

Program No 1: Write a program that asks a user to enter an integer n and then determines whether n is prime or not. Your program can perform this by dividing n by all integers from 2 to n-1 and by checking whether the remainder is 0.

```
import java.util.*;

public class PrimeNumber {

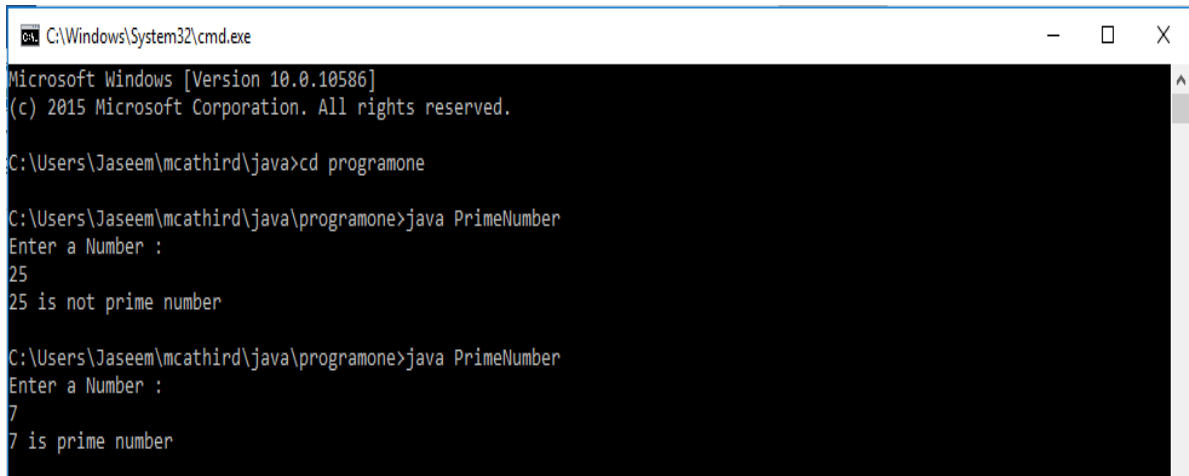
    public static void main(String[] args) {

        int m=0,flag=0;
        int n = 0;
        Scanner in = new Scanner(System.in);
        System.out.println("Enter a Number :");
        n = in.nextInt();
        m = n/2;

        if( n == 0 || n == 1 )
        {
            System.out.println(n+" is not prime number");
        }else{
            for(int i = 2; i <= m; i++)
            {
                if(n%i == 0)
                {
                    flag=1;
                    break;
                }
            }
        }

        if(flag==0)
        {
            System.out.println(n+" is prime number");
        }else{
            System.out.println(n+" is not prime number");
        }
    }
}
```

Output:



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\Jaseem\mcathird\java>cd programone

C:\Users\Jaseem\mcathird\java\programone>java PrimeNumber
Enter a Number :
25
25 is not prime number

C:\Users\Jaseem\mcathird\java\programone>java PrimeNumber
Enter a Number :
7
7 is prime number
```

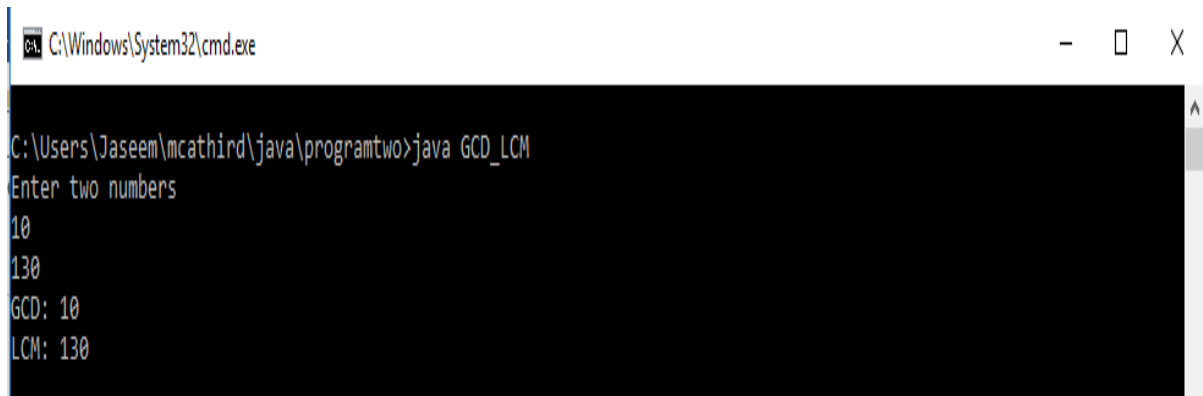
Program No 2: Write a Java program to find GCD and LCM of two numbers (GCD is calculated using Euclidean Algorithm. LCM is found using factorization method.).

```
import java.util.*;

public class GCD_LCM {
    public static void main(String[] args) {
        int gcd, lcm, n1, n2;
        Scanner input = new Scanner(System.in);
        System.out.println("Enter two numbers");
        n1 = input.nextInt();
        n2 = input.nextInt();
        gcd = gcd(n1, n2);
        System.out.println("GCD: " + gcd);
        System.out.println("LCM: " + ((n1 * n2) / gcd) );
    }

    public static int gcd(int n1, int n2){
        while(n1 != n2){
            if(n1 > n2){
                n1 -= n2;
            }else{
                n2 -= n1;
            }
        }
        return n1;
    }
}
```

Output:



```
C:\Windows\System32\cmd.exe

C:\Users\Jaseem\mcathird\java\programtwo>java GCD_LCM
Enter two numbers
10
130
GCD: 10
LCM: 130
```

Program No 3: Write a program that computes $C(n, k)$, i.e. the number of k -element subsets of a set with n elements. Remember that $C(n, k) = n!/(k!(n-k)!)$. Your program should ask the user to enter n and k , and compute and print $C(n, k)$.

```
import java.util.*;

public class Combination {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

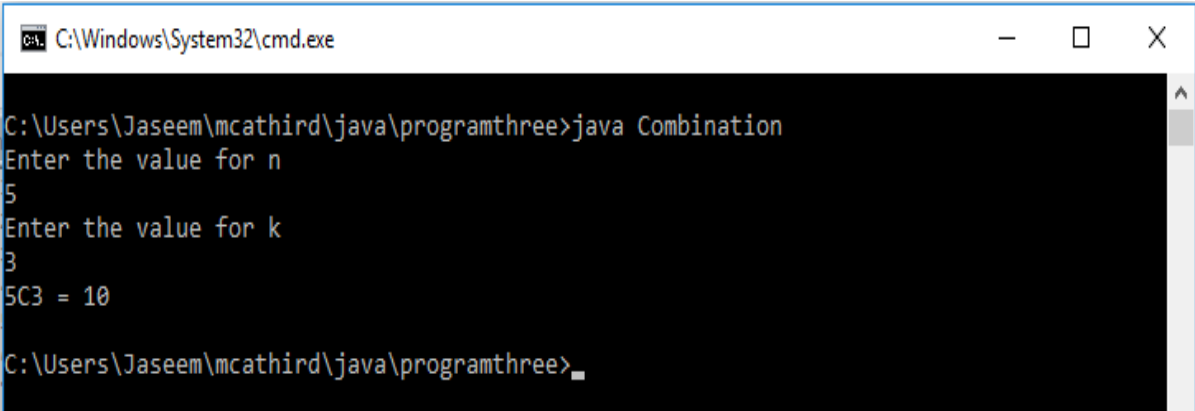
        int n = 0, k = 0;
        System.out.println("Enter the value for n");
        n = input.nextInt();

        System.out.println("Enter the value for k");
        k = input.nextInt();

        System.out.print(n + "C" + k + " = ");
        System.out.println(factorial(n) /
                           (factorial(k) * factorial(n - k)));
    }

    public static int factorial(int n){
        int fact = 1;
        for(int i = n; i > 0; i--){
            fact *= i;
        }
        return fact;
    }
}
```

Output:



```
C:\Windows\System32\cmd.exe

C:\Users\Jaseem\mcathird\java\programthree>java Combination
Enter the value for n
5
Enter the value for k
3
5C3 = 10

C:\Users\Jaseem\mcathird\java\programthree>
```

Program No 4: Write a Java program implement basic queue operations.

