

PROGRAM 1

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

ALGORITHM:

Step 1: Start

Step 2 : Define a class having name Product and members as pcode, pname and price

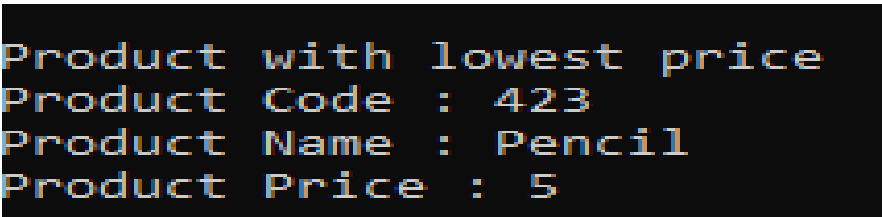
Step 3: Declare three objects in the class and add the values of each data members into objects.

Step 4: Using if condition check which object the lowest price and print it.

Step 5 stop

SOURCE CODE:

```
class product
{
    int pcode,price;
    String name;
    product(int a,int b,String s)
    {
        this.pcode=a;
        this.price=b;
        this.name=s;
    }
    public static void main(String args[])
    {
        product obj=new product(101,5,"Tea");
        product obj1=new product(102,10,"Coffee");
        product obj2=new product(111,200,"Biscuits");
        if((obj.price<obj1.price)&&(obj.price<obj2.price))
            System.out.println("Product with low price:"+obj.name);
        else if(obj1.price<obj2.price)
            System.out.println("Product with low price:"+obj1.name);
        else
            System.out.println("Product with low price:"+obj2.name);
    }
}
```

OUTPUT:


```
Product with lowest price
Product Code : 423
Product Name : Pencil
Product Price : 5
```

PROGRAM 2

AIM: Read 2 matrices from the console and perform matrix addition.

ALGORITHM:

Step 1:Start

Step 2 :Define a class having name Matrix

Step 3: Read rows, number (m), column number (n) and Initialize the two dimensional arrays a[],b[],c[] with same row number,column number.

Step 4: Store the first matrix elements into the 2D array matrix a[] using two for loops: i indicates row number, j indicates column number. Similarly second matrix b[].

Step 5:: Add the two matrices using for loop.
 for i=0 to i<m
 for j=0 to j<n
 a[i][j] + b[i][j] and store it into the matrix c[i][j]

Step 6: Print Sum of matrices c[i][j]

Step 7: Stop

SOURCE CODE:

```
import java.util.Scanner;
class Matrix
{
    int m,n,i,j;
    int[][] a = new int[5][5];
    int[][] b = new int[5][5];
    int[][] c = new int[5][5];
    Scanner myObj = new Scanner(System.in);
    void get()
    {
        System.out.println("Enter no: of rows : ");
        m = myObj.nextInt();
        System.out.println("Enter no: of column : ");
        n = myObj.nextInt();
        System.out.println("Enter elements of matrix A : ");
        for(i=0;i<m;i++)
            for(j=0;j<n;j++)
                a[i][j]=myObj.nextInt();
        System.out.println("Enter elements of matrix B : ");
        for(i=0;i<m;i++)
            for(j=0;j<n;j++)
                b[i][j]=myObj.nextInt();
    }
    void add()
    {
```

```

        for(i=0;i<m;i++)
            for(j=0;j<n;j++)
                c[i][j]=a[i][j]+b[i][j];
    }
    void display()
    {
        System.out.println("Matrix A : ");
        for(i=0;i<m;i++)
        {
            for(j=0;j<n;j++)
                System.out.print(a[i][j]+" ");
            System.out.println();
        }
        System.out.println("Matrix B : ");
        for(i=0;i<m;i++)
        {
            for(j=0;j<n;j++)
                System.out.print(b[i][j]+" ");
            System.out.println();
        }
        System.out.println("Matrix C : ");
        for(i=0;i<m;i++)
        {
            for(j=0;j<n;j++)
                System.out.print(c[i][j]+" ");
            System.out.println();
        }
    }
}
public class Matrixes
{
    public static void main(String[] args)
    {
        Matrix ob = new Matrix();
        ob.get();
        ob.add();
        ob.display();
    }
}

```

OUTPUT:

```

Enter no: of rows :
2
Enter no: of column :
2
Enter elements of matrix A :
2 3 4 5
Enter elements of matrix B :
6 7 8 9
Matrix A :
2 3
4 5
Matrix B :
6 7
8 9
Matrix C :
8 10
12 14

```

PROGRAM 3**AIM:** Add complex numbers**ALGORITHM:**

Step 1: Start

Step 2: Define a class having name Complex and data members r,i

Step 3: Define a function Complex number and add values to variables.

Step 4: Define a function add, to add Complex Number
Using 3rd Complex number object and return the Value.

Step 5: Print the sum value

Step 6: Stop.

SOURCE CODE:

```

import java.util.*;
class com
{
    int real,imaginary;
    com()
    {
    }
    com(int realtemp,int imaginarytemp)
    {
        real=realtemp;
        imaginary=imaginarytemp;
    }
    com addcomplex(com ip1,com ip2)
    {
        com temp=new com();
        temp.real=ip1.real+ip2.real;
        temp.imaginary=ip1.imaginary+ip2.imaginary;
        return temp;
    }
}
class complex
{
    public static void main(String args[])
    {
        com ip1=new com(15,24);
        System.out.println("The 1st complex no. is defined as:"+ip1.real+"+i"+ip1.imaginary);
        com ip2=new com(3,7);
        System.out.println("The 2nd complex no. is defined as:"+ip2.real+"+i"+ip2.imaginary);
        com result=new com();
        result=result.addcomplex(ip1,ip2);
        System.out.println("The sum of complex no. is:"+result.real+"+i"+result.imaginary);
    }
}

```

OUTPUT:

```
The 1st complex no. is defined as:15+i24  
The 2nd complex no. is defined as:3+i7  
The sum of complex no. is:18+i31
```

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:

Step 1: start

Step 2: Define a class cpu with data member price and class processor.

Step 3: Class processor contain data members cores, producer and a nested class RAM.

