

Experiment No: 1**Date:****Aim: 1) Basic Operations in Java Programming**

1.1) Write a Java program to find the discriminant value D and find out the roots of the quadratic equation of the form $ax^2+bx+c=0$.

Source Code:

```
import java.util.*;
public class Roots
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        // value a, b, and c
        double a , b, c;
        double root1, root2;

        System.out.println("Enter a, b and c values");
        a=sc.nextDouble();
        b=sc.nextDouble();
        c=sc.nextDouble();

        // calculate the determinant (b2 - 4ac)
        double determinant = b * b - 4 * a * c;

        // check if determinant is greater than 0
        if (determinant > 0) {

            // two real and distinct roots
            root1 = (-b + Math.sqrt(determinant)) / (2 * a);
            root2 = (-b - Math.sqrt(determinant)) / (2 * a);

            System.out.format("root1 = %.2f and root2 = %.2f", root1, root2);
        }

        // check if determinant is equal to 0
        else if (determinant == 0) {

            // roots are equal
            root1 = root2 = -b / (2 * a);
            System.out.format("root1 = root2 = %.2f;", root1);
        }

        // if determinant is less than zero
        else {
```

```
// roots are complex number and distinct
double real = -b / (2 * a);
double imaginary = Math.sqrt(-determinant) / (2 * a);
System.out.format("root1 = %.2f+%.2fi", real, imaginary);
System.out.format("\nroot2 = %.2f-%.2fi", real, imaginary);
}
}
}
```

Output:

D:\>javac Roots.java

D:\>java Roots

Enter a, b and c values

2

3

4

root1 = -0.75+1.20i

root2 = -0.75-1.20i



- 1.2) **Five Bikers Compete in a race such that they drive at a constant speed which may or may not be the same as the other. To qualify the race, the speed of a racer must be more than the average speed of all 5 racers. Take as input the speed of each racer and print back the speed of qualifying racers.**

Source Code:

```
import java.io.*;
class Bike_Racers
{
    public static void main(String args[])throws Exception
    {
        BufferedReader br=new BufferedReader(new
            InputStreamReader(System.in));
        int racer1_Speed,racer2_Speed,racer3_Speed,racer4_Speed,racer5_Speed;
        int sum;
        float avg_Speed;

        System.out.println("Enter 5 Bike Racers Speeds");
        racer1_Speed=Integer.parseInt(br.readLine());
        racer2_Speed=Integer.parseInt(br.readLine());
        racer3_Speed=Integer.parseInt(br.readLine());
        racer4_Speed=Integer.parseInt(br.readLine());
        racer5_Speed=Integer.parseInt(br.readLine());

        sum=racer1_Speed+racer2_Speed+racer3_Speed+racer4_Speed+racer5_Speed;
        avg_Speed=(float)sum/5;

        System.out.println("Average Speed is:"+avg_Speed);

        System.out.println("The Qualified Racers are:");
        if(racer1_Speed>avg_Speed)
            System.out.println("Racer1");
        if(racer2_Speed>avg_Speed)
            System.out.println("Racer2");
        if(racer3_Speed>avg_Speed)
            System.out.println("Racer3");
        if(racer4_Speed>avg_Speed)
            System.out.println("Racer4");
        if(racer5_Speed>avg_Speed)
            System.out.println("Racer5");
    }
}
```

Output:

```
D:\Java_Programs>javac Bike_Racers.java
```

```
D:\Java_Programs>java Bike_Racers
```

```
Enter 5 Bike Racers Speeds
```

```
124
```

```
132
```

```
115
```

```
117
```

```
123
```

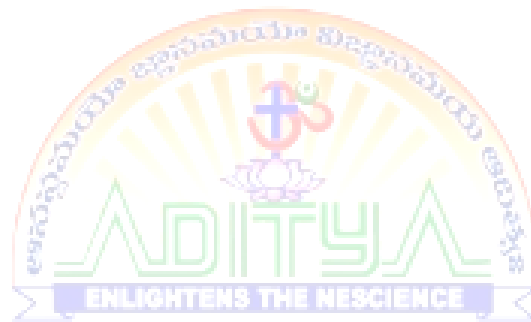
```
Average Speed is:122.2
```

```
The Qualified Racers are:
```

```
Racer1
```

```
Racer2
```

```
Racer5
```



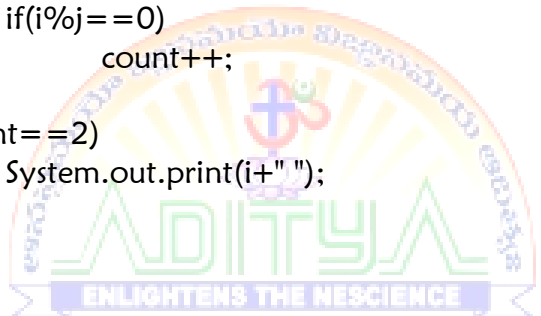
Experiment No: 2**Date:****Aim: 2) Working with Raptor tool****2.1) Write a Java program to select all the prime numbers within the range of 1to100.****Source Code:**

```

import java.io.*;
class Prime_Numbers
{
    public static void main(String args[])throws Exception
    {
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        int num,i,j,count=0;
        System.out.println("Enter a number to find prime numbers upto it");
        num=Integer.parseInt(br.readLine());

        for(i=2;i<=num;i++)
        {
            count=0;
            for(j=1;j<=i;j++)
            {
                if(i%j==0)
                    count++;
            }
            if(count==2)
                System.out.print(i+" ");
        }
    }
}

```

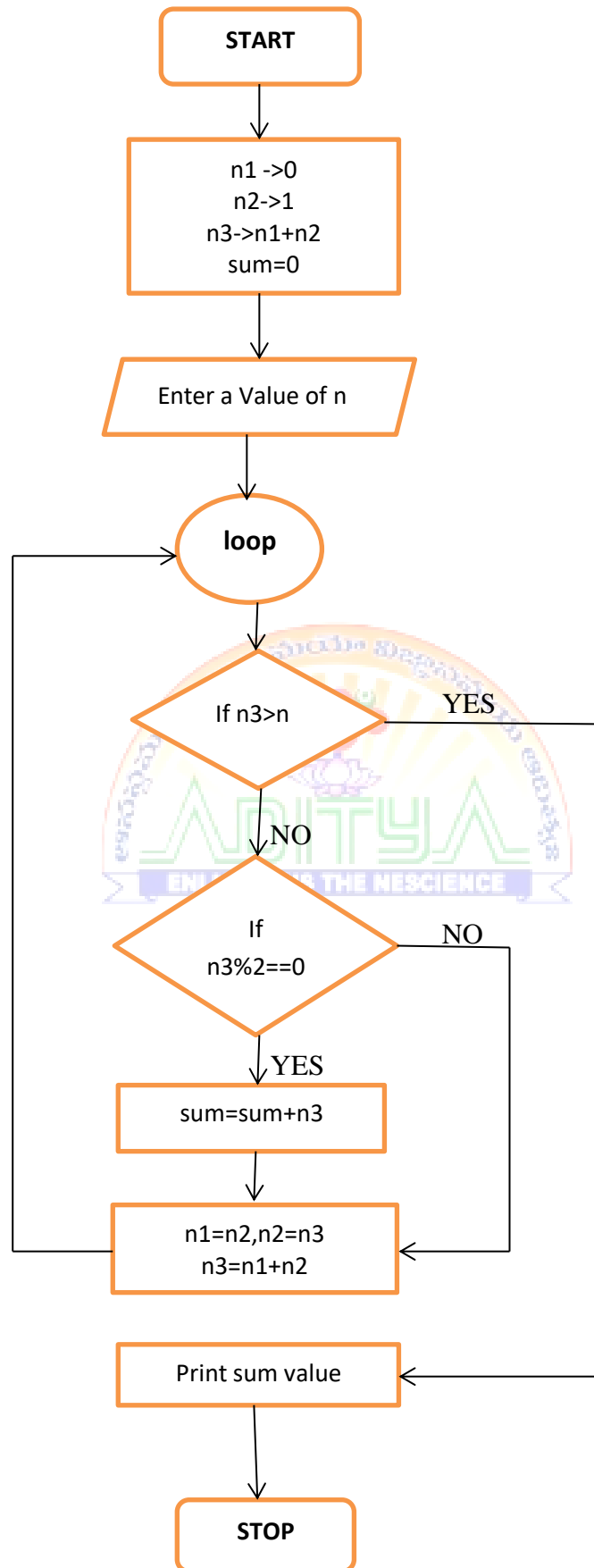

Output:

D:\Java_Programs>java Prime_Numbers

Enter a number to find prime numbers upto it
100

2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97

2.2) Write a Java program to Find the sum of all even terms in the Fibonacci sequence up to the given range N.