```
import java.util.*;

public class Queue {
    public static int MAX = 10;
    public static int[] queue = new int[MAX];
    public static int rear = 0;
    public static int front = 0;

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

        while(true){
            System.out.println("");
            System.out.println("1.Insert \n 2.Delete \n 3.Display \n 4.Exit ");
            System.out.println("Enter any option: ");
            option = input.nextInt();

            switch(option){
                case 1:
                    System.out.println("Enter the value to enter");
                    int val = input.nextInt();
                    insert(val);
                    break;
                case 2:
                    delete();
                    break;
                case 3:
                    display();
                    break;
                case 4:
                    System.exit(0);
                    break;
                default:
                    System.out.println("Invalid Option");
                    break;
            }
        }
    }

    public static void insert(int val){
        if(rear >= MAX){
            System.out.println("Queue is full");
            System.out.println("");
            return;
        } queue[rear] = val;
        rear++;
    }
}
```

```
public static void delete(){
    System.out.println("");
    if(rear <= 0){
        System.out.println("queue is empty");
        System.out.println("");
        return;
    }

    System.out.println("deleted: " + queue[front]);

    for(int i = front; i < rear - 1; i++){
        int temp = queue[i + 1];
        queue[i] = temp;
    }
    rear--;
}

public static void display(){
    System.out.println("");
    if(rear <= 0){
        System.out.println("queue is empty");
        System.out.println("");
        return;
    }

    System.out.println("Element in the queue are: ");
    for(int i= 0; i < rear; i++){
        System.out.println(queue[i]);
    }
}
}
```


Output:

```
C:\Windows\System32\cmd.exe

C:\Users\Jaseem\mcathird\java\programfour>java Queue

1.Insert
2.Delete
3.Display
4.Exit
Enter any option:
1
Enter the value to enter
10

1.Insert
2.Delete
3.Display
4.Exit
Enter any option:
1
Enter the value to enter
20

1.Insert
2.Delete
3.Display
4.Exit
Enter any option:
1
Enter the value to enter
30
```

```
C:\Windows\System32\cmd.exe

1.Insert
2.Delete
3.Display
4.Exit
Enter any option:
3

Element in the queue are:
10
20
30

1.Insert
2.Delete
3.Display
4.Exit
Enter any option:
2

deleted: 30

1.Insert
2.Delete
3.Display
4.Exit
Enter any option:
4

C:\Users\Jaseem\mcathird\java\programfour>
```

Program No 5: Write a Java program to count the frequency of words, characters in the given line of text.

```
import java.util.*;

class FrequencyCounter {

    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);
        String[] rWords;
        ArrayList<String> nWords = new ArrayList<String>();
        ArrayList<Integer> nFreq = new ArrayList<Integer>();
        System.out.println("Enter the line");
        String line = scan.nextLine();
        line = line.trim();
        line = line.replaceAll(" +", " ");
        rWords = line.split(" ");

        for(int i = 0; i < rWords.length; i++){
            String current = rWords[i];
            int repeatition = 0;

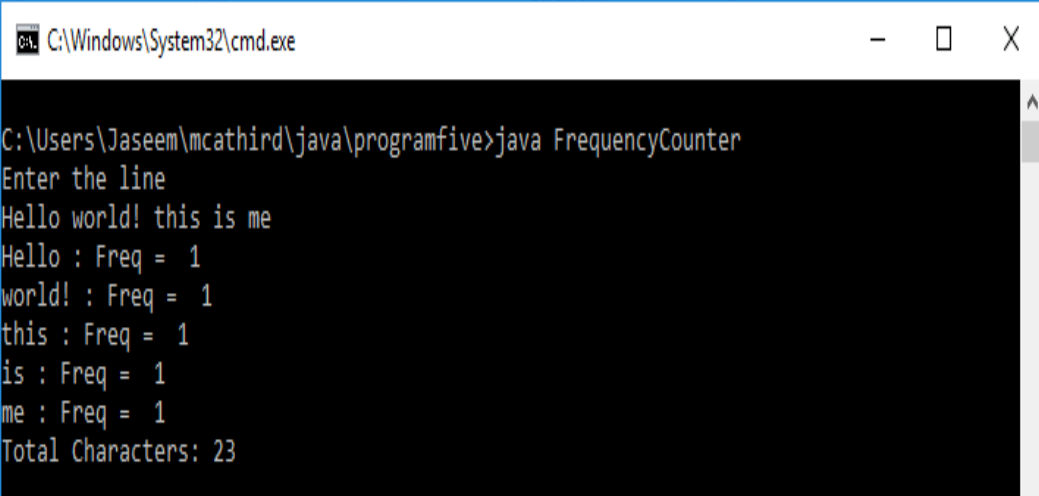
            for(int j = 0; j < rWords.length; j++){
                if(current.equals(rWords[j])){
                    repeatition++;
                }
            }

            if(!nWords.contains(current)){
                nWords.add(current);
                nFreq.add(repeatition);
            }
        }

        for(int i = 0; i < nWords.size(); i++){
            System.out.println(nWords.get(i) + " : Freq = " +
                               nFreq.get(i));
        }

        System.out.println("Total Characters: " + line.length());
    }
}
```

Output:



```
C:\Windows\System32\cmd.exe

C:\Users\Jaseem\mcathird\java\programfive>java FrequencyCounter
Enter the line
Hello world! this is me
Hello : Freq = 1
world! : Freq = 1
this : Freq = 1
is : Freq = 1
me : Freq = 1
Total Characters: 23
```

Program No 6: Write a Java program that creates an object and initializes its data members using constructor. Use constructor overloading concept.

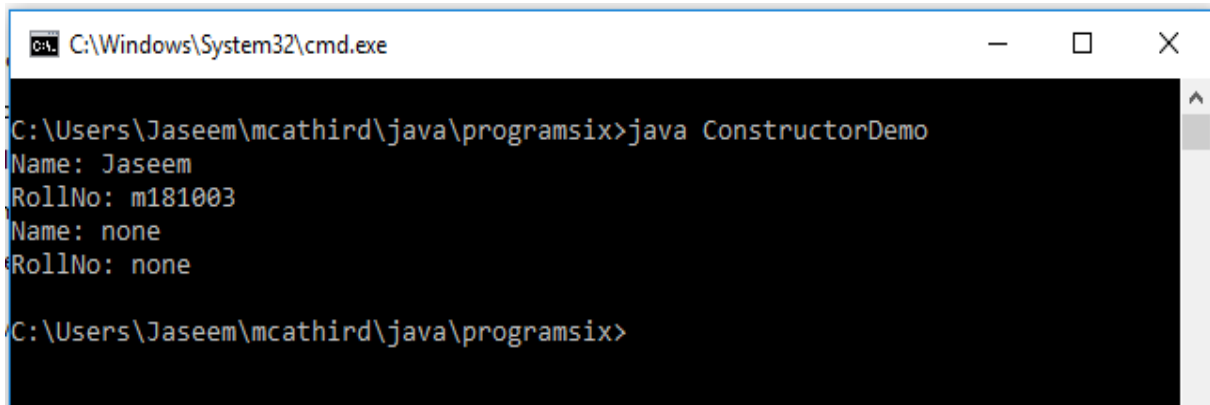
```
class ConstructorDemo{
    String name = "none";
    String rollno = "none";

    public ConstructorDemo(){
    }

    public ConstructorDemo(String name, String rollno){
        this.name = name;
        this.rollno = rollno;
    }

    public static void main(String[] args){
        ConstructorDemo cd = new ConstructorDemo("Jaseem", "m181003");
        System.out.println("Name: " + cd.name );
        System.out.println("RollNo: " + cd.rollno );
        ConstructorDemo cd2 = new ConstructorDemo();
        System.out.println("Name: " + cd2.name );
        System.out.println("RollNo: " + cd2.rollno );
    }
}
```

Output:

A screenshot of a Windows command prompt window. The title bar shows the path 'C:\Windows\System32\cmd.exe'. The command prompt shows the following text:

```
C:\Users\Jaseem\mcathird\java\programsix>java ConstructorDemo
Name: Jaseem
RollNo: m181003
Name: none
RollNo: none
C:\Users\Jaseem\mcathird\java\programsix>
```

Program No 7: Write a Java Program to implement inheritance and demonstrate use of method overriding (example: Bank account/Employee) .