Step 4: Class RAM contain memory, manuf as data members

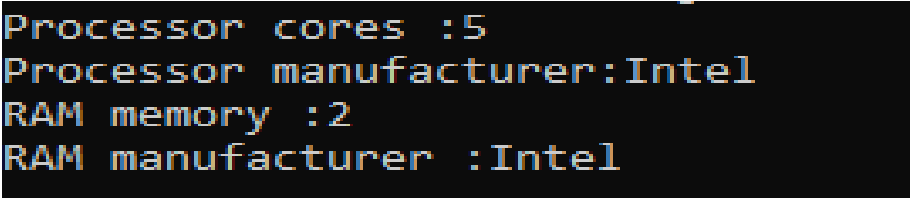
Step 5: Create object in corresponding classes and display it's details

Step 6: Stop

SOURCE CODE:

```
class CPU
{
    double price;
    CPU(int p)
    {
        price = p;
    }
    class processor
    {
        int cores;
        String manufacturer;
        processor(int c,String manu)
        {
            cores=c;
            manufacturer=manu;
        }
        void display()
        {
            System.out.println("Processor cores :"+cores);
            System.out.println("Processor manufacturer:"+manufacturer);
        }
    }
    static class RAM
    {
        int memory;
        String manufacturer;
        RAM(int m,String Manufact)
        {
            memory=m;
            manufacturer=Manufact;
        }
    }
}
```

```
void disp()
    {
        System.out.println("RAM memory :"+memory);
        System.out.println("RAM manufacturer :"+manufacturer);
    }
}
public class Main
{
    public static void main(String[] args)
    {
        CPU cpu=new CPU(30000);
        CPU.processor processor1=cpu.new processor(5,"Intel");
        CPU.RAM ram=new CPU.RAM(2,"Intel");
        processor1.display();
        ram.disp();
    }
}
```

OUTPUT:A screenshot of a terminal window with a black background and white text. The output consists of four lines: 'Processor cores :5', 'Processor manufacturer: Intel', 'RAM memory :2', and 'RAM manufacturer : Intel'.

```
Processor cores :5
Processor manufacturer: Intel
RAM memory :2
RAM manufacturer : Intel
```

PROGRAM 5

AIM: Program to Sort strings.

ALGORITHM:

Step 1 Start

Step 2: Select the first element of the list (ie, element at first position in the list).

Step 3: Compare the selected element with all the other elements in the list.

Step 4: In every comparison, if any element is found smaller than the selected element (for ascending order), then both are swapped.

Step 5: Repeat the same procedure with element in the next position in the list till the entire list is sorted.

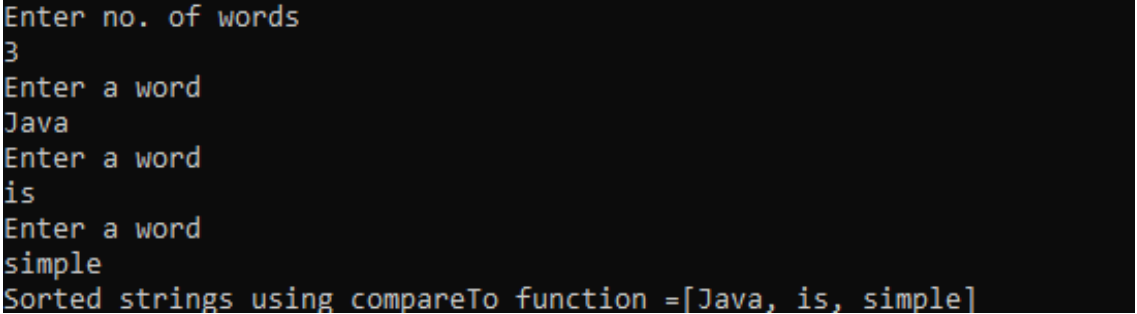
Step 6: Stop.

SOURCE CODE:

```
import java.util.Scanner;
import java.util.Arrays;
public class sort
{
    public static void main(String args[])
    {
        int i,j;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter no. of words");
        int n=sc.nextInt();
        String word[]=new String[n];
        sc.nextLine();
        for(i=0;i<n;i++)
        {
            System.out.println("Enter a word");
            word[i]=sc.nextLine();
        }
        for(i=0;i<n-1;i++)
        {
            for(j=i+1;j<n;j++)
            {
                if(word[i].compareTo(word[j])>0)
                {
                    String t=word[i];
                    word[i]=word[j];
                    word[j]=t;
                }
            }
        }
        System.out.println("Sorted strings using compareTo function =" +Arrays.toString(word));
    }
}
```



```
System.out.println(word);  
}  
}
```

OUTPUT:A screenshot of a terminal window with a black background and white text. The text shows the execution of a Java program. It starts with a prompt 'Enter no. of words' followed by the input '3'. Then it prompts 'Enter a word' three times, with inputs 'Java', 'is', and 'simple' respectively. Finally, it displays the output 'Sorted strings using compareTo function =[Java, is, simple]'.

```
Enter no. of words  
3  
Enter a word  
Java  
Enter a word  
is  
Enter a word  
simple  
Sorted strings using compareTo function =[Java, is, simple]
```

PROGRAM 6**AIM:** Perform string manipulations**ALGORITHM:**

Step 1: Start

Step 2: Take the string provided by the user

Step 3: Display the length of the string

Step 4: Display the first character of the string

Step 5: Display the string with lowercase

Step 6: Display the String with uppercase

Step 7: Display the string after replacing the 'ca' with 'Kol'

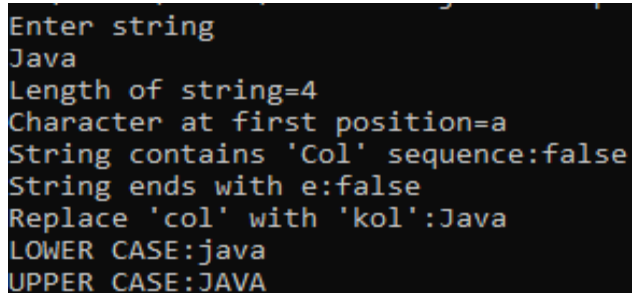
Step 8: Stop

SOURCE CODE:

```

import java.util.Scanner;
public class manipulation
{
    public static void main(String args[])
    {
        System.out.println("Enter string");
        Scanner sc=new Scanner(System.in);
        String s1=sc.nextLine();
        System.out.println("Length of string="+s1.length());
        System.out.println("Character at first position="+s1.charAt(1));
        System.out.println("String contains 'Col' sequence:"+s1.contains("Col"));
        System.out.println("String ends with e:"+s1.endsWith("e"));
        System.out.println("Replace 'col' with 'kol':"+s1.replaceAll("Col","kol"));
        System.out.println("LOWER CASE:"+s1.toLowerCase());
        System.out.println("UPPER CASE:"+s1.toUpperCase());
    }
}

