// reads message from client until "Over" is sent
			while (!line.equals("Over"))
			{
				try
				{
					line = in.readUTF();
					System.out.println(line);
```

```
                        }
                        catch(IOException i)
                        {
                                System.out.println(i);
                        }
                }
                System.out.println("Closing connection");

                // close connection
                socket.close();
                in.close();
        }
        catch(IOException i)
        {
                System.out.println(i);
        }
    }

    public static void main(String args[])
    {
        Server server = new Server(5000);
    }
}
```

RESULT:-

The program is done and result is obtained successfully.

OUTPUT:-

```
Server started
Waiting for a client ...
```