```
public class Inheritance_Overriding{
    public static void main(String[] args){
        Manager manager = new Manager("Mr.Robot", 7313);

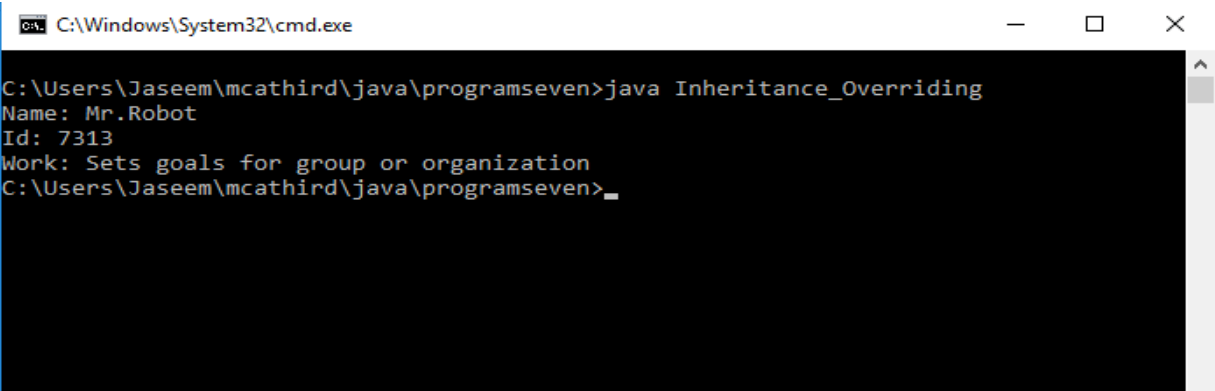
        System.out.println("Name: " + manager.name );
        System.out.println("Id: " + manager.id );
        System.out.print("Work: " + manager.doesWork());
    }
}

class Employee{
    String name;
    int id;

    public Employee(String n, int i){
        name = n;
        id = i;
    }
    public String doesWork(){
        return "Some Work";
    }
}

class Manager extends Employee{
    public Manager(String n, int i){
        super(n, i);
    }
    public String doesWork(){
        return "Sets goals for group or organization";
    }
}
```

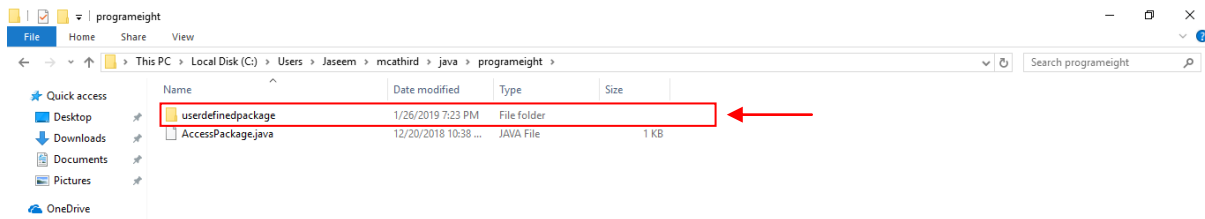
Output:



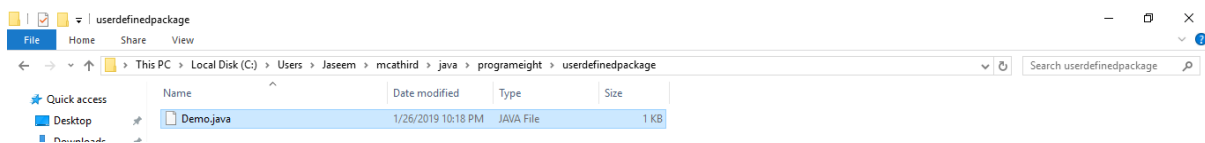
```
C:\Windows\System32\cmd.exe

C:\Users\Jaseem\mcathird\java\programseven>java Inheritance_Overriding
Name: Mr.Robot
Id: 7313
Work: Sets goals for group or organization
C:\Users\Jaseem\mcathird\java\programseven>_
```

Program No 8: Write a program to demonstrate use of user defined package by importing the package and access the member variable of classes contained in the package.



Create a folder “userdefinedpackage” and inside it create a java file “Demo.java”& compile.



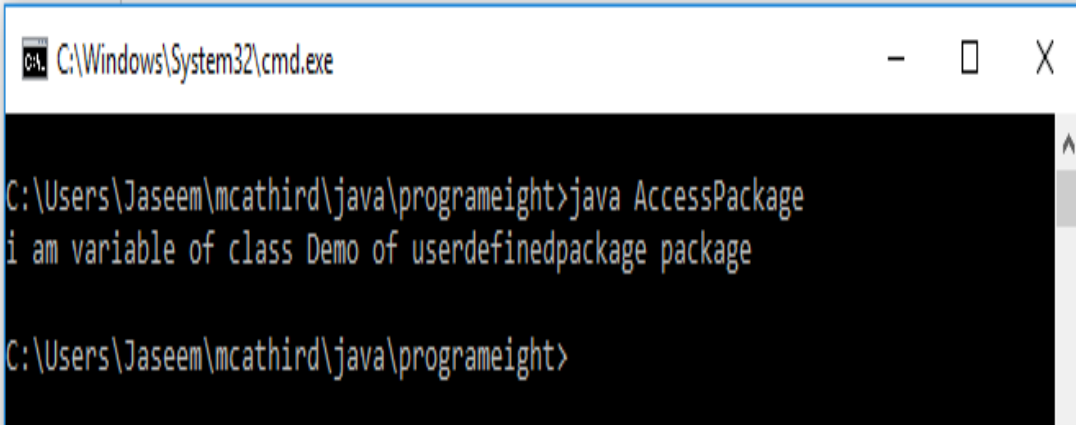
```
package userdefinedpackage;
```

```
public class Demo{  
    public String name;  
  
    public Demo(){  
        name = "i am variable of class Demo of userdefinedpackage package";  
    }  
}
```

Create “AccessPackage.java” outside folder “userdefinedpackage” and compile.

```
import userdefinedpackage.*;  
  
class AccessPackage{  
    public static void main(String[] args){  
        Demo d = new Demo();  
        System.out.println(d.name);  
    }  
}
```


Output:



```
C:\Windows\System32\cmd.exe

C:\Users\Jaseem\mcathird\java\programeight>java AccessPackage
i am variable of class Demo of userdefinedpackage package

C:\Users\Jaseem\mcathird\java\programeight>
```

Program No 9: Write a program to demonstrate use of interfaces for two different classes. Interface should also include constants along with function prototypes.

```
public class InterfaceDemo{
    public static void main(String[] args){
        Square s = new Square(10.0);
        Rectangle r = new Rectangle(9.0, 4.0);
        System.out.println("Area of square: " + s.calculateArea());
        System.out.println("Area of rectangle: " + r.calculateArea());
    }
}

interface Shape{
    double calculateArea();
}

class Square implements Shape{
    double side;

    public Square(double s){
        side = s;
    }

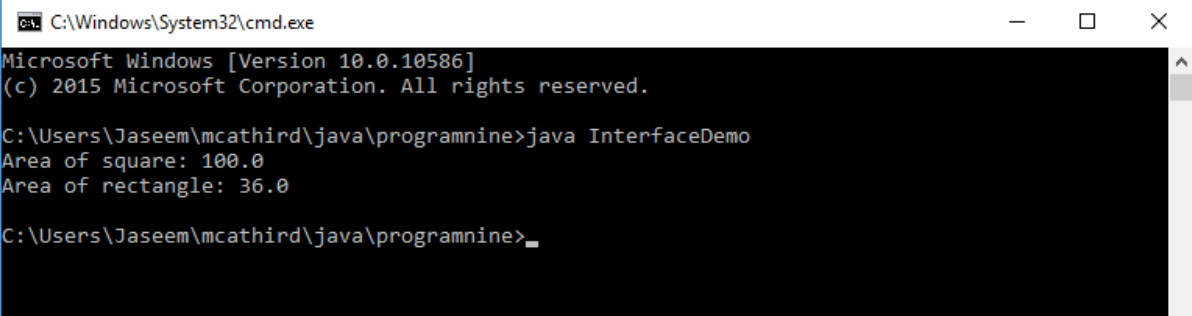
    public double calculateArea(){
        return side * side;
    }
}

class Rectangle implements Shape{
    double length, breadth;

    public Rectangle(double l, double b){
        length = l;
        breadth = b;
    }