```

OUTPUT:


```

Enter string
Java
Length of string=4
Character at first position=a
String contains 'Col' sequence:false
String ends with e:false
Replace 'col' with 'kol':Java
LOWER CASE:java
UPPER CASE:JAVA

```

PROGRAM 7

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

ALGORITHM:

Step 1: Start

Step 2: Search the 'eno' attribute of the list of employee objects for the 'eno' provided by the user

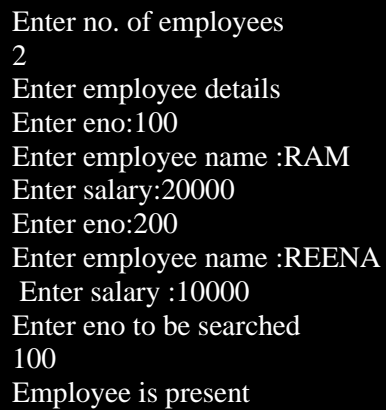
Step 3: If user provided 'eno' is found inside the employee object list, display the details of the corresponding employee

Step 4: Stop

SOURCE CODE:

```
import java.util.*;
class Main
{
    public static void main(String args[])
    {
        int count,eno,salary,k;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter no. of employees");
        count=sc.nextInt();
        employee[] e= new employee[count];
        System.out.println("Enter employee details");
        for(int i=0;i<count;i++)
        {
            System.out.println("Enter eno:");
            eno=sc.nextInt();
            System.out.print("Enter employee name : ");
            String ename = sc.nextLine();
            System.out.print("Enter salary : ");
            salary = sc.nextInt();
            e[i]=new employee(eno,ename,salary);
        }
        System.out.println("Enter eno to be searched");
        k=sc.nextInt();
        int flag=0;
        for(int i=0;i<count;i++)
        {
            if(k==e[i].eno)
            {
                flag=1;
                break;
            }
        }
        if(flag==0)
            System.out.println("Employee is present");
```

```
else
System.out.println("Employee is not present");
}
}
class employee
{
    int eno;
    String ename;
    int salary;
    employee(int eno,String ename,int salary)
    {
        eno=eno;
        ename=ename;
        salary=salary;
    }
}
```

OUTPUT:A screenshot of a terminal window with a black background and white text. The text shows the execution of a Java program. It starts with a prompt 'Enter no. of employees' followed by the input '2'. Then it prompts 'Enter employee details'. The first employee is entered with 'Enter eno:100', 'Enter employee name :RAM', and 'Enter salary:20000'. The second employee is entered with 'Enter eno:200', 'Enter employee name :REENA', and 'Enter salary :10000'. Finally, it prompts 'Enter eno to be searched' with the input '100', and the output 'Employee is present' is displayed.

Enter no. of employees
2
Enter employee details
Enter eno:100
Enter employee name :RAM
Enter salary:20000
Enter eno:200
Enter employee name :REENA
Enter salary :10000
Enter eno to be searched
100
Employee is present

PROGRAM 8

AIM: Area of different shapes using overloaded functions.

ALGORITHM:

Step 1: Start

Step 2: Define the main class

Step 3: Define methods with the same methodname that performs the area operation for each shape

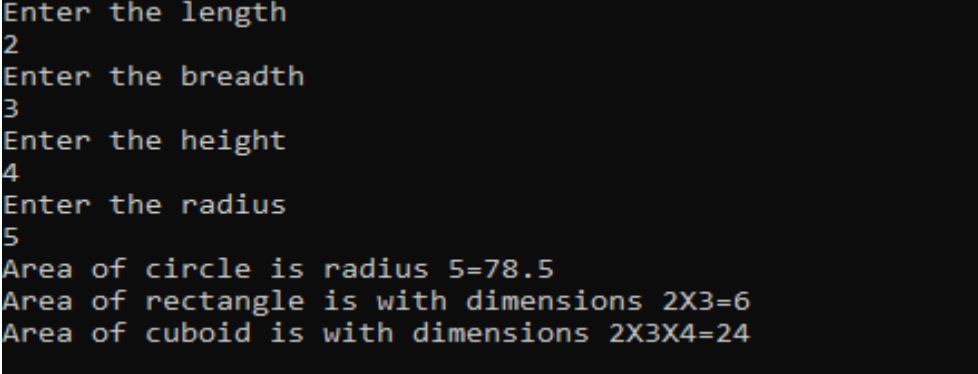
Step 4: Display the areas of each shapes.

Step 5: Stop

SOURCE CODE:

```
import java.util.Scanner;
public class shapes
{
    void area(int r1)
    {
        double area=3.14*r1*r1;
        System.out.println("Area of circle is radius" + r1+"="+area);
    }
    void area(int a1,int b1)
    {
        int area=a1*b1;
        System.out.println("Area of rectangle is with dimensions" + a1+"X"+b1+"="+area);
    }
    void area(int a1,int b1,int c1)
    {
        int area=a1*b1*c1;
        System.out.println("Area of cuboid is with dimensions" +
a1+"X"+b1+"X"+c1+"="+area);
    }
    public static void main(String args[])
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the length");
        int l=sc.nextInt();
        System.out.println("Enter the breadth");
        int b=sc.nextInt();
        System.out.println("Enter the height");
        int h=sc.nextInt();
        System.out.println("Enter the radius");
        int r=sc.nextInt();
        shapes s=new shapes();
        s.area(r);
        s.area(l,b);
    }
}
```

```
s.area(l,b,h);  
}  
}
```

OUTPUT:A screenshot of a terminal window with a black background and light blue/green text. It shows the execution of a program where the user is prompted to enter dimensions for a circle, rectangle, and cuboid. The inputs are 2, 3, 4 for the rectangle and 5 for the circle. The program then calculates and displays the areas for each shape.

```
Enter the length  
2  
Enter the breadth  
3  
Enter the height  
4  
Enter the radius  
5  
Area of circle is radius 5=78.5  
Area of rectangle is with dimensions 2X3=6  
Area of cuboid is with dimensions 2X3X4=24
```

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

ALGORITHM:

Step 1: Start

Step 2: create class "employee" with the provided data members and define the constructors

Step 3: create another class "Teachers" that performs inheritance of employee class and define constructors for the same