Source Code:

```

import java.io.*;
class Fibbonaci
{
    public static void main(String args[])throws Exception
    {
        int n1=0,n2=1,n3,n,sum=0;
        n3=n1+n2;
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        System.out.println("Enter the n value");
        n=Integer.parseInt(br.readLine());

        while(n3<=n)
        {
            if(n3%2==0)
                sum=sum+n3;
            n1=n2;
            n2=n3;
            n3=n1+n2;
        }
        System.out.println("sum of all even terms in the Fibonacci sequence up to the given
        range "+n+" is: "+sum);
    }
}

```

Output:

D:\Java_Programs>java Fibbonaci

Enter the n value

15

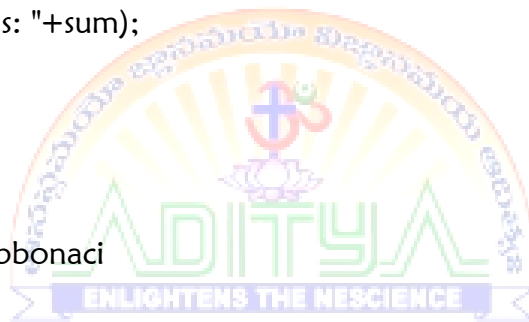
sum of all even terms in the Fibonacci sequence up to the given range 15 is: 10

D:\Java_Programs>java Fibbonaci

Enter the n value

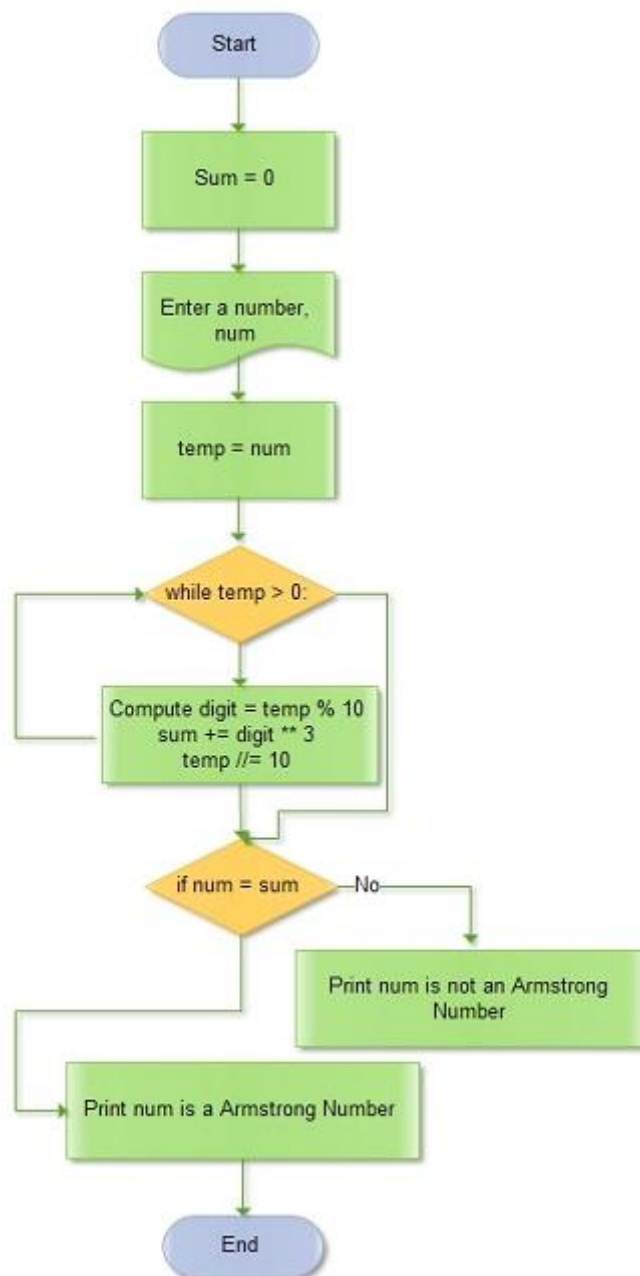
150

sum of all even terms in the Fibonacci sequence up to the given range 150 is: 188



2.3) Write a Java program to check whether a given number is Armstrong or not.

Definition: An Armstrong number or Narcissistic number is an n-digit number equivalent to the sum of digits raised to the nth power of digits from the number. A few Armstrong numbers are: 0, 1, 2, 3, 153, 370, 407, 1634, 8208, etc.



Flow chart to find the given 3 digit number is Armstrong or not

Source Code:

```

import java.io.*;
class Armstrong
{
    public static void main(String args[])throws Exception
    {
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        int num,sum=0,rem,m,d;
        System.out.println("Enter a number");
        num=Integer.parseInt(br.readLine());

        d=(int)Math.log10(num)+1;
        m=num;
        while(num>0)
        {
            rem=num%10;
            sum=sum+(int)Math.pow(rem,d);
            num=num/10;
        }

        if(sum==m)
            System.out.println(m+" is Armstrong Number");
        else
            System.out.println(m+" is not an Armstrong Number");
    }
}

```

**Output:**

```

D:\Java_Programs>java Armstrong
Enter a number
1634
1634 is Armstrong Number

```

```

D:\Java_Programs>java Armstrong
Enter a number
125
125 is not an Armstrong Number

```

Experiment No: 3**Date:****Aim: 3) Class Mechanism**

3.1) Write a Java program to display the details of a person. Personal details should be given in one method and the qualification details in another method.

Source Code:

```
import java.util.*;
class Employee
{
    int empid;
    String empname,desg, Organization,ug,pg,address;
    float Sal,per_marks_ug,per_marks_pg;

    Scanner sc=new Scanner(System.in);

    public void get_PersonalInfo()
    {
        System.out.println("Enter your empid, name, desg, salary, organization name,
            address");
        empid=sc.nextInt();
        sc.nextLine();
        empname=sc.nextLine();
        desg=sc.nextLine();
        Sal=sc.nextFloat();
        sc.nextLine();
        Organization=sc.nextLine();
        address=sc.nextLine();
    }

    public void get_QualificationInfo()
    {
        System.out.println("Enter your UG course, marks and PG course and
            Marks");
        ug=sc.nextLine();
        per_marks_ug=sc.nextFloat();
        sc.nextLine();
        pg=sc.nextLine();
        per_marks_pg=sc.nextFloat();
    }

    public void show_PersonalInfo()
    {
        System.out.println("=====");
        System.out.println("                PERSONAL INFORMATION                ");
        System.out.println("=====");
        System.out.println("EMPID: "+empid);
        System.out.println("EMP NAME: "+empname);
        System.out.println("DESGINATION: "+desg);
        System.out.println("SALARY: "+Sal);
        System.out.println("ORGANIZATION NAME: "+Organization);
    }
}
```

```

        System.out.println("ADDRESS: "+address);
    }


    public void show_QualificationInfo()
    {
        System.out.println("=====");
        System.out.println("                QUALIFICATION INFORMATION                ");
        System.out.println("=====");

        System.out.println("UG COURSE: "+ug);
        System.out.println("UG PERCENTAGE: "+per_marks_ug);
        System.out.println("PG COURSE: "+pg);
        System.out.println("PG PERCENTAGE: "+per_marks_pg);

    }

    public static void main(String args[])
    {
        // Object Creation - memory for member variable declared inside the class
        Employee e1=new Employee();
        // reading of employee information
        e1.get_PersonalInfo();
        e1.get_QualificationInfo();
        // showing of employee information
        e1.show_PersonalInfo();
        e1.show_QualificationInfo();
    }
}

```



Output:

```
D:\Java_Programs>java Employee
Enter your empid, name, desg, salary, organization name, address
1111
Ramesh S
Assistant Professor
35000
Aditya Engineering College
Kakinada
Enter your UG course, marks and PG course and Marks
B.Tech CSE
75
M.Tech CSE
82
```

```
=====
PERSONAL INFORMATION
=====
```

```
EMPID: 1111
EMP NAME: Ramesh S
DESIGNATION: Assistant Professor
SALARY: 35000.0
ORGANIZATION NAME: Aditya Engineering College
ADDRESS: Kakinada
```

```
=====
QUALIFICATION INFORMATION
=====
```

```
UG COURSE: B.Tech CSE
UG PERCENTAGE: 75.0
PG COURSE: M.Tech CSE
PG PERCENTAGE: 82.0
```

3.2) Write a Java program to implement constructor.**Source Code:**

```

class Test
{
    int x;
    public Test() // default constructor
    {
        System.out.println("Default Constructor");
        x = 1;
    }
    public Test(int x) // parameterized constructor
    {
        System.out.println("Parameterized constructor");
        this.x = x;
    }
    public Test(Test t) // copy constructor
    {
        this.x = t.x;
        System.out.println("Copy Constructor");
    }
    void show()
    {
        System.out.println("X: "+x);
    }
    public static void main (String args [])
    {
        Test t1 = new Test();           // default constructor
        Test t2 = new Test(13);         // parameterized
        Test t3 = new Test(t2);         // copy constructor
        t1.show ();                     // 1
        t2.show ();                     // 13
        t3.show ();                     // 13
    }
}

```

Output:

D:\ECE-A>java Test

X: 1

X: 13

X: 13

3.3) Write a Java program to implement method overloading.**Source Code:**

```

class Method_Overloading
{
    public void methodOne()
    {
        System.out.println("no argument");
    }
    public void methodOne(int x,int y)
    {
        System.out.println(x+y);
    }
    public void methodOne(int d)
    {
        System.out.println(d);
    }
    public void methodOne(double d)
    {
        System.out.println(d);
    }
    public static void main(String args[])
    {
        Method_Overloading mo=new Method_Overloading();

        mo.methodOne();
        mo.methodOne(10);
        mo.methodOne(10,20);
        mo.methodOne(3.14);
    }
}

```

Output:

D:\Java_Programs>javac Method_Overloading.java

D:\Java_Programs>java Method_Overloading

no argument

10

30

3.14

Experiment No: 4**Date:****Aim: 4) Working with Arrays****4.1) Write a Java program to perform addition and multiplication of two matrices.****Source Code:**

```

import java.util.*;
class Matrix
{
    int mat[][] , row, col, i, j, k;
    Scanner sc = new Scanner(System.in);

    public Matrix(int row, int col)
    {
        this.row = row;
        this.col = col;
        mat = new int[row][col];
    }

    public void read_Matrix()
    {
        System.out.println("Enter " + (row * col) + " Elements");
        for(i=0; i<row; i++)
            for(j=0; j<col; j++)
                mat[i][j] = sc.nextInt();
    }

    public void show_Matrix()
    {
        for(i=0; i<row; i++)
        {
            for(j=0; j<col; j++)
            {
                System.out.print(mat[i][j] + " ");
            }
            System.out.println();
        }
    }

    public Matrix Addition(Matrix m)
    {
        Matrix m3 = new Matrix(m.row, m.col);
        for(int i=0; i<m3.row; i++)
            for(int j=0; j<m3.col; j++)
                m3.mat[i][j] = this.mat[i][j] + m.mat[i][j];

        return m3;
    }
}

```

```

public Matrix Multiplication(Matrix m)
{
    if(this.col!=m.row)
    {
        System.out.println("Multiplication is not possible");
        return null;
    }
    else
    {
        Matrix m3=new Matrix(this.row,m.col);
        for(i=0;i<row;i++)
        {
            for(j=0;j<row;j++)
            {
                m3.mat[i][j]=0;
                for(k=0;k<m.col;k++)
                {
                    m3.mat[i][j]=m3.mat[i][j]+this.mat[i][k]*m.mat[k][j];
                }
            }
        }
        return m3;
    }
}

public static void main(String args[])
{
    Matrix m1=new Matrix(2,2);
    Matrix m2=new Matrix(2,2);
    Matrix m3;

    System.out.println("Matrix-1");
    m1.read_Matrix();
    System.out.println("Matrix-2");
    m2.read_Matrix();
    System.out.println("Given Matrix are");
    System.out.println("Matrix-1 is: ");
    m1.show_Matrix();
    System.out.println("\nMatrix-2 is: ");
    m2.show_Matrix();

    System.out.println("\nMatrix addition = ");
    m3=m1.Addition(m2);
    m3.show_Matrix();

    System.out.println("\nMatrix Multiplication =");
    m3=m1.Multiplication(m2);
    m3.show_Matrix();
}
}

```


Output:

D:\>javac Matrix.java

D:\>java Matrix

Matrix-1

Enter 4 Elements

2

2

2

2

Matrix-2

Enter 4 Elements

1

1

1

1

Given Matrix are

Matrix-1 is:

2 2

2 2

Matrix-2 is:

1 1

1 1

Matrix addition =

3 3

3 3

Matrix Multiplication =

4 4

4 4



4.2) Write a Java program to implement binary search.**Source Code:**

```

import java.util.*;
class BinarySearchExample
{
    public static void binarySearch(int arr[], int first, int last, int key)
    {
        int mid = (first + last)/2;
        while( first <= last )
        {
            if ( arr[mid] < key )
            {
                first = mid + 1;
            }
            else if(arr[mid] == key )
            {
                System.out.println("Element is found at index: " + mid);
                break;
            }
            else
            {
                last = mid - 1;
            }
            mid = (first + last)/2;
        }
        if ( first > last )
        {
            System.out.println("Element is not found!");
        }
    }
    public static void main(String args[])
    {
        Scanner sc=new Scanner(System.in);
        int n,key,arr[];
        System.out.println("Enter the number of elements");
        n=sc.nextInt();
        arr=new int[n];
        System.out.println("Enter "+n+" elements");
        for(int i=0;i<n;i++)
            arr[i]=sc.nextInt();
        System.out.println("Enter the number to search");
        key=sc.nextInt();
        int last=n-1;
        binarySearch(arr,0,last,key);
    }
}

```

Output:

```
D:\>java BinarySearchExample
Enter the number of elements
8
Enter 8 elements
11 22 33 44 55 66 77 88
Enter the number to search
44
Element is found at index: 3
```



Experiment No: 5**Date:****Aim: 5) Working with Strings****5.1) Write a Java program to sort given set of strings.****Source Code:**

```

import java.util.*;
public class String_Sort
{
    public static void main(String[] args)
    {
        int count;
        String temp;
        Scanner scan = new Scanner(System.in);

        //User will be asked to enter the count of strings
        System.out.println("Enter number of strings you would like to enter:");
        count = scan.nextInt();

        String str[] = new String[count];

        //User is entering the strings and they are stored in an array
        System.out.println("Enter the Strings one by one:");
        scan.nextLine();
        for(int i = 0; i < count; i++)
        {
            str[i] = scan.nextLine();
        }

        //Sorting the strings
        for (int i = 0; i < count; i++)
        {
            for (int j = i + 1; j < count; j++) {
                if (str[i].compareTo(str[j])>0)
                {
                    temp = str[i];
                    str[i] = str[j];
                    str[j] = temp;
                }
            }
        }
    }
}

```

```
//Displaying the strings after sorting them based on alphabetical order
System.out.print("Strings in Sorted Order:");
for (int i = 0; i <= count - 1; i++)
{
    System.out.print(str[i] + ", ");
}
}
```

Output:

```
D:\>java String_Sort
Enter number of strings you would like to enter:
5
Enter the Strings one by one:
Rama
Sita
Laxman
Hanuma
Bharata
Strings in Sorted Order:Bharata, Hanuma, Laxman, Rama, Sita
```



5.2) Write a Java program for using String Buffer to remove or delete a character.**Source Code:**

```

class StringBuffer_Demo
{
    public static void main(String args[])
    {
        StringBuffer s1=new StringBuffer();
        System.out.println(s1.capacity());    // 16 => C=(S+1)*2 , 34
        System.out.println(s1.length());      // 0

        StringBuffer s2=new StringBuffer("Welcome ");
        System.out.println(s2.capacity());    // 24

        System.out.println(s2.charAt(4));    // o

        s2.setCharAt(4,'a');
        System.out.println(s2); // Welcame

        s2.deleteCharAt(4);
        System.out.println(s2);           // Welcme

        s2.append(" Srinu");
        System.out.println(s2);           // Welcme Srinu

        s2.insert(4,"a");
        System.out.println(s2);           // Welcame Srinu

        s2.delete(8,13);
        System.out.println(s2);           // Welcame

        s2.append(true);
        System.out.println(s2);           // Welcame true

        s2.reverse();
        System.out.println(s2);           // eurt emacleW
    }
}

```

Output:

```

D:\>java StringBuffer_Demo
16
0
24
o
Welcame
Welcme
Welcme Srinu
Welcame Srinu
Welcame u
Welcame utrue
eurtu emacleW

```

5.3) Write a Java program to find the number of tokens in a given string without using count-Tokens() method but by using other methods of String Tokenizer class.

Source Code:

```
import java.util.StringTokenizer;
public class String_TokenizerDemo
{
    public static void main(String args[])
    {
        StringTokenizer st = new StringTokenizer("my name is khan and your name is
            salman"," ");
        //System.out.println(st.countTokens());
        int count=0;
        while (st.hasMoreTokens())
        {
            System.out.println(st.nextToken()); // my nam
            count++;
        }

        System.out.println("No of Tokens: "+count);
    }
}
```

Output:

```
D:\>java String_TokenizerDemo
my
name
is
khan
and
your
name
is
salman
No of Tokens: 9
```



Experiment No: 6**Date:****Aim: 6) Working with Inheritance, Interface & Abstract Class****6.1) Write a Java program to find the available balance in a customer account details should be taken in another class. (Note: Make use of Multi-Level Inheritance.)****Source Code:**

```

import java.util.*;
class Customer
{
    String cust_id, cust_name, address;
    float balance;
    Scanner sc=new Scanner(System.in);

    public void get_CustomerInfo()
    {
        System.out.println("Enter Customer ID, Name, Balance and address");
        cust_id=sc.nextLine();
        cust_name=sc.nextLine();
        balance=sc.nextFloat();
        sc.nextLine();
        address=sc.nextLine();
    }
    public void show_CustomerInfo()
    {
        System.out.println("Customer Details are:");
        System.out.println("Id: "+cust_id);
        System.out.println("Name: "+cust_name);
        System.out.println("Balance: "+balance);
        System.out.println("Address: "+address);
    }
}
class Transaction extends Customer
{
    public void deposit(float amt)
    {
        System.out.println("Amount Deposited: "+amt);
        balance=balance+amt;
    }
    public void withdraw(float amt)
    {
        System.out.println("Amount withdrawn: "+amt);
        balance=balance-amt;
    }
}

```



```
        public void show_Bal()
        {
            System.out.println("Available Balance: "+balance);
        }
    }
    class Bank extends Transaction
    {
        static String bankname="Canara Bank",ifsc="CBN0003268";

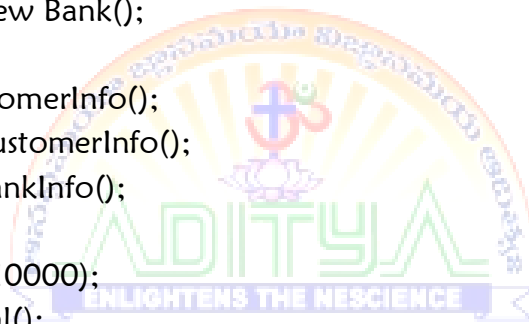
        public void show_BankInfo()
        {
            System.out.println("Bank Name: "+Bank.bankname);
            System.out.println("IFSC Code: "+Bank.ifsc);
        }

        public static void main(String args[])
        {
            Bank c1=new Bank();

            c1.get_CustomerInfo();
            c1.show_CustomerInfo();
            c1.show_BankInfo();

            c1.deposit(10000);
            c1.show_Bal();

            c1.withdraw(5000);
            c1.show_Bal();
        }
    }
}
```



Output:

D:\Inheritance>java Bank

Enter Customer ID, Name, Balance and address

32682210001421

M.Srinu

200000

Kakinada

Customer Details are:

Id: 32682210001421

Name: M.Srinu

Balance: 200000.0

Address: Kakinada

Bank Name: Canara Bank

IFSC Code: CBN0003268

Amount Deposited: 10000.0

Available Balance: 210000.0

Amount withdrawn: 5000.0

Available Balance: 205000.0



6.2) Take the details of internal exam marks in one Interface. Take the details of external exam marks in another interface. Write a Java program to find the total marks obtained in each subject by a student. (Note: Make use of Multiple Inheritance using interfaces.).