    public double calculateArea(){
        return length * breadth;
    }
}
```

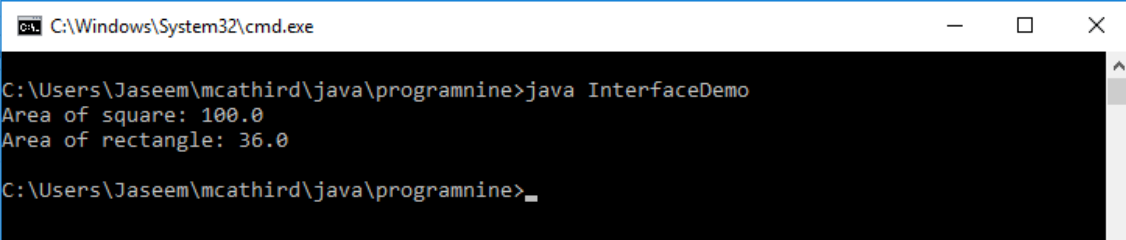
Output:



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\Jaseem\mcathird\java\programnine>java InterfaceDemo
Area of square: 100.0
Area of rectangle: 36.0

C:\Users\Jaseem\mcathird\java\programnine>_
```



```
C:\Windows\System32\cmd.exe

C:\Users\Jaseem\mcathird\java\programnine>java InterfaceDemo
Area of square: 100.0
Area of rectangle: 36.0

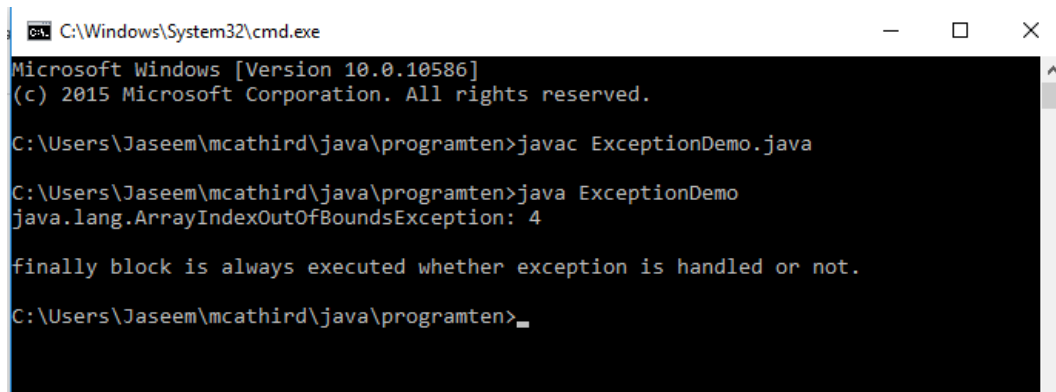
C:\Users\Jaseem\mcathird\java\programnine>_
```

Program No 10: Write a java program to implement exception handling using multiple catch statements. Also include code to identify the significance of finally block in handling exceptions.

```
class ExceptionDemo{
    public static void main(String[] args){
        int a = 4;
        int b = 3;
        int c;
        int arr[] = new int[4];

        try{
            arr[a] = 10;
            c = arr[b] / 0;
        }catch(ArrayIndexOutOfBoundsException e){
            System.out.println(e);
        }catch(ArithmeticException e){
            System.out.println(e);
        }finally{
            System.out.println("\nfinally block is always executed
whether exception is handled or not.");
        }
    }
}
```

Output:



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\Jaseem\mcathird\java\programten>javac ExceptionDemo.java

C:\Users\Jaseem\mcathird\java\programten>java ExceptionDemo
java.lang.ArrayIndexOutOfBoundsException: 4

finally block is always executed whether exception is handled or not.

C:\Users\Jaseem\mcathird\java\programten>_
```

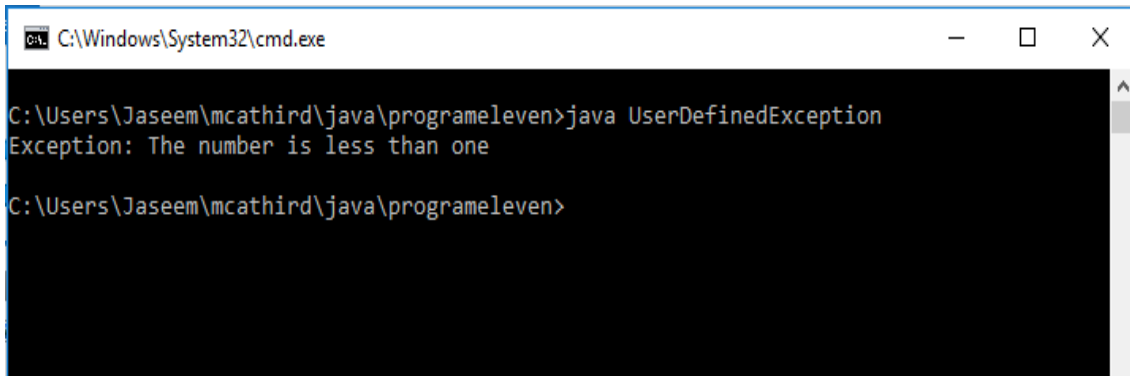
Program No 11: Write a program to implement the concept of Exception Handling by creating user defined exceptions.

```
class UserDefinedException{
    public static void main(String[] args){
        int a = 0;

        try{
            if( a < 1)
                throw new MyException();
        }catch(MyException e){
            System.out.println(e.getMessage());
        }
    }
}

class MyException extends Exception{
    public String getMessage(){
        return "The number is less than one";
    }
}
```

Output:



```
cmd C:\Windows\System32\cmd.exe

C:\Users\Jaseem\mcathird\java\programeleven>java UserDefinedException
Exception: The number is less than one

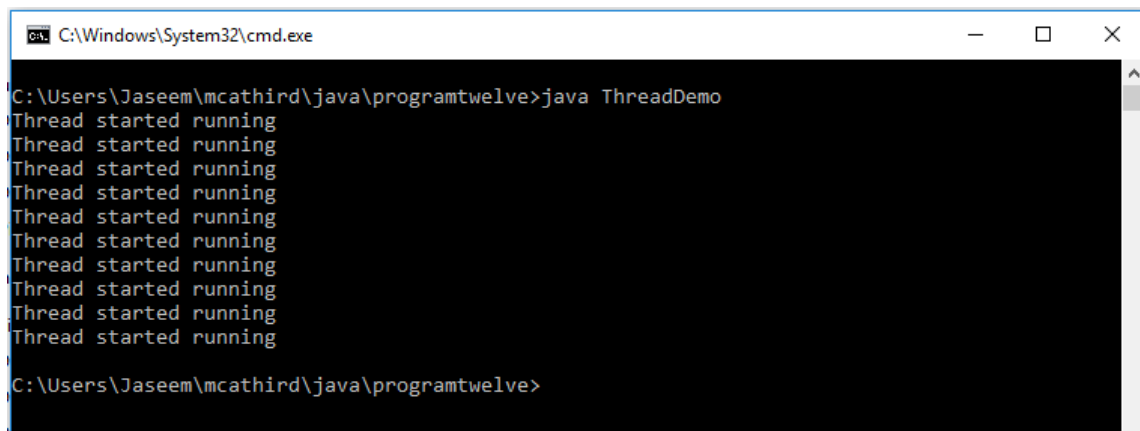
C:\Users\Jaseem\mcathird\java\programeleven>
```

Program No 12: Illustrate creation of thread by extending Thread class/ implementing runnable interface.