Step 4: create an array of objects in the corresponding class

Step 5: Display the details for the number of teachers provided

Step 6: Stop

SOURCE CODE:

```
import java.util.Scanner;
class employee
{
    int id;
    String name,address;
    double salary;
    employee(int n,String na,double s,String a)
    {
        this.id=n;
        this.name=na;
        this.salary=s;
        this.address=a;
    }
}
class teacher extends employee
{
    String dept;
    String subject;
    teacher(int n,String na,double s,String a,String d,String sub)
    {
        super(n,na,s,a);
        this.dept=d;
        this.subject=sub;
    }
}
```

```

void display()
{
    System.out.println("Employee id:"+id);
    System.out.println("Employee name:"+name);
    System.out.println("Employee salary:"+salary);
    System.out.println("Employee address:"+address);
    System.out.println("Department:"+dept);
    System.out.println("Subject:"+subject);
}
public static void main(String args[])
{
    System.out.println("Enter no. of employees");
    Scanner sc=new Scanner(System.in);
    int num=sc.nextInt();
    teacher t[]=new teacher[num];
    for(int i=0;i<num;i++)
    {
        Scanner sc1=new Scanner(System.in);
        System.out.println("Enter employee id");
        int empid=sc1.nextInt();
        System.out.println("Enter employee name");
        String name=sc1.next();
        System.out.println("Enter employee salary");
        double salary=sc1.nextDouble();
        System.out.println("Enter employee address");
        String address=sc1.next();
        System.out.println("Enter department");
        String dept=sc1.next();
        System.out.println("Enter subject");
        String subject=sc1.next();
        t[i]=new teacher(empid,name,salary,address,dept,subject);
        //sc1.close();
    }
    System.out.println("*****Informations of all the employees*****");
    for(int i=0;i<num;i++)
    {
        int j=i+1;
        System.out.println("\n"+j+").");
        t[i].display();
    }
    sc.close();
}
}

```


OUTPUT:

```
Enter no. of employees
2
Enter employee id
100
Enter employee name
RAM
Enter employee salary
10000
Enter employee address
KANNUR
Enter department
SALES
Enter subject
BCA
Enter employee id
200
Enter employee name
REENA
Enter employee salary
20000
Enter employee address
KOLLAM
Enter department
MCA
Enter subject
IT
*****Informations of all the employees*****
1).
Employee id:100
Employee name:RAM
Employee salary:10000.0
Employee address:KANNUR
Department:SALES
Subject:BCA
2).
Employee id:200
Employee name:REENA
Employee salary:20000.0
Employee address:KOLLAM
Department:MCA
Subject:IT
```

PROGRAM 10

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:

Step 1: Start

Step 2: To create a package named graphics, create a folder of the same name in the directory. Here inside that we have another module named calculate

Step 3: Inside the graphics folder, create modules for finding the areas of rectangle, circle, triangle and square.

Step 4: Outside the graphics folder, write a program to access the modules mentioned above and print the output

Step 5: Stop

SOURCE CODE:**DRIVER**

```
import graphics.circle;
import graphics.rectangle;
import graphics.square;
import graphics.triangle;
import java.util.Scanner;
class driver
{
public static void main(String[] args)
{
Scanner sc = new Scanner(System.in);
int choice;
circle obj1 = new circle();
rectangle obj2 = new rectangle();
square obj3 = new square();
triangle obj4 = new triangle();
System.out.println("Choose any\n 1)Circle\n 2)Rectangle\n 3)Square\n 4)Triangle:\n 5)Exit ");
choice = sc.nextInt();
switch (choice)
{
case 1:
obj1.area();
break;
case 2:
obj2.area();
break;
case 3:
obj3.area();
break;
case 4:
obj4.area();
```

```

case 5:
System.exit(0);
default:
break;
}
}
}

```

CIRCLE

```

package graphics;
import java.util.Scanner;
public class circle implements area_cal
{
    int radius;
    @Override
    public void area()
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Input radius of circle : ");
        radius = sc.nextInt();
        String area = Double.toString(Math.PI*radius*radius);
        System.out.println("Area of the circle is : "+area);
        sc.close();
    }
}

```

RECTANGLE

```

package graphics;
import java.util.Scanner;
public class rectangle implements area_cal
{
    int l,b;
    @Override
    public void area()
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the length of the rectangle :");
        l = sc.nextInt();
        System.out.println("Enter the breath of the rectangle");
        b = sc.nextInt();
        System.out.println("Area of the rectangle = "+l*b);
    }
}

```

SQUARE

```

package graphics;
import java.util.Scanner;
public class square implements area_cal
{
    int side;
    @Override
    public void area()
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Input side length of square : ");
    }
}

```

```

side = sc.nextInt();
String area = Double.toString(side*side);
System.out.println("Area of the square : "+area);
}
}

```

TRIANGLE

```

package graphics;
import java.util.Scanner;
public class triangle implements area_cal
{
int height;
int breadth;
@Override
public void area()
{
Scanner sc = new Scanner(System.in);
System.out.println("Input height of the triangle : ");
height = sc.nextInt();
System.out.println("Input breadth of triangle : ");
breadth = sc.nextInt();
String area = Double.toString((height*breadth)/2f);
System.out.println("Area of the triangle is : "+area);
}
}

```

AREA_CAL

```

package graphics;
public interface area_cal
{
void area();
}

```

OUTPUT:

```

Choose any
1)Circle
2)Rectangle
3)Square
4)Triangle:
5)Exit
1
Input radius of circle :
5
Area of the circle is : 78.53981633974483

```

PROGRAM 11

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

ALGORITHM:

Step 1: Start

Step 2: Create a class named usernameex that inherits Exception class with a constructor that calls Exception class constructor and pass error message.

Step 3: Create a class named passwordex that inherits Exception class with a constructor that calls Exception class constructor and pass error message.

Step 4: Inside the main(), Read the username and password.