Source Code:

```
import java.util.*;

interface Internal
{
    void get_InternalMarks();
}

interface External
{
    void get_ExternalMarks();
}

interface Marks extends Internal, External
{
    void show_Marks();
}

class Result implements Marks
{
    // s1_i -> sub1 internal, s1_e -> sub1 external
    float s1_i,s1_e,s2_i,s2_e,s3_i,s3_e;
    Scanner sc=new Scanner(System.in);
    public void get_InternalMarks()
    {
        System.out.println("Enter 3 subjects internal marks (0 - 40)");
        s1_i=sc.nextFloat();
        s2_i=sc.nextFloat();
        s3_i=sc.nextFloat();
    }
    public void get_ExternalMarks()
    {
        System.out.println("Enter 3 subjects External marks (0 - 60)");
        s1_e=sc.nextFloat();
        s2_e=sc.nextFloat();
        s3_e=sc.nextFloat();
    }
}
```

```
public void show_Marks()
{
    System.out.println("Subject \t Internal\tExternal\tTotal_Marks: ");
    System.out.println(" Sub1 \t\t "+s1_i+"\t\t"+s1_e+"\t\t"+(s1_i+s1_e));
    System.out.println(" Sub2 \t\t "+s2_i+"\t\t"+s2_e+"\t\t"+(s2_i+s2_e));
    System.out.println(" Sub3 \t\t "+s3_i+"\t\t"+s3_e+"\t\t"+(s3_i+s3_e));
}
}

class Mainclass
{
    public static void main(String args[])
    {
        Marks m=new Result();
        m.get_InternalMarks();
        m.get_ExternalMarks();
        m.show_Marks();
    }
}
```



Output:

D:\Inheritance>java Mainclass

Enter 3 subjects internal marks (0 - 40)

37 29 38

Enter 3 subjects External marks (0 - 60)

57 49 58

Subject	Internal	External	Total_Marks:
Sub1	37.0	57.0	94.0
Sub2	29.0	49.0	78.0
Sub3	38.0	58.0	96.0



6.3) Write a Java program to find the areas of different shapes using abstract classes.**Source Code:**

```

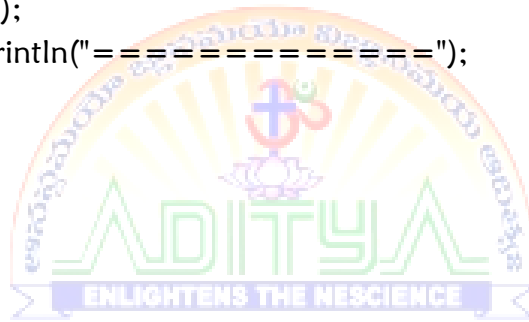
import java.util.*;
abstract class Shape
{
    Scanner sc=new Scanner(System.in);
    float s1,s2,a;
    final float pi=3.14f;
    public abstract void get_Input();
    public abstract void Cal_Area();
    public void show_Area()
    {
        System.out.println("Area is:"+a);
    }
}
class Rect extends Shape
{
    public void get_Input()
    {
        System.out.println("Enter L and B values");
        s1=sc.nextFloat();
        s2=sc.nextFloat();
    }
    public void Cal_Area()
    {
        a=s1*s2;
    }
}
class Circle extends Shape
{
    public void get_Input()
    {
        System.out.println("Enter radius of the Circle");
        s1=sc.nextFloat();
    }
    public void Cal_Area()
    {
        a=pi*s1*s1;
    }
}

```

```
class Mainclass
{
    public static void main(String args[])throws Exception
    {
        Shape s;

        s=new Rect();
        System.out.println("Rectangle:");
        s.get_Input();
        s.Cal_Area();
        s.show_Area();
        System.out.println("=====");

        s=new Circle();
        System.out.println("Circle:");
        s.get_Input();
        s.Cal_Area();
        s.show_Area();
        System.out.println("=====");
    }
}
```



Output:

D:\Inheritance>java Mainclass

Rectangle:

Enter L and B values

14

16

Area is:224.0

=====

Circle:

Enter radius of the Circle

5.6

Area is:98.4704

=====



Experiment No: 7**Date:****Aim: 7) Working with Packages****7.1) Write a Java program that import and use user defined package.****User defined packages**

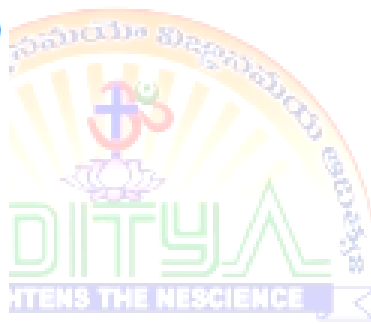
User-defined packages are those which are developed by users in order to group related classes, interfaces and sub packages. With the help of an example program, let's see how to create packages, compile Java programs inside the packages and execute them.

Steps involved in user defined package creation:

1. Creation of user defined package file
2. Compilation of user defined package file
3. Setting of class path
4. Importing of user defined package in another application.

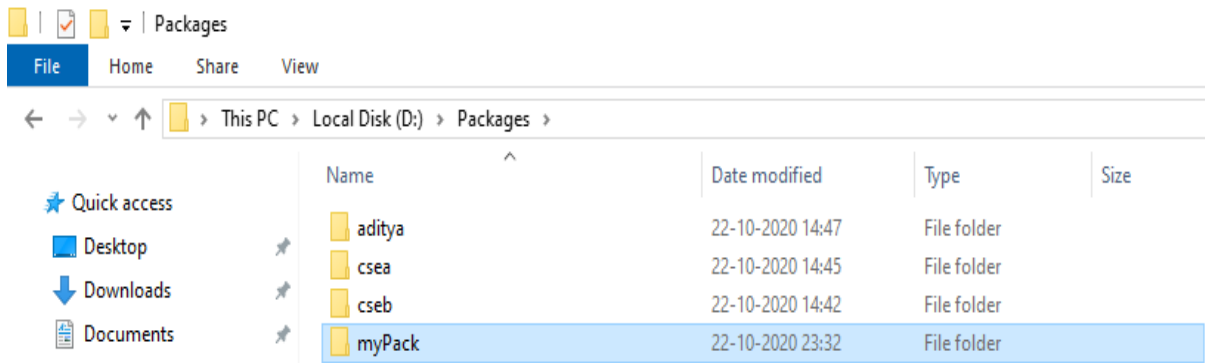
Source Code:**Step – 1:**

```
package myPack;
public class Compare
{
    public int getMax(int n, int m)
    {
        if(n>m)
            return n;
        else
            return m;
    }
    public int getMin(int n,int m)
    {
        if(n<m)
            return n;
        else
            return m;
    }
    public void getEqual(int n,int m)
    {
        if(n==m)
            System.out.println("Equal");
        else
            System.out.println("Not Equal");
    }
}
```

**Step – 2:**

```
D:\Java_Programs>javac -d D:\Packages Compare.java
```

```
D:\Java_Programs>
```

**Step – 3 & 4:**

```

1 import java.util.Scanner;    // built in package
2 import myPack.Compare;      // user defined package
3 class NumberDemo
4 {
5     public static void main(String args[])
6     {
7         int n1,n2;
8         Scanner sc=new Scanner(System.in);
9         Compare c=new Compare();
10        System.out.println("Enter any two numbers");
11        n1=sc.nextInt();
12        n2=sc.nextInt();
13
14        int max=c.getMax(n1,n2);
15        int min=c.getMin(n1,n2);
16        c.getEqual(n1,n2);
17
18        System.out.println("Maximu: "+max);
19        System.out.println("Minimum: "+min);
20    }
21 }

```

Output:

```
D:\Java_Programs>set classpath=.;D:\Packages;%classpath%;
```

```
D:\Java_Programs>javac NumberDemo.java
```

```
D:\Java_Programs>java NumberDemo
```

```
Enter two numbers
```

```
25
```

```
36
```

```
Not Equal
```

```
Maximum value is:36
```

```
Minimum value is:25
```

7.2) Write a Java program to illustrate the use of protected members in a package.

Protected: The protected access modifier is accessible within package and outside the package but through inheritance only.

The protected access modifier can be applied on the data member, method and constructor. It can't be applied on the class.

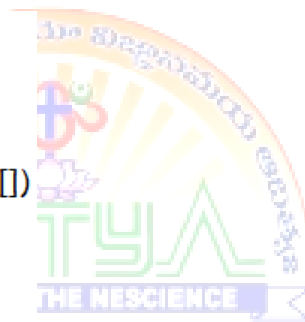
Source Code:**CASE – 1:**

```

1 package pack1;
2 public class A
3 {
4     protected void m1()
5     {
6         System.out.println("protected method in A");
7     }
8 }

1 package pack2;
2 import pack1.A;
3 public class B extends A
4 {
5     public static void main(String args[])
6     {
7         B b=new B();
8         b.m1();
9     }
10 }

```

**Output:**

```
D:\ECE-A\package>javac -d D:\ECE A.java
```

```
D:\ECE-A\package>javac -d D:\ECE B.java
```

```
D:\ECE-A\package>java pack2.B
protected method in A
```

CASE – 2:

```

1 package pack1;
2 public class A
3 {
4     protected void m1()
5     {
6         System.out.println("protected method in A");
7     }
8 }

1 package pack2;
2 import pack1.A;
3 public class B
4 {
5     public static void main(String args[])
6     {
7         A a=new A();
8         a.m1();
9     }
10 }

```

Output:

D:\ECE-A\package>javac -d D:\ECE A.java

D:\ECE-A\package>javac -d D:\ECE B.java

B.java:8: error: m1() has protected access in A
 a.m1();
 ^

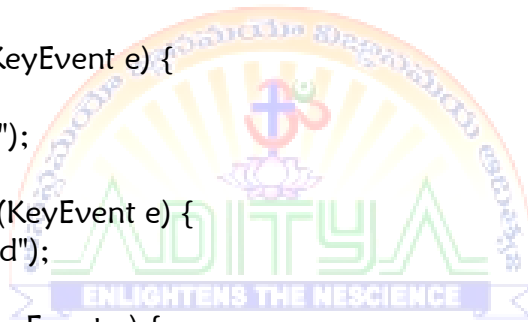
1 error

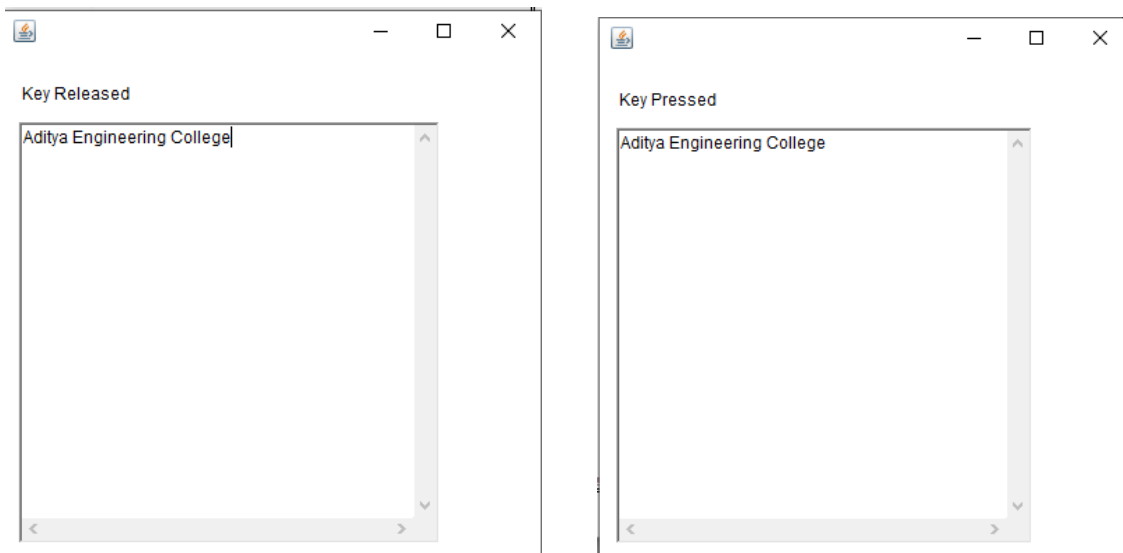
Experiment No: 8**Date:****Aim: 8) Working with Event Handling****8.1) Write a Java program to illustrate the Keyboard Events.****Source Code:**

```

import java.awt.*;
import java.awt.event.*;
class KeyListenerExample extends Frame implements KeyListener
{
    Label l;
    TextArea area;
    public KeyListenerExample()
    {
        l=new Label();
        l.setBounds(20,50,100,20);
        area=new TextArea();
        area.setBounds(20,80,300, 300);
        area.addKeyListener(this);
        add(l);
        add(area);
    }
    public void keyPressed(KeyEvent e) {
        l.setText("Key Pressed");
    }
    public void keyReleased(KeyEvent e) {
        l.setText("Key Released");
    }
    public void keyTyped(KeyEvent e) {
        l.setText("Key Typed");
    }
    public static void main(String[] args)
    {
        KeyListenerExample k1=new KeyListenerExample();
        k1.setSize(400,400);
        k1.setLayout(null);
        k1.setVisible(true);
    }
}

```



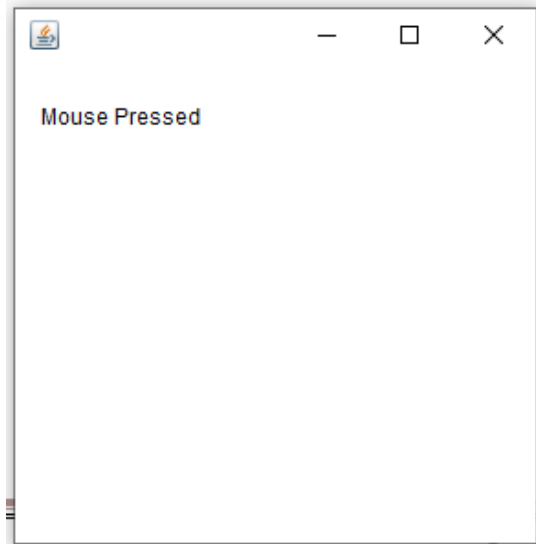
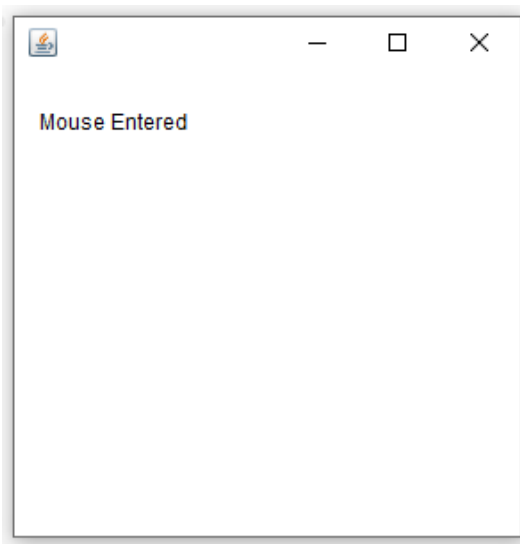
Output:

8.2) Write a Java program to illustrate the Mouse Events.**Source Code:**

```

import java.awt.*;
import java.awt.event.*;
class MouseListenerExample extends Frame implements MouseListener
{
    Label l;
    MouseListenerExample()
    {
        addMouseListener(this);
        l=new Label();
        l.setBounds(20,50,100,20);
        add(l);
    }
    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) {
        MouseListenerExample m1=new MouseListenerExample();
        m1.setSize(300,300);
        m1.setLayout(null);
        m1.setVisible(true);
    }
}

```

Output:

Experiment No: 9**Date:****Aim: 9) Working with AWT****9.1) Write a Java program to generate a simple calculator using AWT components.****Source Code:**

```

import java.awt.*;
import java.awt.event.*;
class Cal extends Frame implements ActionListener
{
    String msg=" ";
    int v1,v2,result;
    TextField t1;
    Button b[]=new Button[10];
    Button add,sub,mul,div,clear,mod,EQ;
    char OP;
    public Cal()
    {
        setLayout(new FlowLayout());
        Panel p1=new Panel();
        t1=new TextField(30);
        Panel p2=new Panel();
        p2.setLayout(new GridLayout(4,4));
        for(int i=0;i<10;i++)
        {
            b[i]=new Button(""+i);
        }
        add=new Button("+");
        sub=new Button("-");
        mul=new Button("*");
        div=new Button("/");
        clear=new Button("clear");
        EQ=new Button("=");
        t1.addActionListener(this);
        p1.add(t1);

        for(int i=0;i<10;i++)
        {
            p2.add(b[i]);
        }
        p2.add(add);
        p2.add(sub);
        p2.add(mul);
        p2.add(div);
    }
}

```

```

        p2.add(clear);
        p2.add(EQ);
        add(p1);
        add(p2);
        for(int i=0;i<10;i++)
        {
            b[i].addActionListener(this);
        }
        add.addActionListener(this);
        sub.addActionListener(this);
        mul.addActionListener(this);
        div.addActionListener(this);
        clear.addActionListener(this);
        EQ.addActionListener(this);
    }
    public void actionPerformed(ActionEvent ae)
    {
        String str=ae.getActionCommand();
        char ch=str.charAt(0);
        if ( Character.isDigit(ch))
            t1.setText(t1.getText()+str);
        else if(str.equals("+"))
        {
            v1=Integer.parseInt(t1.getText());
            OP='+';
            t1.setText("");
        }
        else if(str.equals("-"))
        {
            v1=Integer.parseInt(t1.getText());
            OP='-';
            t1.setText("");
        }
        else if(str.equals("*"))
        {
            v1=Integer.parseInt(t1.getText());
            OP='*';
            t1.setText("");
        }
        else if(str.equals("/"))
        {
            v1=Integer.parseInt(t1.getText());
            OP='/';
            t1.setText("");
        }
    }

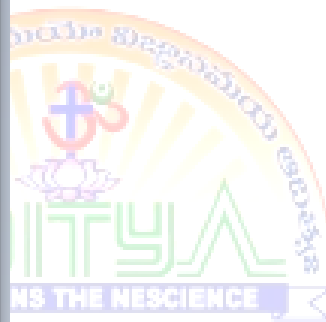
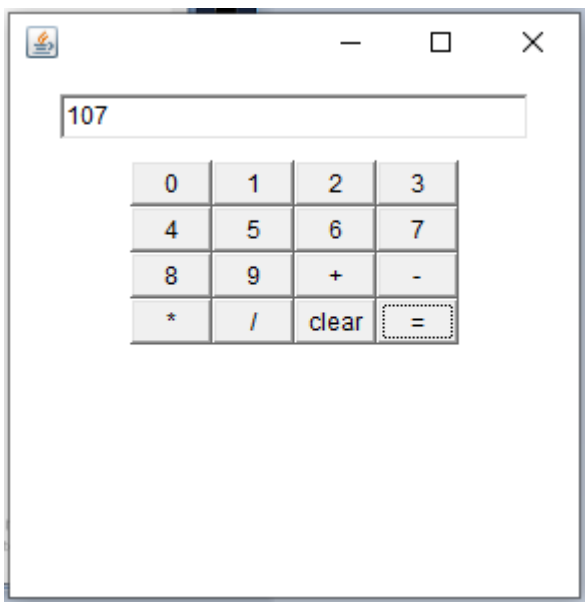
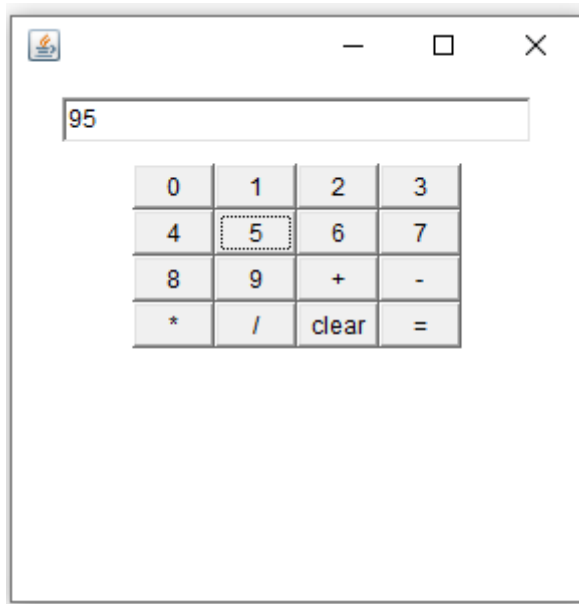
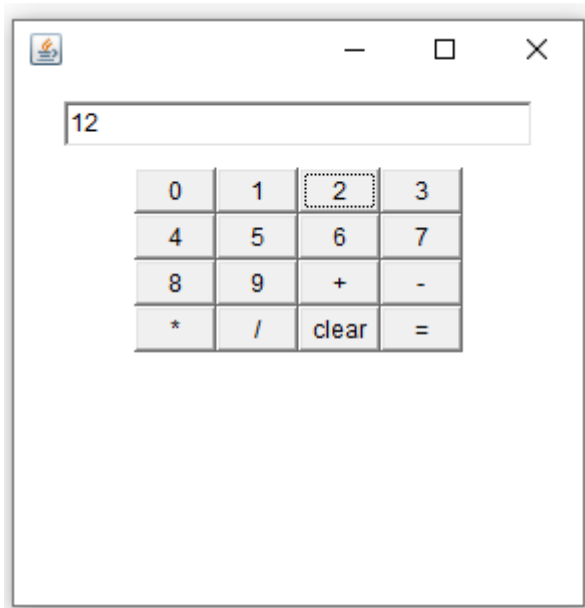
```

```
}

if(str.equals("="))
{
    v2=Integer.parseInt(t1.getText());
    if(OP=='+')
        result=v1+v2;
    else if(OP=='-')
        result=v1-v2;
    else if(OP=='*')
        result=v1*v2;
    else if(OP=='/')
        result=v1/v2;
    t1.setText(""+result);
}
if(str.equals("clear"))
{
    t1.setText("");
}
}

public static void main(String args[])
{
    Cal c=new Cal();
    c.setSize(300,300);
    c.setVisible(true);
}
}
```



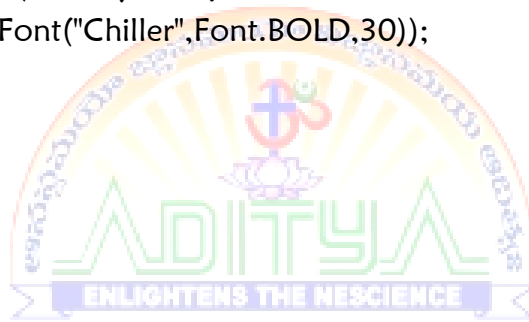
Output:

Experiment No: 10**Date:****Aim: 10) Working with Swings****10.1) Write a Java program to create a single ball bouncing inside a JPanel.****Source Code:**

```

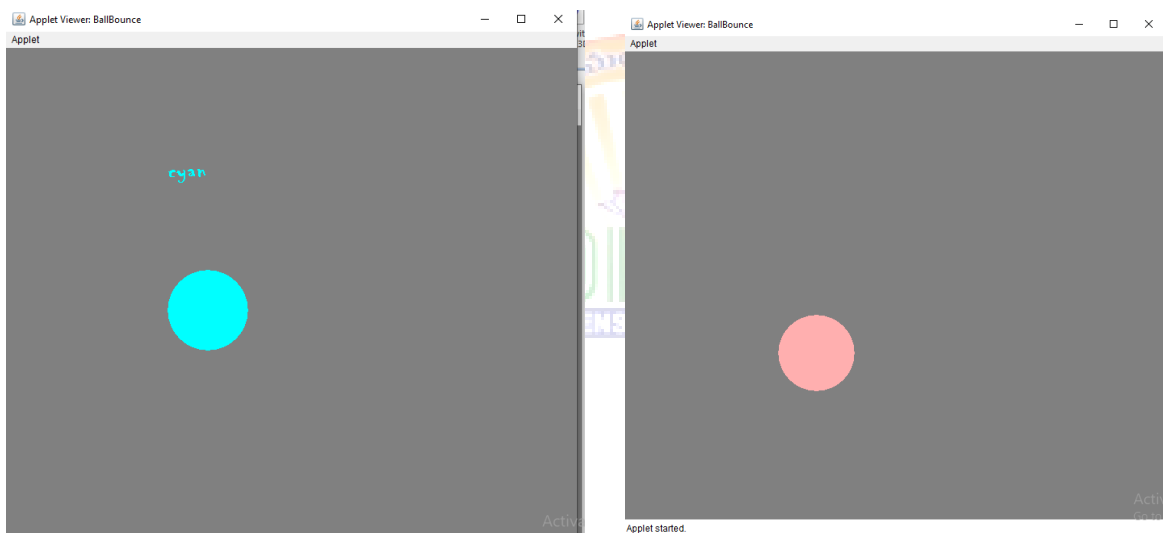
import java.awt.*;
import java.applet.*;
/* <applet code=BallBounce width=600 height=700>
</applet> */
class BallBounce extends Applet implements Runnable
{
    int stpx=200, stpy=200, x=100, y=100, i=0;
    Thread t;
    public void init()
    {
        t=new Thread(this);
        setBackground(Color.gray);
        setForeground(Color.yellow);
        setFont(new Font("Chiller",Font.BOLD,30));
        t.start();
    }
    public void run()
    {
        try
        {
            for(;;)
            {
                if(stpy==200)
                    i=0;
                repaint();
                Thread.sleep(20);
                if(stpy==500)
                    i=1;
            }
        }
        catch(Exception e){ }
    }
    public void paint(Graphics g)
    {
        if(i==0)
        {
            g.setColor(Color.cyan);
            g.drawString("cyan",200,160);

```



```
        g.fillOval(stpx, stpy, x, y);  
        stpy += 5;  
    }  
    if(i == 1)  
    {  
        g.setColor(Color.pink);  
        g.drawString("pink", 225, 650);  
        g.fillOval(stpx, stpy, x, y);  
        stpy -= 5;  
    }  
}  
}
```

Output:

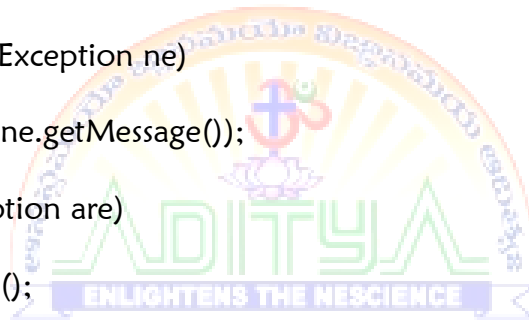


Experiment No: 11**Date:****Aim: 11) Working with Exception Handling****11.1) Write a Java program to illustrate exception handling mechanism using multiple catch clauses.****Source Code:**

```

class ExceptionDemo
{
    public static void main(String[] args)
    {
        int m, n, o=0;
        try
        {
            m = Integer.parseInt(args[0]);
            n = Integer.parseInt(args[1]);
            o = m/n;
        }
        catch(ArrayIndexOutOfBoundsException ae)
        {
            System.out.println(ae.getMessage());
        }
        catch(NumberFormatException ne)
        {
            System.out.println(ne.getMessage());
        }
        catch(ArithmeticException are)
        {
            are.printStackTrace();
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
        finally
        {
            System.out.println("Cleanup code");
            System.out.println(o);
        }
    }
}

```



Output:

D:\>java ExceptionDemo 10 2

Cleanup code

5

D:\>java ExceptionDemo 10

Index 1 out of bounds for length 1

Cleanup code

0

D:\>java Exception1 10 0

java.lang.ArithmeticException: / by zero at Exception1.main(Exception1.java:9)

Cleanup code

0

D:\>java Exception1 10 a

For input string: "a"

Cleanup code

0



11.2) Write a Java program to make use of Built-in and user-defined Exceptions in handling a run.

Source Code:

```
class TooYoungException extends RuntimeException
```

```
{
    public TooYoungException(String s)
    {
        super(s);
    }
}
```

```
class TooOldException extends RuntimeException
```

```
{
    public TooOldException(String s)
    {
        super(s);
    }
}
```

```
class Exception2
```

```
{
    public static void main(String[] args)
    {
        int age = Integer.parseInt(args[0]);
        try
        {
            if(age < 18)
            {
                throw new TooYoungException("You have to wait until you get 18");
            }
            else if(age > 60)
            {
                throw new TooOldException("You are too old");
            }
            else
            {
                System.out.println("You are eligible");
            }
        }
        catch(TooOldException oe)
        {
            System.out.println(oe);
        }
        catch(TooYoungException ye)
        {
            System.out.println(ye);
        }
    }
}
```

Output:

```
D:\> javac Exception2.java
```

```
D:\> java Exception2 19
```

You are eligible

```
D:\> java Exception2 60
```

TooOldException: You are too old

```
D:\> java Exception2 17
```

TooYoungException: You have to wait until you get 18



Experiment No: 12**Date:****Aim: 12) Working with Multithreading**

Write a Java program that creates threads by extending Thread class. First thread display “Good Morning” every 1 sec, the second thread displays “Hello” every 2 seconds and the third display “Welcome” every 3 seconds, (Repeat the same by implementing Runnable).

Source Code:

```
class MyThread1 extends Thread
```

```
{
    public void run()
    { try{
        while(true)
        {
            System.out.println(Thread.currentThread().getName()+" : Good Morning");
            Thread.sleep(1000);
        }
    }
    catch (InterruptedException ie)
    {
    }
}
```

```
class MyThread2 extends Thread
```

```
{
    public void run()
    {
        try{
            while(true)
            {
                System.out.println(Thread.currentThread().getName()+" : Hello");
                Thread.sleep(2000);
            }
        }
        catch (InterruptedException ie)
        {
        }
    }
}
```

```
class MyThread3 extends Thread
```

```
{
    public void run()
    {
        try{
            while(true)
```



```

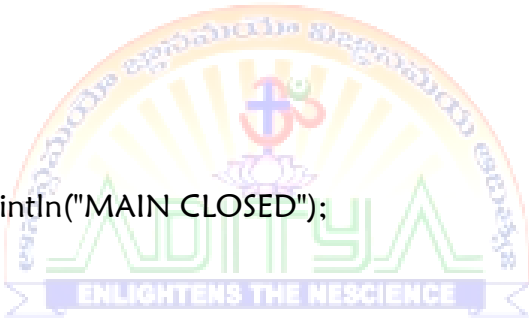
        {
            System.out.println(Thread.currentThread().getName()+" : Welcome");
            Thread.sleep(3000);
        }

    }
    catch (InterruptedException ie)
    {
    }
}
}
class MainDemo
{
    public static void main(String args[])
    {
        MyThread1 t1=new MyThread1();
        MyThread2 t2=new MyThread2();
        MyThread3 t3=new MyThread3();
        t1.setName("A");
        t2.setName("B");
        t3.setName("C");

        t1.start();
        t2.start();
        t3.start();

        System.out.println("MAIN CLOSED");
    }
}

```



Output:

```

D:\java_prog\MultiThreading>java MainDemo
MAIN CLOSED
C: Welcome
B: Hello
A: Good Morning
A: Good Morning
B: Hello
A: Good Morning
C: Welcome
A: Good Morning
B: Hello
A: Good Morning
A: Good Morning
C: Welcome
B: Hello
A: Good Morning
A: Good Morning

```

12.2) Write a Java program to solve Producer-Consumer problem using synchronization.**Source Code:**

```

import java.util.*;
//critical section: buffer object is passed into both producer and consumer threads
class Buffer
{
    String data;
    boolean avail=false;
    public synchronized void put(String data)    //used by producer
    {
        while (avail==true)
        {
            try
            {
                wait ();
            }
            catch (InterruptedException ie)
            {System.out.println(ie);}
        }
        this.data=data;
        System.out.println("Produced:"+data);
        avail = true;
        notify();
    }
    public synchronized String get()            // used by consumer
    {
        while(avail==false)
        {
            try
            {
                wait ();
            }
            catch (InterruptedException ie)
            {System.out.println(ie);}
        }
        avail = false;
        notify ();
        return data;
    }
}
class Producer extends Thread
{
    String data;
    Scanner sc=new Scanner(System.in);
    Buffer buf;
    public Producer(Buffer buf)
    {
        super("Producer");
        this.buf=buf;
    }
}

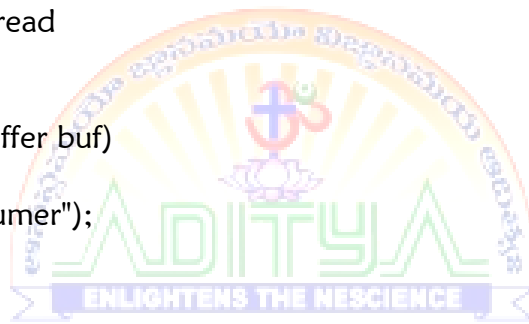
```

```

    }
    public void run ()
    {
        try
        {
            while (true)
            {
                System.out.println("Enter data");
                data=sc.nextLine();
                buf.put(data);
                Thread.sleep(500);           //InterruptedException
            }
        }
        catch (InterruptedException e)
        {
            System.out.println(e);
        }
    }
}

class Consumer extends Thread
{
    Buffer buf;
    public Consumer(Buffer buf)
    {
        super ("Consumer");
        this.buf=buf;
    }
    public void run ()
    {
        try
        {
            while (true)
            {
                System.out.println("Consumed:"+ buf.get());
                Thread.sleep (500);
            }
        }
        catch(InterruptedException e)
        {
            System.out.println(e);
        }
    }
}

```



```

class MainDemo
{
    public static void main (String args [])
    {
        Buffer buf = new Buffer ();
        Producer p = new Producer (buf);
        Consumer c = new Consumer (buf);

        p.start();
        c.start();
    }
}

```

Output:

D:\java_prog\MultiThreading>java MainDemo

Enter data

CSE

Produced:CSE

Consumed:CSE

Enter data

ECE

Produced:ECE

Consumed:ECE

Enter data

EEE

Produced:EEE

Consumed:EEE

Enter data

MECH

Produced:MECH

Consumed:MECH

Enter data



Augmented Experiments**Date:**

(Any 2 of the given experiments can be performed)

13) Write a Java program that simulates a traffic light. The program lets the user select one of three lights: red, yellow, or green with radio buttons. On selecting a button, an appropriate message with "Stop" or "Ready" or "Go" should appear above the buttons in selected color.

Source Code:

```
import java.awt.*;
import java.awt.event.*;
class TrafficLightsExample extends Frame implements ItemListener
{
    CheckboxGroup grp;
    Checkbox redLight, yellowLight, greenLight;
    Label msg;
    public TrafficLightsExample()
    {
        grp=new CheckboxGroup();
        redLight = new Checkbox("Red", grp, false);
        yellowLight = new Checkbox("Yellow", grp, false);
        greenLight = new Checkbox("Green", grp, false);
        msg = new Label("");

        setLayout(new FlowLayout());
        redLight.addItemListener(this);
        yellowLight.addItemListener(this);
        greenLight.addItemListener(this);

        add(redLight);
        add(yellowLight);
        add(greenLight);
        add(msg);
        msg.setFont(new Font("Serif", Font.BOLD, 20));
    }
    public void itemStateChanged(ItemEvent ie)
    {
        redLight.setForeground(Color.BLACK);
        yellowLight.setForeground(Color.BLACK);
        greenLight.setForeground(Color.BLACK);

        if(redLight.getState() == true)
        {
            redLight.setForeground(Color.RED);
            msg.setForeground(Color.RED);
            msg.setText("STOP");
        }
        else if(yellowLight.getState() == true)
        {
            yellowLight.setForeground(Color.YELLOW);
            msg.setForeground(Color.YELLOW);
            msg.setText("READY");
        }
    }
}
```



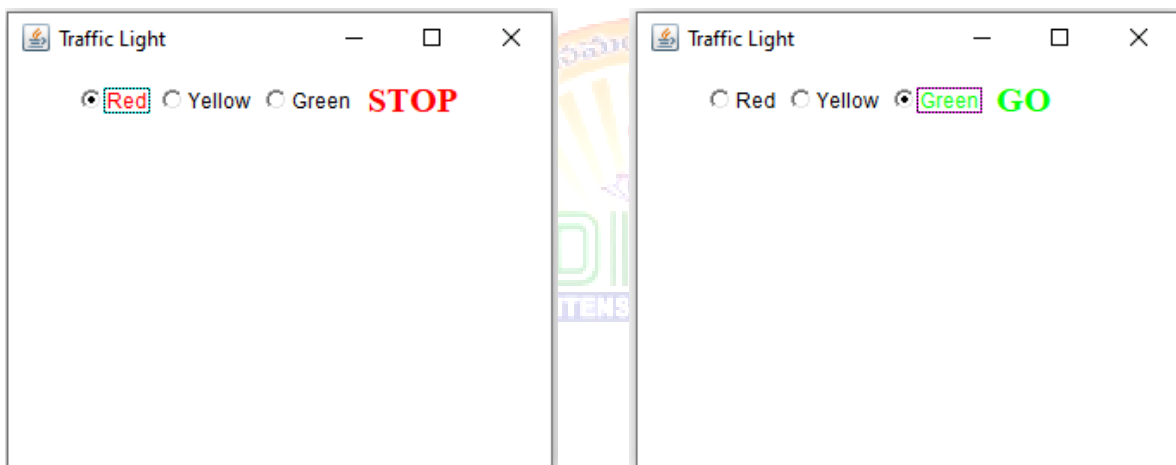
```

    }
    else
    {
        greenLight.setForeground(Color.GREEN);
        msg.setForeground(Color.GREEN);
        msg.setText("GO");
    }
}

public static void main(String args[])
{
    TrafficLightsExample tf=new TrafficLightsExample();
    tf.setSize(500,500);
    tf.setTitle("Traffic Light");
    tf.setVisible(true);
}
}

```

Output:



14) Develop a java program that receives an integer in one text field, and computes its factorial Value and returns it in another text field, when the button named “Compute” clicked.

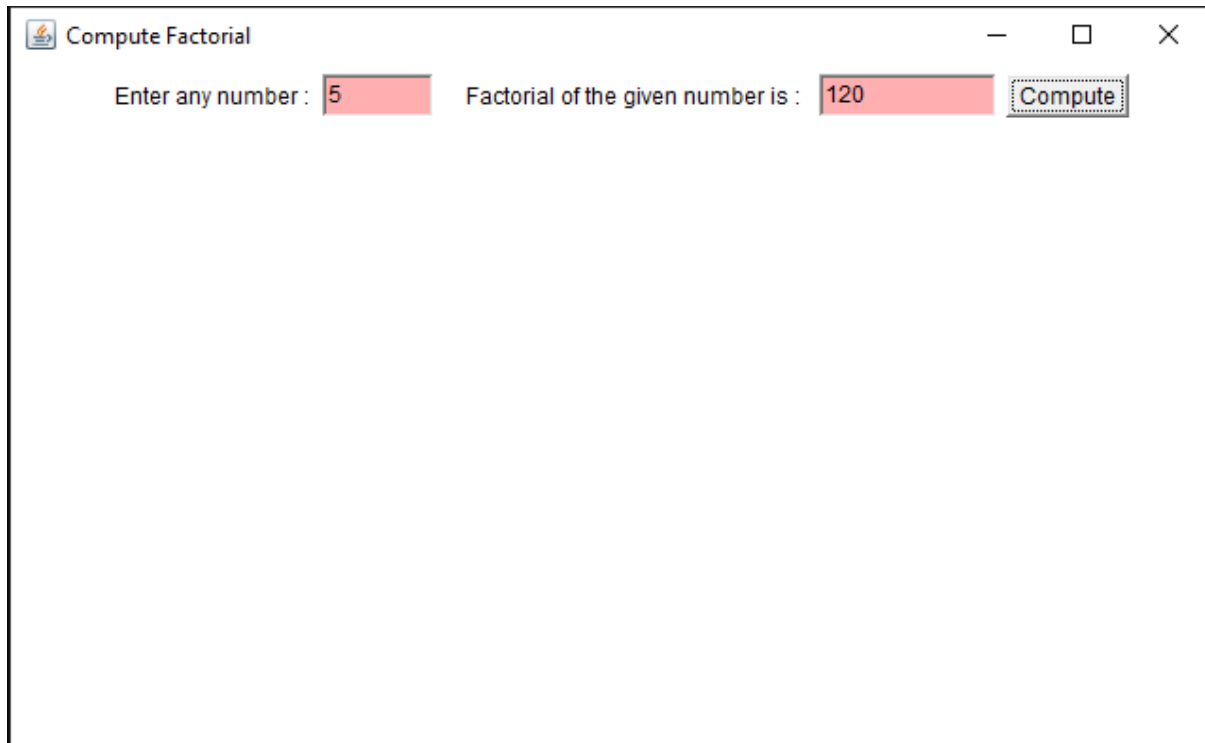
Source Code:

```
import java.awt.*;
import java.awt.event.*;
class Factorial extends Frame implements ActionListener
{
    TextField input,output;
    Button compute;
    int fact=0;
    Label inp,opt;
    public Factorial()
    {
        setLayout(new FlowLayout());
        compute=new Button("Compute");
        inp=new Label("Enter any number :",Label.RIGHT);
        opt=new Label("Factorial of the given number is : ",Label.RIGHT);
        input=new TextField(5);
        output=new TextField(10);
        input.setBackground(Color.pink);
        output.setBackground(Color.pink);
        add(inp);
        add(input);
        add(opt);
        add(output);
        add(compute);
        output.setText("0");
        output.setEditable(false);
        input.addActionListener(this);
        output.addActionListener(this);
        compute.addActionListener(this);
    }
    public void actionPerformed(ActionEvent ae)
    {
        String str=ae.getActionCommand();
        if(str.equals("Compute"))
        {
            fact=1;
            int n=Integer.parseInt(input.getText());
            if(n<=12)
            {
                for(int i=n;i>=2;i--)
                {
                    fact=fact*i;
                    output.setText(""+fact);
                }
            }
            else
            {
                fact=-1;
                output.setText(fact+"");
            }
        }
    }
}
```

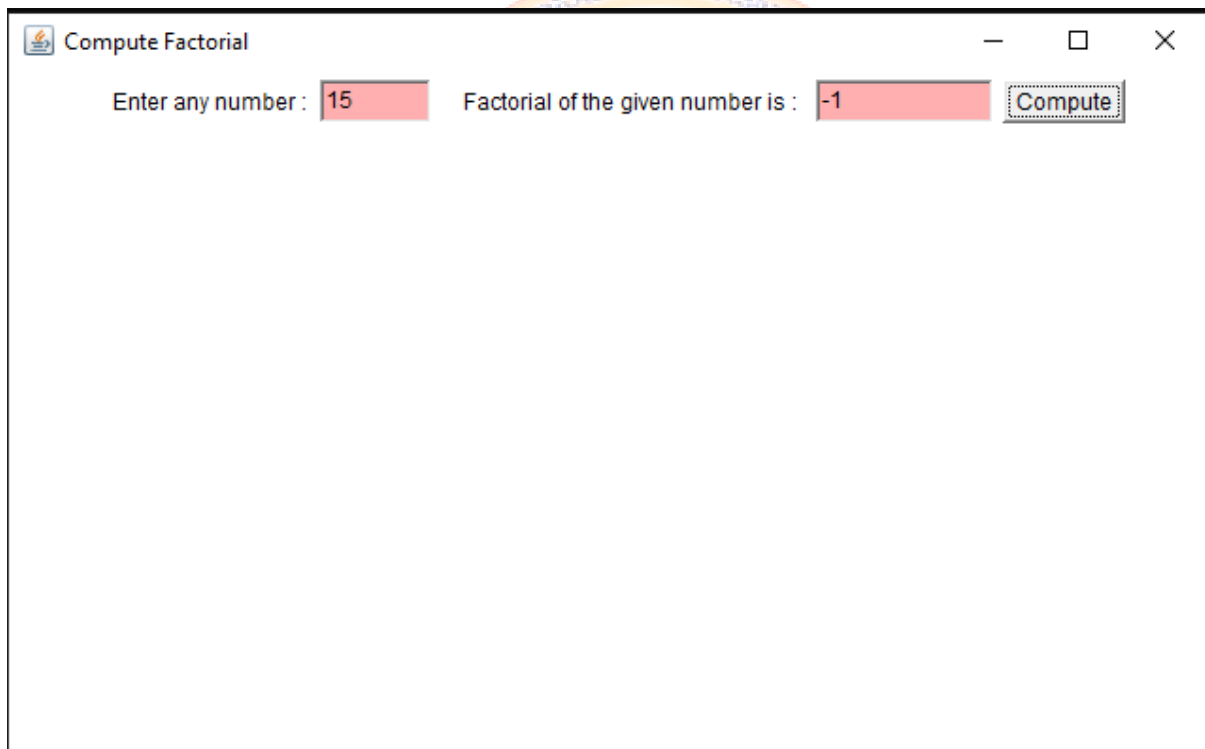


```
    }  
}  
public static void main(String args[])  
{  
    Factorial f=new Factorial();  
    f.setSize(400,400);  
    f.setTitle("Compute Factorial");  
    f.setVisible(true);  
}  
}
```



Output:

A Java Swing window titled "Compute Factorial" with standard window controls (minimize, maximize, close). The window contains two text labels: "Enter any number :" and "Factorial of the given number is :". The first label is followed by a text input field containing the number "5". The second label is followed by a text output field containing the number "120". To the right of the output field is a button labeled "Compute".



A Java Swing window titled "Compute Factorial" with standard window controls (minimize, maximize, close). The window contains two text labels: "Enter any number :" and "Factorial of the given number is :". The first label is followed by a text input field containing the number "15". The second label is followed by a text output field containing the number "-1". To the right of the output field is a button labeled "Compute".