```
class ThreadDemo{
    public static void main(String[] args){
        Thread thread = new Thread(new MyThreadDemo());
        thread.start();
    }
}

class MyThreadDemo implements Runnable{
    public void run(){
        for(int i = 0; i < 10; i++)
            System.out.println("Thread started running");
    }
}
```


Output:



```
C:\Windows\System32\cmd.exe

C:\Users\Jaseem\mcathird\java\programtwelve>java ThreadDemo
Thread started running
Thread started running
Thread started running
Thread started running
Thread started running
Thread started running
Thread started running
Thread started running
Thread started running
Thread started running
Thread started running

C:\Users\Jaseem\mcathird\java\programtwelve>
```

Program No 13: Write a Java program that creates three threads. First thread displays “Good Morning” every one second, the second thread displays “Hello” every five seconds and the third thread displays “Welcome” every ten seconds.

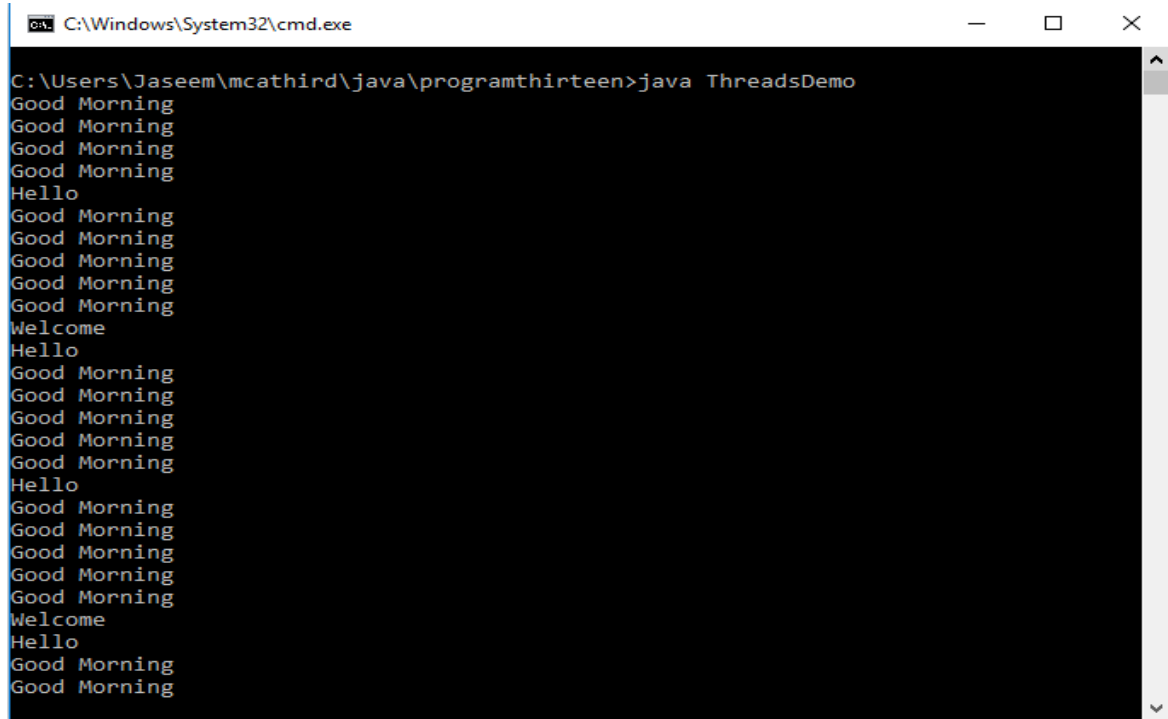
```
public class ThreadsDemo{
    public static void main(String[] args) throws InterruptedException{
        ThreadOne th1 = new ThreadOne();
        ThreadTwo th2 = new ThreadTwo();
        ThreadThree th3 = new ThreadThree();
        th1.start(); th2.start(); th3.start();
    }
}

class ThreadOne extends Thread{
    public void run(){
        while(true){
            try{
                Thread.sleep(1000);
                System.out.println("Good Morning");
            }catch(InterruptedException e){
                System.out.println(e);
            }
        }
    }
}

class ThreadTwo extends Thread{
    public void run(){
        while(true){
            try{
                Thread.sleep(5000);
                System.out.println("Hello");
            }catch(InterruptedException e){
                System.out.println(e);
            }
        }
    }
}

class ThreadThree extends Thread{
    public void run(){
        while(true){
            try{
                Thread.sleep(10000);
                System.out.println("Welcome");
            }catch(InterruptedException e){
```

Output:



```
C:\Windows\System32\cmd.exe
C:\Users\Jaseem\mcathird\java\programthirteen>java ThreadsDemo
Good Morning
Good Morning
Good Morning
Good Morning
Hello
Good Morning
Good Morning
Good Morning
Good Morning
Good Morning
Welcome
Hello
Good Morning
Good Morning
Good Morning
Good Morning
Good Morning
Hello
Good Morning
Good Morning
Good Morning
Good Morning
Good Morning
Welcome
Hello
Good Morning
Good Morning
```

Program No 14: Illustrate thread join concept.

```
public class ThreadJoinExample {
    public static void main(String[] args) {
        MyThread t1 = new MyThread("t1");
        MyThread t2 = new MyThread("t2");
        MyThread t3 = new MyThread("t3");
        t1.start();

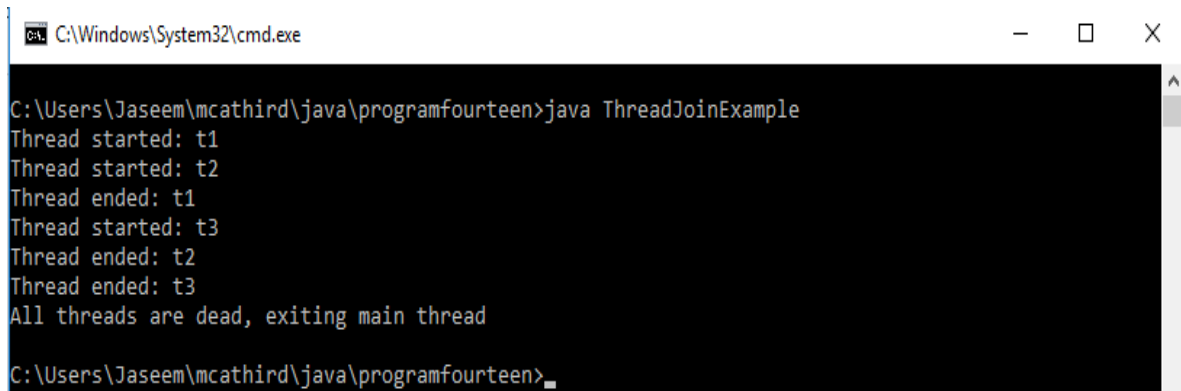
        try {
            //start 2nd thread after waiting for 2 seconds or if it's dead
            t1.join(2000);
            t2.start();

            //start 3rd thread only when first thread is dead
            t1.join();
            t3.start();

            //let all threads finish execution before finishing main thread
            t1.join();
            t2.join();
            t3.join();
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
        System.out.println("All threads are dead, exiting main thread");
    }
}

class MyThread extends Thread{
    public MyThread(String name){
        super(name);
    }
    public void run() {
        System.out.println("Thread started: " + Thread.currentThread().getName());
        try {
            Thread.sleep(4000);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
        System.out.println("Thread ended: "+Thread.currentThread().getName());
    }
}
```

Output:



```
cmd C:\Windows\System32\cmd.exe

C:\Users\Jaseem\mcathird\java\programfourteen>java ThreadJoinExample
Thread started: t1
Thread started: t2
Thread ended: t1
Thread started: t3
Thread ended: t2
Thread ended: t3
All threads are dead, exiting main thread

C:\Users\Jaseem\mcathird\java\programfourteen>_
```

Program No 15: Write a java program to implement mouse events like mouse pressed, mouse released and mouse moved by means of adapter classes.

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

public class MouseAdapterDemo{
    public static void main(String[] args) {
        MyAdapter m = new MyAdapter();
    }
}

class MyAdapter extends MouseAdapter{
    JFrame frame;
    JLabel label;

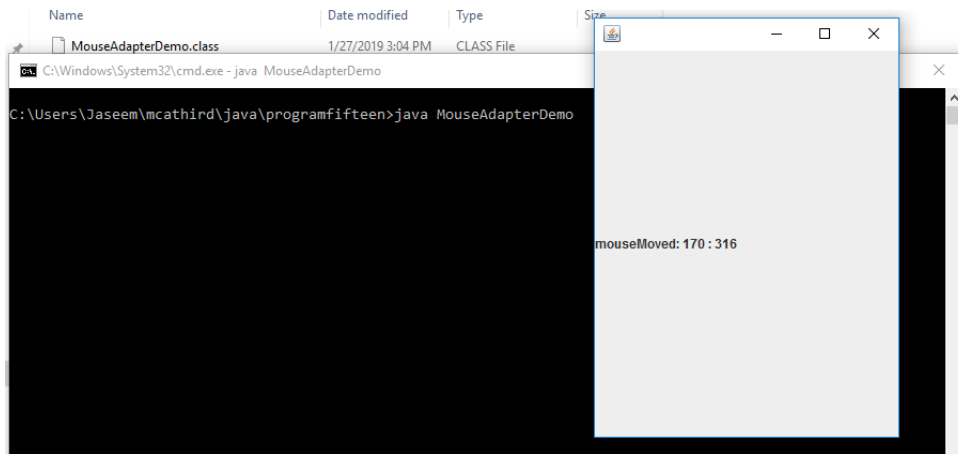
    public MyAdapter(){
        frame = new JFrame();
        label = new JLabel();
        frame.add(label);
        frame.setSize(300, 400);
        frame.setVisible(true);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.addMouseListener(this);
        frame.addMouseMotionListener(this);
    }

    public void mouseMoved(MouseEvent e) {
        super.mouseMoved(e);
        label.setText("mouseMoved: " + e.getX() + " : " + e.getY());
    }

    public void mouseReleased(MouseEvent e) {
        super.mouseReleased(e);
        label.setText("Mouse Released");
    }

    public void mousePressed(MouseEvent e) {
        super.mousePressed(e);
        label.setText("Mouse Pressed");
    }
}
```

Output:



Program No 16: Write a java program that creates a user interface to perform integer divisions. The user enters two numbers in the textfields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an ArithmeticException Display the exception in a message dialog box.