Step 5: Inside the try block, we throw usernameex and passwordex with appropriate message if any of the condition is true:

- a. If username is empty
- b. If password is empty
- c. If password doesn't contain special characters
- d. If username length is less than 6
- e. If password is not strong enough

Step 6: Inside the catch block with parameter usernameex's object, print "USERNAMEEXCEPTION OCCURED"

Step 7: Inside the catch block with parameter passwordex's object, print "PASSWORDEXCEPTION OCCURED"

Step 8: Stop

SOURCE CODE:**AUTHENTICATION**

```
import java.util.Scanner;
import Excep.*;
public class authentication
{
    public static void main(String args[])
    {
        String username="Tom";
        String password="Cat";
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the username");
        String u1=sc.nextLine();
        System.out.println("Enter the password");
        String u2=sc.nextLine();
```

```

        try
        {
            if((u1.equals(username))&&(u2.equals(password)))
            {
                System.out.println("Access Granted");
            }
            else
            {
                throw new credentialexception("Invalid credentials");
            }
        }
        catch(credentialexception e)
        {
            System.out.println(e.getMessage());
        }
    }
}

```

CREDENTIALEXCEPTION

```

package Excep;
public class credentialexception extends Exception
{
    public credentialexception(String s)
    {
        super(s);
    }
}

```

OUTPUT:

```

Enter the username
Tom
Enter the password
Cat
Access Granted

```

PROGRAM 12

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:

Step 1: Start

Step 2: Create an interface Calculation that has the methods to take inputs and compute area and perimeter

Step 3: Create classes Circle and Rectangle that implements calculation

Step 4: Display the area and perimeter of circle or rectangle depending upon the choice the user selects.

Step 5: Stop

SOURCE CODE:

```
import java.util.Scanner;
interface prop
{
    void getdata(); void area();
    void perimeter();
}
class Circle implements prop
{
    double pi = 3.14; double r;
    Scanner sc = new Scanner(System.in);
    @Override
    public void getdata()
    {
        System.out.println("Enter the radius of the circle:");
        r = sc.nextDouble();
    }
    @Override
    public void perimeter()
    {
        System.out.println("Perimeter of the circle: "+(2*pi*r));
    }
    @Override
    public void area()
    {
        System.out.println("Perimeter of the circle: "+(pi*r*r));
    }
}
class Rectangle implements prop
{
    double l,b;
    Scanner sc = new Scanner(System.in);
```

```

@Override
public void getdata()
{
    System.out.println("Enter the length of the rectangle:");
    l = sc.nextDouble();
    System.out.println("Enter the breadth of the rectangle:");
    b = sc.nextDouble();
}
@Override
public void area()
{
    System.out.println("Perimeter of a rectangle: "+(l*b));
}
@Override
public void perimeter()
{
    System.out.println("Perimeter of a rectangle: "+(2*(l+b)));
}
}
public class menu
{
    public static void main(String[] args)
    {
        int ch;
        Scanner sc = new Scanner(System.in);
        Circle ob = new Circle();
        Rectangle obj = new Rectangle();
        do
        {
            System.out.println("\n1.Circle\n2.Rectangle\n3.exit"); System.out.println("Enter your choice:");
            ch = sc.nextInt();
            switch(ch)
            {
                case 1 :
                    ob.getdata();
                    ob.area();
                    ob.perimeter();
                    break;
                case 2 :
                    obj.getdata();
                    obj.area();
                    obj.perimeter();
                    break;
                case 3:
                    System.out.println("Exited....");
                    System.exit(0);
            }
        }
        while(true);
    }
}

```


OUTPUT:

```
1.Circle
2.Rectangle
3.exit
Enter your choice:
1
Enter the radius of the circle:
3
Perimeter of the circle: 28.259999999999998
Perimeter of the circle: 18.84

1.Circle
2.Rectangle
3.exit
Enter your choice:
2
Enter the length of the rectangle:
5
Enter the breadth of the rectangle:
4
Perimeter of a rectangle: 20.0
Perimeter of a rectangle: 18.0

1.Circle
2.Rectangle
3.exit
Enter your choice:
3
Exited....
```

PROGRAM 13

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

ALGORITHM:

Step 1: Start

Step 2: Read a number from user at run time.

Step 3: Throw an exception if the entered number is negative.

Step 4: Add to an array otherwise.

Step 5: Find the sum of the positive numbers and display the average.

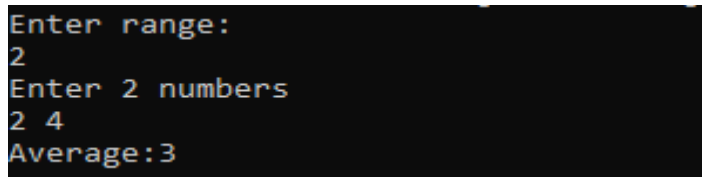
Step 6: Print the sum.

Step 7: Stop.

SOURCE CODE:

```
import java.util.*;
class EntryNegative extends Exception
{
private int n; EntryNegative(int a)
{
n=a;
}
public String toString()
{
return "EntryNegativeException number should be positive";
}
}
class mainNegativeException
{
static int r,s,sum=0;
static int count,avg;
static int entry(int n,int ra) throws EntryNegative
{
r=n; s=ra;
for (int i=0;i<s;i++)
{
if(n<0)
{
throw new EntryNegative(n);
}
else
{
sum+=n;
count+=1;
}
}
```

```
}
avg=sum/count;
return avg;
}
public static void main(String args[])
{
int num,range;
Scanner sc=new Scanner(System.in);
System.out.println("Enter range:");
range=sc.nextInt();
System.out.println("Enter "+range+" numbers");
int a[]=new int[range];
for(int i=0;i<range;i++)
{
try
{
a[i]=sc.nextInt();
avg=entry(a[i],range);
}
catch(EntryNegative e)
{
System.out.println("Caught "+e);
}
}
System.out.println("Average:"+avg);
}
}
```

OUTPUT:A screenshot of a terminal window showing the output of the Java program. The text is as follows:

```
Enter range:
2
Enter 2 numbers
2 4
Average:3
```

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:

Step 1: Start

Step 2: Create a class named even that implements Runnable interface with function run()

Step 3: Inside run(), we read the limit for printing even numbers and print it using for loop.

Step 4: Create another class fib that implements Runnable interface with function run().

Step 5: Inside run(), Initialise n1 as 0, n2 as 1 and n3 as 0.

Step 6: Check if n<0, if true, print "Enter a positive number" else goto step 7

Step 7: Repeat step 8 to 11 until n3>n

Step 8: Print n1

Step 9: n3=n1+n2

Step 10: n1=n2

Step 11: n2=n3

Step 12: Create object e of even and create an object t1 of Thread with its parameterized constructor passing e as parameter

Step 13: Call start() using t1

Step 14: Do the same for class odd with Thread object t2 and call start() using t2

Step 15: Stop

SOURCE CODE:**DRIVER**

```
import java.util.Scanner;
public class fibeven
{
    public static void main(String args[])
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the lower limit of range : ");
        int x=sc.nextInt();
        System.out.println("Enter the upper limit of range : ");
        int y=sc.nextInt();
        Runnable r=new fibannoci(x,y);
```

```

        Thread obj1=new Thread(r);
        obj1.start();
        Runnable p=new even(x,y);
        Thread obj2=new Thread(p);
        obj2.start();
    }
}
class even implements Runnable
{
    int n1,n2;
    even(int x, int y)
    {
        n1=x;
        n2=y;
    }
    @Override
    public void run()
    {
        for(int i=n1;i<=n2;i++)
        {
            if(i%2==0)
            {
                System.out.println("Even =" +i);
            }
        }
    }
}
class fibannoci implements Runnable
{
    int n1,n2,num=0,x=0,y=1;
    fibannoci(int l, int u)
    {
        n1=l;
        n2=u;
    }
    @Override
    public void run()
    {
        System.out.println("Fibannoci =" +0);
        System.out.println("Fibannoci =" +1);
        while(num<n2-1)
        {
            num=x+y;
            if((num>=n1)&&(num<=n2))
            {
                System.out.println("Fibannoci =" +num);
            }
            x=y;
            y=num;
        }
    }
}

```

FIBONACCI

```

public class fibannoci implements Runnable
{
    int n1,n2,num=0,x=0,y=1;
    fibannoci(int l, int u)
    {
        n1=l;
        n2=u;
    }
    @Override
    public void run()
    {
        System.out.println("Fibannoci =" +0);
        System.out.println("Fibannoci =" +1);
        while(num<n2-1)
        {
            num=x+y;
            if((num>=n1)&&(num<=n2))
            {
                System.out.println("Fibannoci =" +num);
            }
            x=y;
            y=num;
        }
    }
}

```

EVEN

```

public class even implements Runnable
{
    int n1,n2;
    even(int x, int y)
    {
        n1=x;
        n2=y;
    }
    @Override
    public void run()
    {
        for(int i=n1;i<=n2;i++)
        {
            if(i%2==0)
            {
                System.out.println("Even =" +i);
            }
        }
    }
}

```

OUTPUT:

```
Enter the lower limit of range :  
2  
Enter the upper limit of range :  
5  
Fibannoci =0  
Fibannoci =1  
Fibannoci =2  
Fibannoci =3  
Fibannoci =5  
Even =2  
Even =4
```

PROGRAM 15

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

ALGORITHM:

Step 1: Start

Step 2: Create a class named stack with data members as a(an array),top(set as -1),ch,item,i; afunction named menu()

Step 3: Inside menu(), give choices to push,pop and display the stack

Step 4: If the choice is 1, then check whether the stack is full, else add an element into thestack.

Step 5: If the choice is 2, then check whether the stack is empty, else delete an element intothe stack.

Step 6: If the choice is 3, then check whether the stack is empty, else print all the elements inthe stack.

Step 7: If the choice is greater than 4, then print "Invalid option".

Step 8: Inside the main(), create an object of type stack and call the menu() function.Step 9:Stop

SOURCE CODE:

```
import java.util.*;
class operations
{
    public void operation()
    {
        int top=-1,ch,n,e;
        Scanner s=new Scanner(System.in);
        System.out.println("Enter size of stack");
        n=s.nextInt();
        int size=n-1;
        int[] a=new int[n];
        do
        {
            System.out.println("\n-----\nMENU
:n1.PUSH\n2.POP\n3.DISPLAY\n4.EXIT\n-----");
            System.out.println("Enter your choice");
            ch=s.nextInt();
            switch(ch)
            {
                case 1:
                    if(top==size)
                    {
                        System.out.println("***Stack is full***");
                    }
                    else
```



```

        {
            System.out.println("Enter the element");
            e=s.nextInt();
            top++;
            a[top]=e;
        }
        break;
    case 2:
        if(top==0)
        {
            System.out.println("***Stack is empty***");
        }
        else
        {
            System.out.println("\n"+a[top]+"is removed");
            top--;
        }
        break;
    case 3:
        if(top==0)
        {
            System.out.println("***Stack is empty***");
        }
        else
        {
            System.out.println("\n***Stack:***\n");
            for(int i=top;i>=0;i--)
            {
                System.out.println(""+a[i]);
                System.out.println("-----");
            }
        }
        break;
    case 4:
        System.exit(0);
    default: System.out.println("Invalid choice");
}
}
while(ch!=4);
}
}
public class driver
{
    public static void main(String args[])
    {
        operations o=new operations();
        o.operation();
    }
}

```

OUTPUT:

```
Enter size of stack
2
MENU :
1.PUSH
2.POP
3.DISPLAY
4.EXIT
Enter your choice
1
Enter the element
2
MENU :
1.PUSH
2.POP
3.DISPLAY
4.EXIT
Enter your choice
1
Enter the element
3
MENU :
1.PUSH
2.POP
3.DISPLAY
4.EXIT
Enter your choice
2
3 is removed
MENU :
1.PUSH
2.POP
3.DISPLAY
4.EXIT
Enter your choice
3
Stack:
2
MENU :
1.PUSH
2.POP
3.DISPLAY
4.EXIT
Enter your choice
4
```

PROGRAM 16

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

ALGORITHM:

Step 1: Start

Step 2: Create a FileWriter object

Step 3: Use write() method to write the string into

the file. Step 4: Close the file

Step 5: Create FileReader to read from the file.

Step 6: Use function to read line by line and print the read string to console using System.out.println

Step 7: Stop.

SOURCE CODE:

```
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
public class FileReadWrite
{
    public static void main(String[] args)
    {
        try
        {
            FileWriter writer = new FileWriter("MyFile.txt", true);
            writer.write("Hello Welcome to JAVA PROGRAMMING LANGUAGE");
            FileReader reader = new FileReader("MyFiles.txt");
            int character;
            while ((character = reader.read()) != -1)
            {
                System.out.print((char) character);
            }
            reader.close(); writer.write("\r\n"); writer.write("Good Bye!");
            writer.close();
        }
        catch (IOException e)
        {
            e.printStackTrace();
        }
    }
}
```

OUTPUT:

```
C:\Users\admin\Documents\javarecord>javac FileReadWrite.java  
C:\Users\admin\Documents\javarecord>java FileReadWrite  
java is simple and it is powerful.
```

PROGRAM 17

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

ALGORITHM:

Step 1: Start

Step 2: Create Map object using HashMap.

Step 3: Insert key, value pairs into the map using put().