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

class ProgramSixteen extends JFrame implements ActionListener{
    private JButton btnCalculate;
    private JTextField edtOne, edtTwo;
    private JLabel textDisplay;
    private int num1;
    private int num2;
    private int res;

    public ProgramSixteen(){
        btnCalculate = new JButton("Divide");
        btnCalculate.addActionListener(this);
        edtOne = new JTextField("Enter n1");
        edtTwo = new JTextField("Enter n2");
        textDisplay = new JLabel();

        add(textDisplay);
        add(edtOne);
        add(edtTwo);
        add(btnCalculate);

        setTitle("ProgramSixteen");
        setSize(250, 300);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new GridLayout(4,0));
        setVisible(true);
    }

    public void actionPerformed(ActionEvent ae){
        try{
            num1 = Integer.valueOf(edtOne.getText());
            num2 = Integer.valueOf(edtTwo.getText());
            res = num1 / num2;
            textDisplay.setText(res + "");
        }catch(NumberFormatException e){
            JOptionPane.showMessageDialog(this, e);
        }
    }
}
```

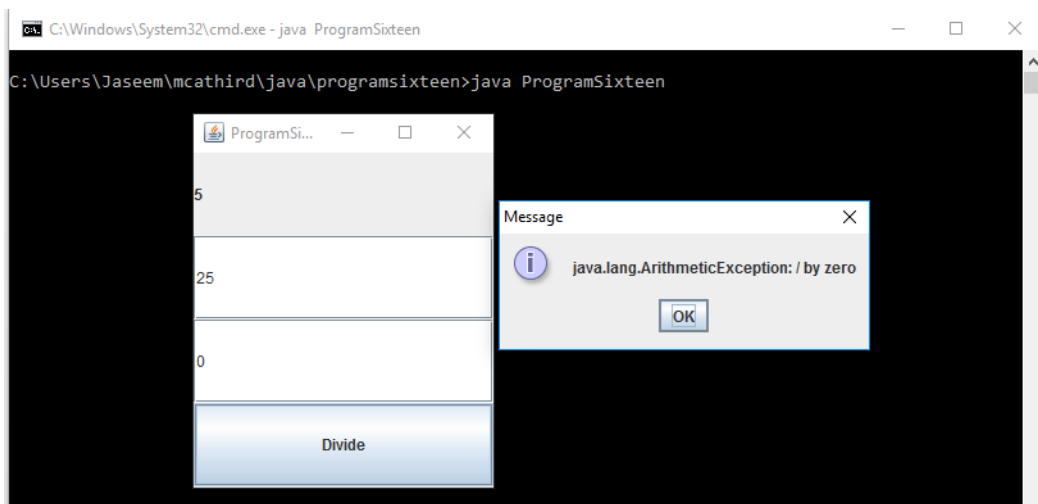
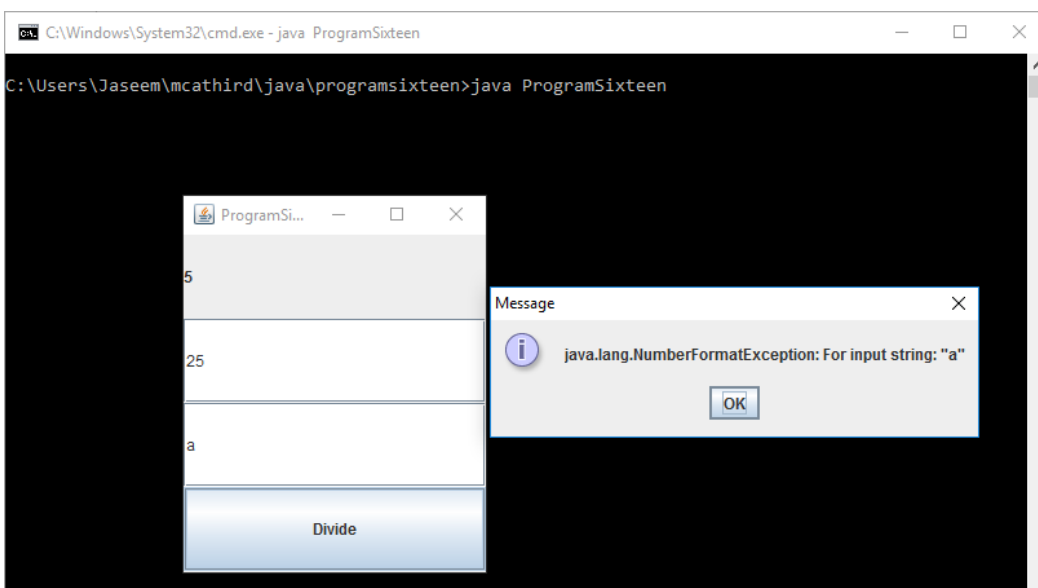
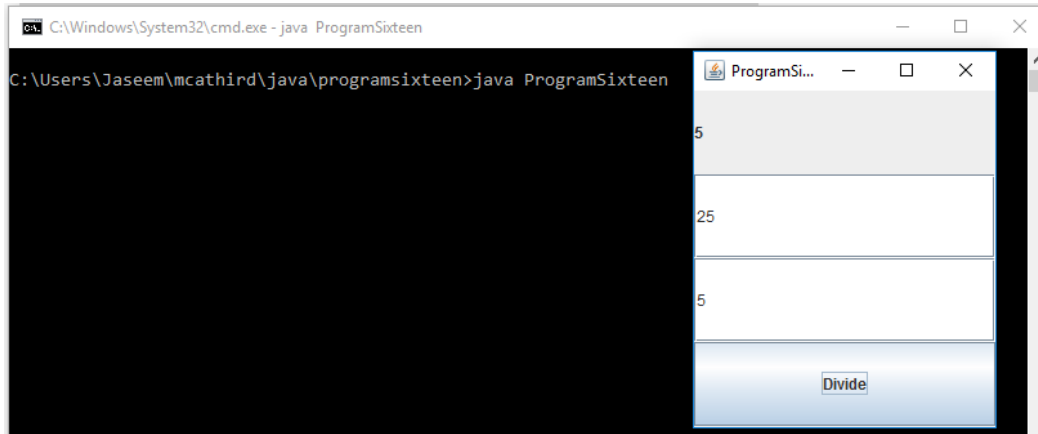