Step 4: Display the map

Step 5: Remove an entry using remove()

Step 6: Display the updated map

Step 7: Change value of a specific key using replace()

Step 8: Display the map.

SOURCE CODE:

```
import java.util.*;
public class Hashmap
{
    public static void main(String args[])
    {
        Map<Integer,
        String> hash_map=new HashMap<Integer,String>();
        hash_map.put(1,"Archana");
        hash_map.put(5,"Neethu ");
        hash_map.put(17,"Amrutha");
        hash_map.put(2,"Aparna");
        hash_map.put(3,"Aswani");
        System.out.println("Mappings are "+hash_map);
        String removed_value=(String)hash_map.remove(3);
        System.out.println("Removed value is "+removed_value);
        hash_map.replace(1,"Nandhitha");
        System.out.println("Updated Mappings are "+hash_map);
    }
}
```

OUTPUT:

```
Mappings are {1=Archana, 17=Amrutha, 2=Aparna, 3=Aswani, 5=Neethu }
Removed value is Aswani
Updated Mappings are {1=Nandhitha, 17=Amrutha, 2=Aparna, 5=Neethu }
```

PROGRAM 18

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

ALGORITHM:

Step 1: Start.

Step 2: Create a queue of type Queue using PriorityQueue.

Step 3: Insert elements using add().

Step 4: display the queue.

Step 5: Remove an element.

Step 6: poll() the queue display the queue.

Step 7: Get the top element using peek()

Step 8: Stop.

SOURCE CODE:

```
import java.util.*;
class PriorityQueue
{
    public static void main(String args[])
    {
        PriorityQueue<String> queue=new PriorityQueue<String>();
        System.out.println("1.Insertion\n2.display Top Element\n3.Print and Delete Top
        Element\n4.Display\n5.Exit\n");
        while(true)
        {
            System.out.println("Enter your choice");
            Scanner s=new Scanner(System.in);
            int ch=s.nextInt();
            switch(ch)
            {
                case 1:
                    System.out.println("Enter size");
                    int n=s.nextInt();
                    System.out.println("Enter the names");
                    for(int i=0;i<n;i++)
                    {
                        String name=s.next();
                        queue.add(name);
                    }
                    break;
                case 2:
                    System.out.println("Top element");
                    System.out.println("head:"+queue.peek());
                    break;
```

```

case 3:
System.out.println("Removed: "+queue.peek());
queue.poll();
break;
case 4:
System.out.println("iterating the queue elements:");
Iterator itr=queue.iterator();
while(itr.hasNext())
{
System.out.println(itr.next());
}
break;
case 5:
System.exit(0);
break;
default:
System.out.println("Invalid Choice");
break;
}
}
}
}

```

OUTPUT:

```

1.Insertion
2.display Top Element
3.Print and Delete Top Element
4.Display
5.Exit

Enter your choice
1
Enter size
2
Enter the names
ram reena
Enter your choice
2
Top element
head:ram
Enter your choice
3
Removed: ram
Enter your choice
4
iterating the queue elements:
reena
Enter your choice
5

```

PROGRAM 19

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

ALGORITHM:

Step 1: Start

Step 2: Create an ArrayList of Strings

Step 3: Insert elements using add().

Step 4: Display the list.

Step 5: Display an element from the list using get().

Step 6: Replace an element at given position using set().

Step 7: sort the list using Collections.sort().

Step 8: Stop.

SOURCE CODE:

```
import java.util.*;
class Array_list
{
    public static void main(String args[])
    {
        ArrayList<String> a=new ArrayList<String>();
        System.out.println("1. Insertion\n2. Remove\n3. Replace\n4. Display\n5. Get element by
        position\n6. Exit\n");
        while(true)
        {
            System.out.println("Enter your choice");
            Scanner s=new Scanner(System.in);
            int ch=s.nextInt();
            switch(ch)
            {
                case 1:
                    System.out.println("Enter the size:");
                    int n=s.nextInt();
                    System.out.println("Enter the elements:");
                    for(int i=0;i<n;i++)
                    {
                        String Arr=s.next();
                        a.add(Arr);
                    }
                    System.out.println(a);
                    break;
                case 2:
                    System.out.println("Enter the position to be removed:");
                    int p=s.nextInt();
```



```

a.remove(p);
System.out.println(a);
break;
case 3:
System.out.println("Enter the position to be replaced:");
int r=s.nextInt();
System.out.println("Enter the String to be replaced:");
String str=s.next();
a.set(r,str);
System.out.println(a);
break;
case 4:
System.out.println(a);
break;
case 5:
System.out.println("Enter the position to be displayed:");
int d=s.nextInt();
System.out.println(a.get(d));
break;
case 6:
System.exit(0);
break;
}
}
}
}

```

OUTPUT:

```

1. Insertion
2. Remove
3. Replace
4. Display
5. Get element by position
6. Exit

Enter your choice
1
Enter the size:
2
Enter the elements:
2 3
[2, 3]
Enter your choice
2
Enter the position to be removed:
1
[2]

```

PROGRAM 20

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

ALGORITHM:

Step 1:Start

Step 2:Event will happen whwn mouse is clicked.

Step 3: Event will happen when mouse is entered.

Step 4:Event will happen when mouse is exited.

Step 5:Event wiil happen when mouse is clicked.

Step 6:Event will happen when mouse is released.

Step 7:Stop

SOURCE CODE:

```
import java.awt.*;
import java.awt.event.*;
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");
}
public static void main(String[] args)
{
    new MouseListenerExample();
}
}
```

OUTPUT:

PROGRAM 21

AIM: Program to find maximum of three numbers using AWT.

ALGORITHM:

Step 1: Start

Step 2: Create a frame by importing awt packages.

Step 3: Insert three textfields to enter three numbers.

Step 4: Insert a button called Compare which performs the comparison of three numbers entered.

Step 5: In the action performed of Compare button extract contents from the text field and convert to integers, perform comparison and display results.

Step 6: Insert a button called Exit which is for quitting the frame.

Step 7: Stop

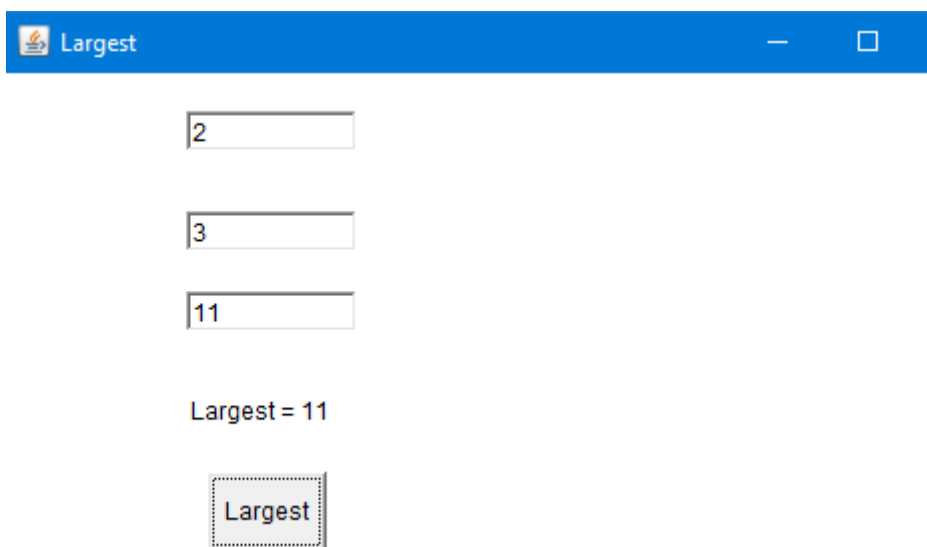
SOURCE CODE:

```
import java.awt.*;
import java.awt.event.*;
public class EventHandle extends Frame
{
    Frame f;
    TextField tf1;
    TextField tf2;
    TextField tf3;
    Label l1;
    Button btn;
    EventHandle ()
    {
        this.setTitle("Largest");
        this.setLayout (null);
        tf1 = new TextField();
        tf1.setBounds(100, 50, 85, 20);
        tf2 = new TextField();
        tf2.setBounds(100, 100, 85, 20);
        tf3 = new TextField();
        tf3.setBounds(100, 140, 85, 20);
        btn = new Button("Largest");
        btn.setBounds(110,230,60,40);
        l1 = new Label("");
        l1.setBounds(100, 190, 85, 20);
        this.add(btn);
        this.add(tf1);
        this.add(tf2);
        this.add(tf3);
```

```

this.add(11);
this.setSize(300,300);
this.setVisible(true);
this.addWindowListener (new WindowAdapter ()
{
public void windowClosing (WindowEvent we)
{
System.exit (0);
}
});
btn.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent e)
{
int a = Integer.parseInt(tf1.getText());
int b = Integer.parseInt(tf2.getText());
int c = Integer.parseInt(tf3.getText());
if(a>=b && a>=c)
11.setText("Largest = " + String.valueOf(a));
else if (b>=c)
11.setText("Largest = " + String.valueOf(b));
else
11.setText("Largest = "+ String.valueOf(c));
}
});
}
public static void main (String args[])
{
new EventHandle ();
}
}

```

OUTPUT:

PROGRAM 22

AIM: Implement a simple calculator using AWT components.

ALGORITHM:

Step 1: Start

Step 2: Create a frame by importing awt packages.

Step 3: Insert two textfields to enter two numbers

Step 4: Insert 4 buttons called +, -, * and / which performs the addition, difference, multiplication and division of two numbers entered respectively.

Step 5: In the action performed of + button extract contents from the text field and convert to integers, perform addition and display results.

Step 6: In the action performed of - button extract contents from the text field and convert to integers, perform subtraction and display results

Step 7: In the action performed of * button extract contents from the text field and convert to integers, perform multiplication and display results

Step 8: In the action performed of / button extract contents from the text field and convert to integers, perform division and display results

Step 9: Insert a button called Exit which is for quitting the frame.

Step 10: Stop.

SOURCE 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");
        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);
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);
}
public void windowClosing(WindowEvent e)
{
f.dispose();
}
public void actionPerformed(ActionEvent e)
{
String z,zt;
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)
    {
        zt=l1.getText();
        z=zt+".";
        l1.setText(z);
    }
    if(e.getSource()==bneg)
    {
        zt=l1.getText();
        z="-"+zt;
    }

```

```

l1.setText(z);
}
if(e.getSource()==bback)
{
zt=l1.getText();
try
{
z=zt.substring(0, zt.length()-1);
}
catch(StringIndexOutOfBoundsException f)
{
return;
}
l1.setText(z);
}
if(e.getSource()==badd)
{
try
{
num1=Double.parseDouble(l1.getText());
}
catch(NumberFormatException f)
{
l1.setText("Invalid Format");
return;
}
z="";
l1.setText(z);
check=1;
}
if(e.getSource()==bsub)
{
try
{
num1=Double.parseDouble(l1.getText());
}
catch(NumberFormatException f)
{
l1.setText("Invalid Format");
return;
}
z="";
l1.setText(z);
check=2;
}
if(e.getSource()==bmult)
{
try
{
num1=Double.parseDouble(l1.getText());
}
catch(NumberFormatException f)
{

```

```

l1.setText("Invalid Format");
return;
}
z="";
l1.setText(z);
check=3;
}
if(e.getSource()==bdiv)
{
try
{
num1=Double.parseDouble(l1.getText());
}
catch(NumberFormatException f)
{
l1.setText("Invalid Format");
return;
}
z="";
l1.setText(z);
check=4;
}
if(e.getSource()==bmod)
{
try
{
num1=Double.parseDouble(l1.getText());
}
catch(NumberFormatException f)
{
l1.setText("Invalid Format");
return;
}
z="";
l1.setText(z);
check=5;
}
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));
}
if(e.getSource()==bclr)
{
num1=0;
num2=0;
check=0;
xd=0;
z="";
l1.setText(z);
}
}
public static void main(String args[])
{
new MyCalc();
}
}

```

OUTPUT:

PROGRAM 23**AIM:** Client server communication using Socket – TCP/IP**ALGORITHM:**

Step 1:Start

Step 2:Create the client application

Step 3:Create the server application

Step 4:Stop

SOURCE CODE:**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 = "";
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);
}
}

```

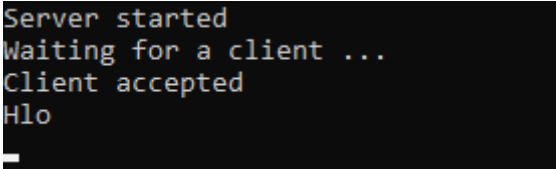
SERVER

```


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);
}
}
```

OUTPUT:

```
Server started
Waiting for a client ...
Client accepted
Hlo
_
```



```
Connected
Hlo
_
```