```
        }catch(ArithmeticException e){

public void actionPerformed(ActionEvent ae){
    try{
        num1 = Integer.valueOf(edtOne.getText());
        num2 = Integer.valueOf(edtTwo.getText());
        res = num1 / num2;
        textDisplay.setText(res + "");
    }catch(NumberFormatException e){
        JOptionPane.showMessageDialog(this, e);
    }catch(ArithmeticException e){
        JOptionPane.showMessageDialog(this, e);
    }
}

public static void main(String[] args) {
    ProgramSixteen pS = new ProgramSixteen();
}
}
```

Output:



Program No 17: Write a Java program to illustrate basic calculator using grid layout manager.

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

public class BasicCalculator extends JFrame implements ActionListener{
    private char symbol;
    private int num1 = -1;
    private int num2 = -1;
    private JTextField textField; private JButton btnOne;
    private JButton btnTwo; private JButton btnThree;
    private JButton btnFour; private JButton btnFive;
    private JButton btnSix; private JButton btnSeven;
    private JButton btnEight; private JButton btnNine;
    private JButton btnZero; private JButton btnPlus;
    private JButton btnMinus; private JButton btnMultiply;
    private JButton btnDivide; private JButton btnEquals;
    private JLabel label;

    public BasicCalculator(){
        textField = new JTextField(11);
        btnOne = new JButton("1");
        btnTwo = new JButton("2");
        btnThree = new JButton("3");
        btnFour = new JButton("4");
        btnFive = new JButton("5");
        btnSix = new JButton("6");
        btnSeven = new JButton("7");
        btnEight = new JButton("8");
        btnNine = new JButton("9");
        btnZero = new JButton("0");

        btnPlus = new JButton("+");
        btnMinus = new JButton("-");
        btnMultiply = new JButton("*");
        btnDivide = new JButton("/");
        btnEquals = new JButton("=");

        label = new JLabel();
        setLayout(new FlowLayout());
    }
}
```

```
JPanel displayPanel = new JPanel(new GridLayout(2, 1));
displayPanel.add(label);
displayPanel.add(textField);

JPanel btnPanel = new JPanel(new GridLayout(5,3));
btnPanel.add(btnOne);
btnPanel.add(btnTwo);

btnPanel.add(btnThree);
btnPanel.add(btnFour);    btnPanel.add(btnFive);
btnPanel.add(btnSix);    btnPanel.add(btnSeven);
btnPanel.add(btnEight);    btnPanel.add(btnNine);
btnPanel.add(btnPlus);    btnPanel.add(btnZero);
btnPanel.add(btnMinus);    btnPanel.add(btnMultiply);
btnPanel.add(btnDivide);    btnPanel.add(btnEquals);

btnOne.addActionListener(this);    btnTwo.addActionListener(this);
btnThree.addActionListener(this);    btnFour.addActionListener(this);
btnFive.addActionListener(this);    btnSix.addActionListener(this);
btnSeven.addActionListener(this);    btnEight.addActionListener(this);
btnNine.addActionListener(this);    btnZero.addActionListener(this);
btnPlus.addActionListener(this);    btnMinus.addActionListener(this);
btnMultiply.addActionListener(this);    btnDivide.addActionListener(this);
btnEquals.addActionListener(this);

add(displayPanel);
add(btnPanel);
setDefaultCloseOperation(EXIT_ON_CLOSE);
setSize(180,240);
setVisible(true);
}

public static void main(String[] args) {
    BasicCalculator calc = new BasicCalculator();
}

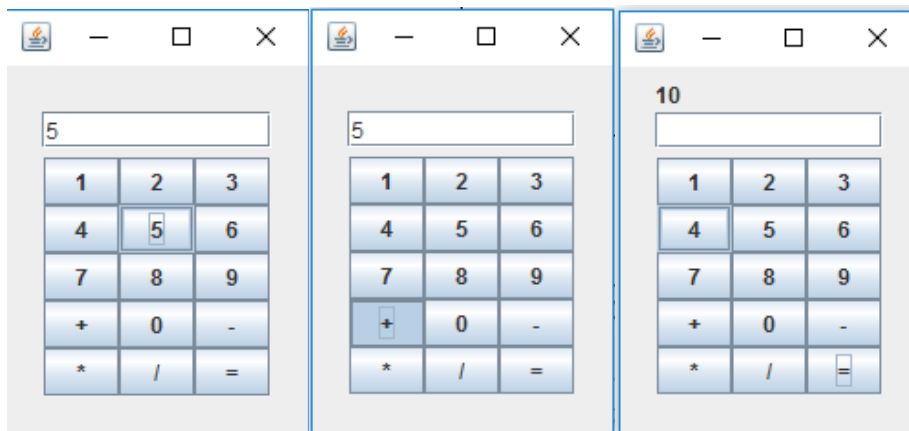
public void actionPerformed(ActionEvent ae){
    String button = ae.getActionCommand();
    String input = textField.getText();
    int res = 0;
    if(button.matches("[+/*]")){
        num1 = Integer.valueOf(input);
        textField.setText("");
        symbol = button.charAt(0);
    }else if(button.matches("[=]")){
        if(num1 == -1)
            return;
    }
```

```
num2 = Integer.valueOf(input);

    if( symbol == '+')
        res = num1 + num2;
    else if(symbol == '-')
        res = num1 - num2;
    else if(symbol == '*')
        res = num1 * num2;
    else if(symbol == '/')
        res = num1 / num2;

    textField.setText("");
    label.setText(String.valueOf(res));
    num1 = -1;
    num2 = -1;
} else {
    input += ae.getActionCommand();
    textField.setText(input);
    label.setText("");
}
}
```

Output:



Program No 18: Develop an applet 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” is clicked.

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

/*<applet code="ComputeFactorial" width="180" height="200" ></applet>*/

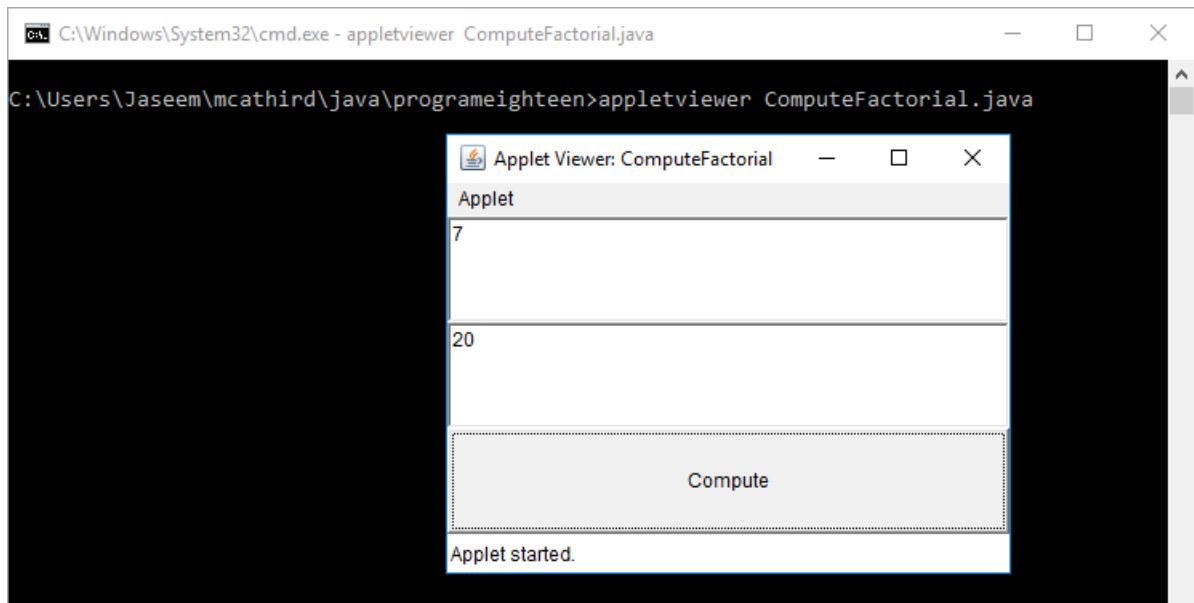
public class ComputeFactorial extends Applet implements ActionListener{
    int fib1 = 0, fib2 = 1 , fib3 = 0;
    TextField txtInput1;
    TextField txtInput2;
    Button btnCompute;

    public void init() {
        setLayout(new GridLayout(3, 0));
        txtInput1 = new TextField();
        txtInput2 = new TextField();
        btnCompute = new Button("Compute");
        btnCompute.addActionListener(this);

        add(txtInput1);
        add(txtInput2);
        add(btnCompute);
    }

    public void actionPerformed(ActionEvent e) {
        String input = txtInput1.getText();
        int n = Integer.valueOf(input);
        int sum = 1;
        if(n < 2){
            txtInput2.setText("" + 0);
        }else if(n == 2){
            txtInput2.setText("" + 1);
        }else if(n > 2){
            for(int i=2; i<n; ++i){
                fib3 = fib1 + fib2;
                fib1 = fib2;    fib2 = fib3;    sum += fib3;
            }
            txtInput2.setText("" + sum);
            fib1 = 0;    fib2 = 1;
        }
    }
}
```

Output:



Program No 19: Write a java program to create student report using applet, read the input using text boxes and display the o/p using buttons.

```
/*<applet code="StudentReport" width="400" height="400"></applet>*/
```

```
import java.applet.*;
```

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

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

```
public class StudentReport extends Applet implements ActionListener{
```

```
    Label lblName, lblRegno, lblJava, lblNetwork;
```

```
    TextField txtName, txtRegno, txtJava, txtNetwork;
```

```
    Button btnGenerate; String name, regno, java, network;
```

```
    public void init(){
```

```
        setLayout(null);
```

```
        lblName = new Label("Name");
```

```
        lblName.setBounds(0,10, 100, 30);
```

```
        txtName = new TextField();
```

```
        txtName.setBounds(100,10, 100, 30);
```

```
        lblRegno = new Label("Regno.");
```

```
        lblRegno.setBounds(0, 40, 100, 30);
```

```
        txtRegno = new TextField();
```

```
        txtRegno.setBounds(100, 40, 100, 30);
```

```
        lblJava = new Label("Java Marks");
```

```
        lblJava.setBounds(0, 70, 100, 30);
```

```
        txtJava = new TextField();
```

```
        txtJava.setBounds(100, 70, 100, 30);
```

```
        lblNetwork = new Label("Network Marks");
```

```
        lblNetwork.setBounds(0, 100, 100, 30);
```

```
        txtNetwork = new TextField();
```

```
        txtNetwork.setBounds(100, 100, 100, 30);
```

```
        btnGenerate = new Button("Generate Report");
        btnGenerate.setBounds(0, 130 ,100, 30);
        btnGenerate.addActionListener(this);

        add(lblName);
        add(txtName);
        add(lblRegno);
        add(txtRegno);
        add(lblJava);
        add(txtJava);
        add(lblNetwork);
        add(txtNetwork);
        add(btnGenerate);
    }

    public void actionPerformed(ActionEvent ae){
        name = txtName.getText();
        regno = txtRegno.getText();
        java = txtJava.getText();
        network = txtNetwork.getText();

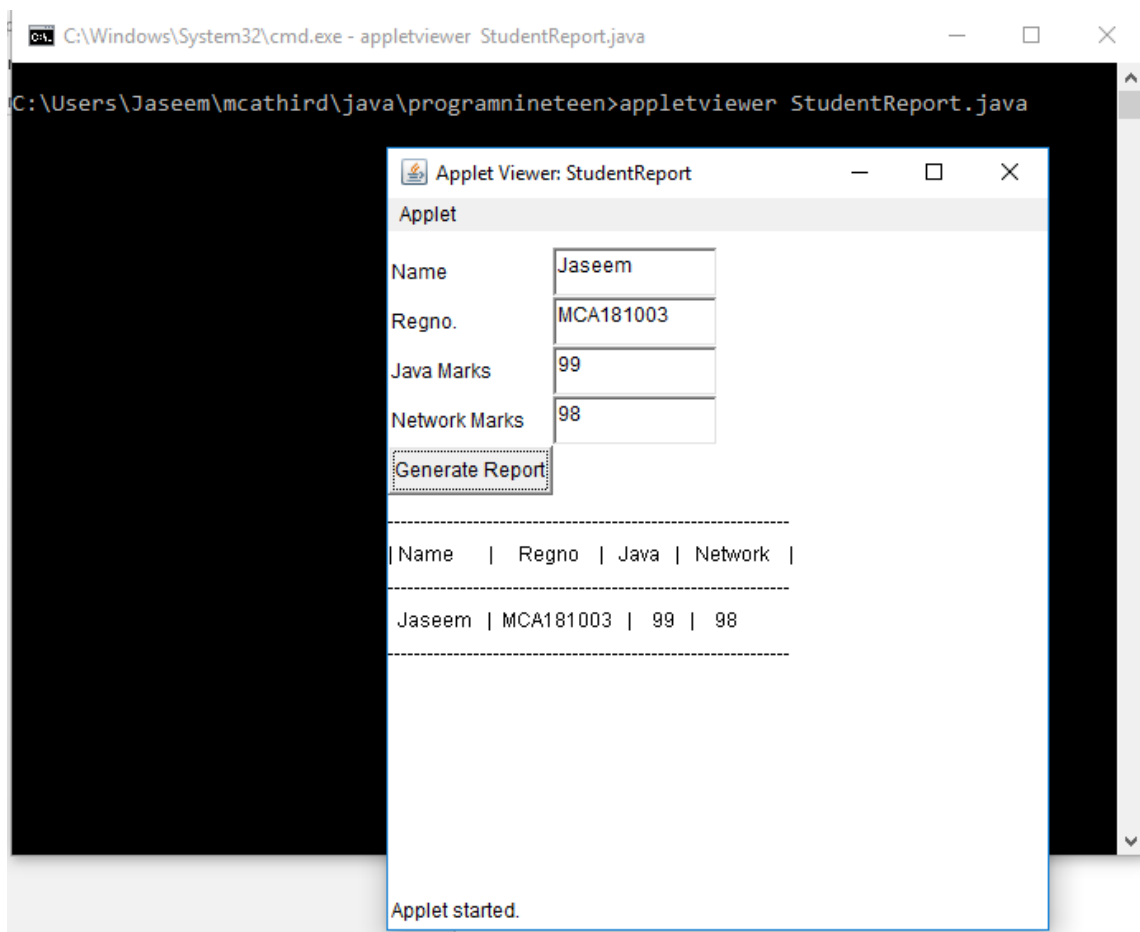
        Graphics g = getGraphics();
        g.drawString("-----", 0,
180);

        g.drawString("| Name      |   Regno   | Java   | Network   |", 0, 200);
        g.drawString("-----", 0,
220);

        g.drawString(" "+ name + "   | " + regno + "   |   " + java + "   |   " +
network, 0, 240);
        g.drawString("-----", 0,
260);

    }
}
```

Output:



Program No 20: Build a Java application for playing the tic-tac-toe game. Description of the game is available on http://en.wikipedia.org/wiki/Tic_tac_toe You are required to implement this game with two classes, TicTacToeGame and TicTacToeTester. TicTacToeTester.java

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

public class TicTacToeTester extends JFrame implements ActionListener{
    private JButton[] btns;
    private boolean playerClicked;
    private int[] board;
    public TicTacToeTester(){
        btns = new JButton[9];
        board = new int[9];
        for(int i=0; i < btns.length; i++){
            btns[i] = new JButton();
            btns[i].setActionCommand(String.valueOf(i));
            btns[i].addActionListener(this);
            add(btns[i]);
        }
        setTitle("TicTacToe Game");
        setLayout(new GridLayout(3,3));
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setSize(400, 400);
        setResizable(false);
        setVisible(true);
    }
    public void actionPerformed(ActionEvent e) {
        JButton button = (JButton) e.getSource();
        for(int i=0; i < btns.length; i++){
            if(button == btns[i]){
                if(playerClicked){
                    button.setText("O");
                    playerClicked = false;
                    board[i] = 2;
                }else{
                    button.setText("X");
                    playerClicked = true;
                    board[i] = 1;
                }
            }
        }
    }
}
```

```
int player = 0;
if(board[0] == board[1] && board[1] == board[2]){
    player = board[0];
}else if(board[3] == board[4] && board[4] == board[5]){
    player = board[3];
}else if(board[6] == board[7] && board[7] == board[8]){
    player = board[6];
}else if(board[0] == board[3] && board[3] == board[6]){
    player = board[0];
}else if(board[1] == board[4] && board[4] == board[7]){
    player = board[1];
}else if(board[2] == board[5] && board[5] == board[8]){
    player = board[2];
}else if(board[0] == board[4] && board[4] == board[8]){
    player = board[0];
}else if(board[2] == board[4] && board[4] == board[6]){
    player = board[2];}
if(player == 1){
    JOptionPane.showMessageDialog(rootPane, "X Win");
    dispose();
}else if(player == 2){
    JOptionPane.showMessageDialog(rootPane, "O Win");
    dispose();
}else{ boolean draw = true;
    for(int i = 0; i < board.length; i++){
        if(board[i] == 0){
            draw = false;
            break; }
    }
    if(draw){
        JOptionPane.showMessageDialog(rootPane, "Draw");
        dispose();}
    }
}
```

TicTacToeGame.java

```
public class TicTacToeGame {
    public static void main(String[] args) {
        TicTacToeTester game = new TicTacToeTester();
    }
}
```

Output:

