

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.

```
package vijay;
import java.util.Scanner;
public class java1
{
    public static void main(String[] args)
    {
        int n;
        boolean flag=false;
        Scanner scan=new Scanner(System.in);
        System.out.println("Enter the number:");
        n=scan.nextInt();
        scan.close();
        for(int i=2;i<n;i++)
        {
            if(n%i==0)
            {
                flag=true; break;
            }
        }
        if(!flag)
            System.out.println("The given number is prime");
        else
            System.out.println("The given number is not prime");
    }
}
```

Output:01

run:

Enter the number:

10

The given number is not prime

BUILD SUCCESSFUL (total time: 3 seconds)

Output:02

run:

Enter the number:

11

The given number is prime

BUILD SUCCESSFUL (total time: 2 seconds)

- 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).**

```
package java2pgm;
import java.util.Scanner;
public class GCD_LCM
{
    public static int gcd(int x,int y)
    {
        int r=0,a,b;
        a=(x>y)?x:y;
        b=(x<y)?x:y;
        r=b;
        while(a%b!=0)
        {
            r=a%b; a=b; b=r;
        }
        return r;
    }
    public static int lcm(int x,int y)
    {
        int a;
        a=(x>y)?x:y;
        while(true)
        {
            if(a%x==0 && a%y==0) return a; ++a;
        }
    }
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        System.out.println("Enter the two numbers: ");
        int x = input.nextInt();
        int y = input.nextInt();
        System.out.println("The GCD of two numbers is: " + gcd(x,y));
        System.out.println("The LCM of two numbers is: " + lcm(x,y));
        input.close();
    }
}
```

Output:1

run:

Enter the two numbers:

4

6

The GCD of two numbers is: 2

The LCM of two numbers is: 12

BUILD SUCCESSFUL (total time: 3 seconds)

Output:2

run:

Enter the two numbers:

3

6

The GCD of two numbers is: 3

The LCM of two numbers is: 6

BUILD SUCCESSFUL (total time: 5 seconds)

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 $C(n, k)$.

```
package java3pgm;
import java.io.*;
import java.util.Scanner;
public class compute
{
    public static void main(String[] args)
    {
        int n,k,fact1=1,fact2=1,a,fact3=1,comb;
        Scanner s=new Scanner(System.in);
        System.out.println("enter the value of n and k");
        n=s.nextInt();
        k=s.nextInt();
        a=n-k;
        for(int i=1;i<n+1;i++)
            fact1=fact1*i;
        for(int i=1;i<k+1;i++)
            fact2=fact2*i;
        for(int i=1;i<a+1;i++)
            fact3=fact3*i;
        comb=fact1/(fact2*fact3);
        System.out.println("The number of combinations available are:"+comb);
    }
}
```

Output:1

```
enter the value of n and k
6
3
The number of combinations available are:20
BUILD SUCCESSFUL (total time: 2 seconds)
```

Output:2

```
enter the value of n and k
8
4
The number of combinations available are:70
BUILD SUCCESSFUL (total time: 5 seconds)
```

4. Write a Java program implement basic queue operations.

```
package java4pgm;
import java.io.*;
import java.util.Scanner;
public class Queue
{
    private static final int max=3;
    int arr[]=new int[max];
    int size=0;
    int front=-1,rear=-1;
    Scanner r=new Scanner(System.in);
    public void insert()
    {
        System.out.println("Enter the element to insert\n");
        int element=r.nextInt();
        if(rear>max-1)
            System.out.println("Overflow");
        else
        {
            rear++;
            arr[rear]=element;
            System.out.println("Element"+element+"is pushed to queue");
        }
    }
    public void delete()
    {
        if(front==rear) System.out.println("Queue underflow\n");
        else
        {
            front++;
            System.out.println("Pop operation is done");
            System.out.println("You removed"+arr[front]+"from the queue");
        }
    }
    public void display()
    {
        if(rear==front)
            System.out.println("Queue is empty");
        else
        {
            System.out.println("Queue elements are\n");
            for(int i=front+1;i<=rear;i++)
```

```
        {
            System.out.println(arr[i]);
        }
    }
}

public static void main(String[] args)
{
    int choice,y=1;
    Queue q=new Queue();
    boolean exit=false;
    while(!exit)
    {
        System.out.println("Queue operations\n");
        System.out.println("1.Insert\n2.Delete\n3.Display\n4.exit");
        System.out.println("Enter your choice\n");
        Scanner input=new Scanner(System.in);
        choice=input.nextInt(); switch(choice)
        {
            case 1:
                System.out.println("Insert element to the queue\n");
                q.insert();
                break;
            case 2:
                System.out.println("Delete element from the queue\n");
                q.delete();
                break;
            case 3:
                q.display();
                break;
            case 4:
                System.out.println("exit");
                System.exit(0);
                break;
        }
    }
}
```

Output:

```
run:
Queue operations
1.Insert
2.Delete
3.Display
4.exit
Enter your choice
1
Insert element to the queue
Enter the element to insert
10
Element10is pushed to queue
Queue operations
1.Insert
2.Delete
3.Display
4.exit
Enter your choice
1
Insert element to the queue
Enter the element to insert
20
Element20is pushed to queue
Queue operations
1.Insert
2.Delete
3.Display
4.exit
Enter your choice
1
Insert element to the queue
Enter the element to insert
30
Element30is pushed to queue
Queue operations
1.Insert
2.Delete
3.Display
4.exit
Enter your choice
3
Queue elements are
```


10

20

30

Queue operations

1.Insert

2.Delete

3.Display

4.exit

Enter your choice

2

Delete element from the queue

Pop operation is done

You removed 10 from the queue

Queue operations

1.Insert

2.Delete

3.Display

4.exit

Enter your choice

3

Queue elements are

20

30

Queue operations

1.Insert

2.Delete

3.Display

4.exit

Enter your choice

4

exit

BUILD SUCCESSFUL (total time: 23 seconds)

5. Write a Java program to count the frequency of words, characters in the given line of text.

```
package java5pgm;
import java.util.*;
class frequency
{
    public static void main(String[] args)
    {
        Scanner scan = new Scanner(System.in);
        String[] rWords; ArrayList<String> nWords = new ArrayList<>();
        ArrayList<Integer> nFreq = new ArrayList<>();
        System.out.println("Enter the line");
        String line = scan.nextLine();
        line = line.replaceAll(" +", " ");
        rWords = line.split(" ");
        for (String current : rWords)
        {
            int r = 0;
            for (String rWord : rWords)
            {
                if (current.equals(rWord))
                {
                    r++;
                }
            }
            if(!nWords.contains(current))
            {
                nWords.add(current);
                nFreq.add(r);
            }
        }
        for(int i = 0; i < nWords.size(); i++)
        {
            System.out.println(nWords.get(i) + " : Freq = " + nFreq.get(i));
        }
        System.out.println("Total words: " + nWords.size());
        System.out.println("Total Characters: " + line.length());
    }
}
```

Output:1

```
run:
Enter the line
vijay kumar
vijay : Freq = 1
kumar : Freq = 1
Total words: 2
Total Characters: 11
BUILD SUCCESSFUL (total time: 5 seconds)
```

Output:2

```
run:
Enter the line
RCUB MCA
RCUB : Freq = 1
MCA : Freq = 1
Total words: 2
Total Characters: 8
BUILD SUCCESSFUL (total time: 13 seconds)
```

6. Write a Java program that creates an object and initializes its data members using constructor. Use constructor overloading concept.

```
package java6pgm;
public class Constructor
{
    double width, height, depth;
    Constructor()
    {
        width = height = depth = 0;
    }
    Constructor(double len)
    {
        width = height = depth = len;
    }
    Constructor(double w, double h, double d)
    {
        width = w;
        height = h;
        depth = d;
    }
    void volume()
    {
        double vol;
        vol= width * height * depth;
        System.out.println(" Volume of mybox is " + vol);
    }
    public static void main(String[] args)
    {
        Constructor a = new Constructor();
        Constructor b = new Constructor(7);
        Constructor c = new Constructor(10, 20, 15);
        a.volume();
        b.volume();
        c.volume();
    }
}
```

Output:

run:

Volume of mybox is 0.0

Volume of mybox is 343.0

Volume of mybox is 3000.0

BUILD SUCCESSFUL (total time: 0 seconds)

7. Write a Java Program to implement inheritance and demonstrate use of method overriding(example: Bank account/Employee.)

package inheritance;

```
public class employee1
{
    protected int emp_no,salary;
    protected String name;
    public employee1(int empno,String nam,int sal)
    {
        emp_no=empno;
        name=nam;
        salary=sal;
    }
    public void empdata()
    {
        System.out.println("Employee no="+emp_no);
        System.out.println("Name="+name);
        System.out.println("Salary="+salary);
    }
}
```

package inheritance;

```
public class manager extends employee1
{
    int reward;
    public manager(int empno,String nam,int sal,int p)
    {
        super(empno,nam,sal);
        reward=p;
    }
    public void managerdata()
    {
        System.out.println("Employee no="+emp_no);
        System.out.println("Name="+name);
        System.out.println("Salary="+salary);
        System.out.println("Rewards="+reward);
    }
}
```

package inheritance;

```
public class Scientist extends employee1
{
    int perks;
    public Scientist (int empno,String nam,int sal,int s)
```

```
{
    Super(empno,nam,sal);
    perks=s;
}
public void Scientistdata()
{
    System.out.println("Employee no="+emp_no);
    System.out.println("Name="+name);
    System.out.println("Salary="+salary);
    System.out.println("Rewards="+perks);
}
}
```

package inheritance;
import java.lang.*;
import java.util.Scanner;
public class Inheritance
{
 public static void main(String[] args)
 {
 employee1 emp=new employee1(1,"Vijay",30000);
 emp.empdata();
 manager mg=new manager(2,"Kumar",40000,2000);
 mg.managerdata();
 Scientist s=new Scientist(3,"Vijaykumar",50000,350000);
 s.Scientistdata();
 }
}

Output:

run:

```
Employee no=1
Name=Vijay
Salary=30000
Employee no=2
Name=Kumar
Salary=40000
Rewards=2000
Employee no=3
Name=Vijaykumar
Salary=50000
Rewards=350000
BUILD SUCCESSFUL (total time: 0 seconds)
```

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.

First_package.java

```
package first_package;
import java.util.Scanner;
public class First_package
{
    public static void main(String[] args)
    {
        Scanner in=new Scanner(System.in);
        evenodd o=new evenodd();
        System.out.println("Enter the number:");
        int num=in.nextInt();
        o.displayevenodd(num);
    }
}
```

evenodd.java

```
package first_package;
public class evenodd
{
    public void displayevenodd(int n)
    {
        if(n%2==0) System.out.println(n+" Even number");
        else System.out.println(n+" Odd number");
    }
}
```

secondpackage .java

```
package Second_package;
import java.util.Scanner;
import first_package.evenodd;
public class secondpackage
{
    public static void main(String[] args)
    {
        Scanner in=new Scanner(System.in);
        evenodd o=new evenodd();
        System.out.println("Enter the number:");
        int num=in.nextInt(); o.displayevenodd(num);
    }
}
```

Output:01

run:

Enter the number:

20

20 Even number

BUILD SUCCESSFUL (total time: 2 seconds)

Output:02

run:

Enter the number:

25

25 Odd number

BUILD SUCCESSFUL (total time: 1 second)

9. Write a program to demonstrate use of interfaces for two different classes. Interface should also include constants along with function prototypes.

test1.java

```
package vijay;
public interface test1
{
    final int n=3,x=5;
    abstract public void fact();
    abstract public void fib();
}
```

Factorial.java

```
package vijay;
abstract class Factorial implements test1
{
    public void fact()
    {
        int i,f=1;
        for( i=1;i<=n;i++)
        {
            f=f*i;
        }
        System.out.println("The factorial of " +n+ " is:"+f);
    }
}
```

Vijay.java

```
package vijay;
public class Vijay extends Factorial implements test1
{
    public void fib()
    {
        int i,f1=0,f2=1,f3=0;
        System.out.println("The fibonacci values of "+x+" are:");
        for(i=0;i<x;i++)
        {
            System.out.println(f3);
            f1=f2; f2=f3; f3=f1+f2;
        }
    }
}
```

```
public static void main(String[] args)
{
    Vijay obj=new Vijay();
    obj.fact();
    obj.fib();
}
}
```

Output:

run:

The factorial of 3 is:6

The fibonacci values of 5 are:

0

1

1

2

3

BUILD SUCCESSFUL (total time: 1 second)

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.

```
package exception_handling;
import java.util.Scanner;
public class Exception
{
    public static void main(String argv[])
    {
        int num1,num2,result;
        Scanner Sc = new Scanner(System.in);
        System.out.print("Enter two numbers : ");
        num1 = Sc.nextInt();
        num2 = Sc.nextInt();
        int arr[] = new int[2];
        try
        {
            result = num1 / num2;
            System.out.println("Result of Division : " + result);
            arr[0] = 0;
            arr[1] = 1;
            arr[2] = 2;
            arr[3] = 3;
            arr[4] = 4;
        }
        catch (ArithmeticException e)
        {
            System.out.println("Err: Divided by Zero");
        }
        catch (ArrayIndexOutOfBoundsException e)
        {
            System.out.println("Err: Array Out of Bound");
        }
        finally
        {
            System.out.println("This is finally block");
        }
    }
}
```

Output:01

run:

Enter two numbers : 15

7

Result of Division : 2

Err: Array Out of Bound

This is finally block

BUILD SUCCESSFUL (total time: 8 seconds)

Output:02

run:

Enter two numbers : 10

0

Err: Divided by Zero

This is finally block

BUILD SUCCESSFUL (total time: 3 seconds)

11. Write a program to implement the concept of Exception Handling by creating user defined exceptions.

```
package userdefinedexception;
import java.util.Scanner;
public class UserDefinedException
{
    public static void main(String[] args)
    {
        int num;
        Scanner Sc = new Scanner(System.in);
        System.out.print("Enter any number : ");
        num = Sc.nextInt();
        try
        {
            if(num%2 == 0)
                System.out.print( num + " is an even number");
            else
                throw(new OddNumberException());
        }
        catch(OddNumberException Ex)
        {
            System.out.print("Error : " + Ex.getMessage());
        }
        System.out.print("\nEnd of program\n");
    }
}

class OddNumberException extends Exception
{
    OddNumberException()
    {
        super("Odd number exception");
    }
}
```

Output:01

Enter any number : 4
4 is an even number
End of program

Output:02

Enter any number : 5
Error : Odd number exception
End of program
BUILD SUCCESSFUL (total time: 1 second)

12. Illustrate creation of thread by extending Thread class/ implementing runnable interface .

```
package thread;
public class Thread_Interface
{
    public static void m1()
    {
        System.out.println("Hello Visitors");
    }
}
class Test extends Thread_Interface implements Runnable
{
    public void run()
    {
        System.out.println("Run method executed by child Thread");
    }
    public static void main(String[] args)
    {
        Test t = new Test();
        t.m1();
        Thread t1 = new Thread(t);
        t1.start();
        System.out.println("Main method executed by main thread");
    }
}
```

Output:

```
run:
Hello Visitors
Main method executed by main thread
Run method executed by child Thread
BUILD SUCCESSFUL (total time: 0 seconds)
```

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.

```
package three_thread;
class Child implements Runnable
{
    Thread t;
    Child(String name)
    {
        t = new Thread(this, name);
        t.start();
    }
    public void run()
    {
        for(int i=1;i<=5;i++)
        {
            try
            {
                if(t.getName().equals("First Thread"))
                {
                    Thread.sleep(1000);
                    System.out.println(t.getName()+": GoodMorning");
                }
                else if(t.getName().equals("Second Thread"))
                {
                    Thread.sleep(3000);
                    System.out.println(t.getName()+": Hello");
                }
                else
                {
                    Thread.sleep(5000);
                    System.out.println(t.getName()+": Welcome");
                }
            }
            catch(InterruptedException e)
            {
                System.out.println(t.getName()+" is interrupted");
            }
        }
    }
}
```



```
class ThreeThreads
{
    public static void main(String args[])
    {
        Child one = new Child("First Thread");
        Child two = new Child("Second Thread");
        Child three = new Child("Third Thread");
    }
}
```

Output:01

run:

First Thread: Good Morning
First Thread: Good Morning
Second Thread: Hello
First Thread: Good Morning
First Thread: Good Morning
Third Thread: Welcome
First Thread: Good Morning
Second Thread: Hello
Second Thread: Hello
Third Thread: Welcome
Second Thread: Hello
Third Thread: Welcome
Second Thread: Hello
Third Thread: Welcome
Third Thread: Welcome
BUILD SUCCESSFUL (total time: 25 seconds)

14. Illustrate thread join concept.

```
package join;
public class Join
{
    public static void main(String[] args)
    {
        Thread th1 = new Thread(new MyClass(), "th1");
        Thread th2 = new Thread(new MyClass(), "th2");
        Thread th3 = new Thread(new MyClass(), "th3");
        th1.start();
        try
        {
            th1.join();
        }
        catch (InterruptedException ie)
        {
            ie.printStackTrace();
        }
        th2.start();
        try
        {
            th2.join();
        }
        catch (InterruptedException ie)
        {
            ie.printStackTrace();
        }
        th3.start();
        try
        {
            th3.join();
        }
        catch (InterruptedException ie)
        {
            ie.printStackTrace();
        }
        System.out.println("All three threads have finished execution");
    }
}

class MyClass implements Runnable
{
    @Override
    public void run()
```

```
{
    Thread t = Thread.currentThread();
    System.out.println("Thread started: " + t.getName());
    try
    {
        Thread.sleep(4000);
    }
    catch (InterruptedException ie)
    {
        ie.printStackTrace();
    }
    System.out.println("Thread ended: " + t.getName());
}
}
```

Output:

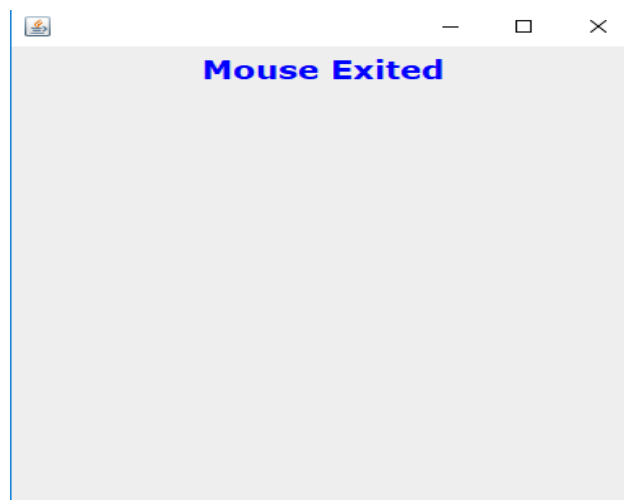
run:
Thread started: th1
Thread ended: th1
Thread started: th2
Thread ended: th2
Thread started: th3
Thread ended: th3
All three threads have finished execution
BUILD SUCCESSFUL (total time: 12 seconds)

15. Write a java program to implement mouse events like mouse pressed, mouse released and mouse moved by means of adapter classes.

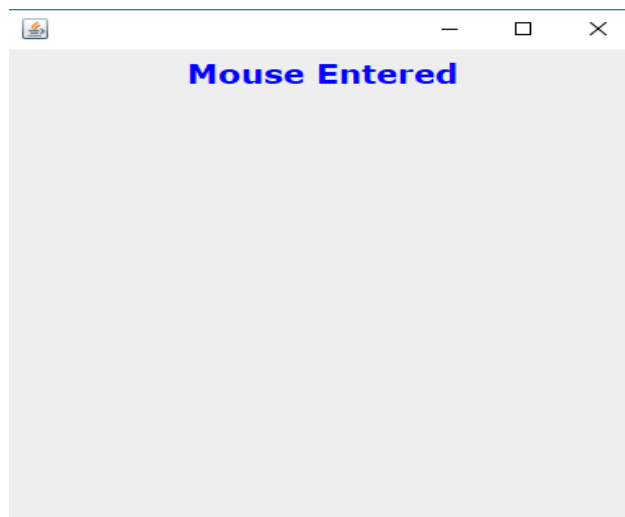
```
package mouse;
import javax.swing.*;
import java.awt.*;
import javax.swing.event.*;
import java.awt.event.*;
class A extends JFrame implements MouseListener
{
    JLabel l1;
    public A()
    {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(400, 400);
        setLayout(new FlowLayout());
        l1 = new JLabel();
        Font f = new Font("Verdana", Font.BOLD, 20);
        l1.setFont(f); l1.setForeground(Color.BLUE);
        l1.setAlignmentX(Component.CENTER_ALIGNMENT);
        l1.setAlignmentY(Component.CENTER_ALIGNMENT);
        add(l1);
        addMouseListener(this);
        setVisible(true);
    }
    public void mouseExited(MouseEvent m)
    {
        l1.setText("Mouse Exited");
    }
    public void mouseEntered(MouseEvent m)
    {
        l1.setText("Mouse Entered");
    }
    public void mouseReleased(MouseEvent m)
    {
        l1.setText("Mouse Released");
    }
    public void mousePressed(MouseEvent m)
    {
        l1.setText("Mouse Pressed");
    }
    public void mouseClicked(MouseEvent m)
    {
        l1.setText("Mouse Clicked");
    }
}
```

```
    }  
}  
  
package mouse;  
public class Mouse  
{  
    public static void main(String[] args)  
    {  
        A a = new A();  
    }  
}
```

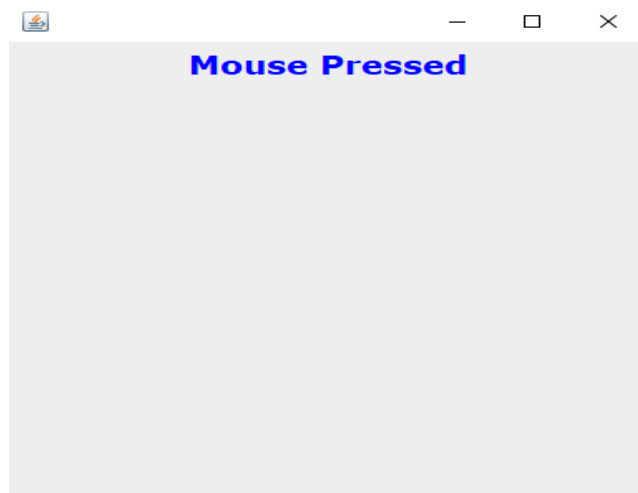
Output:01



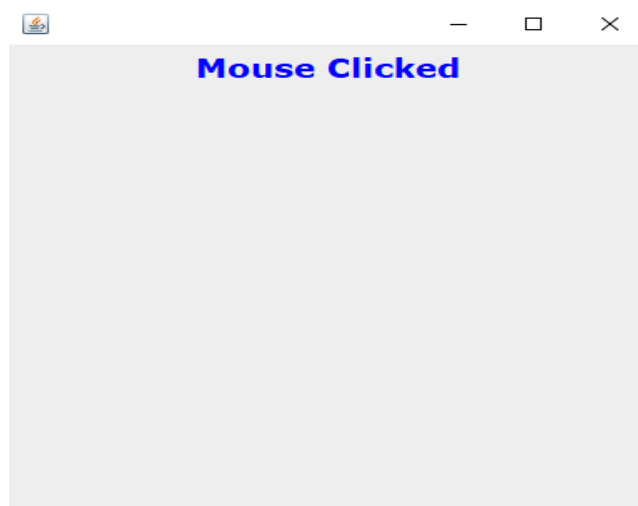
Output:02



Output:03



Output:04



Output:05



16. Write a java program that creates a user interface to perform integer divisions. The user enters two numbers in the textfields, Num1 and

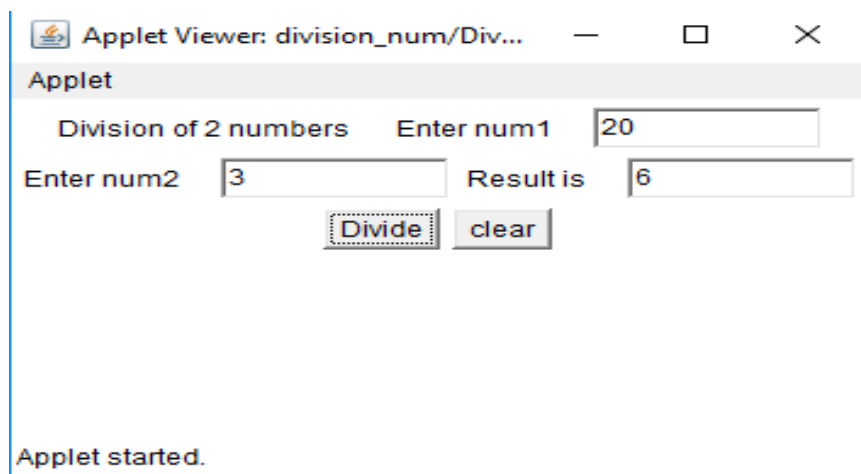
Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an ArithmeticException Display the exception in a message dialog box.

```
package division_num;
import java.awt.*;
import javax.swing.*;
import java.applet.Applet;
import java.awt.event.*;
/*<applet code="Division" width=234 height=456> </applet> */
public class Division extends Applet implements ActionListener
{
    TextField t1,t2,t3;
    Button b,clear;
    Label L1,L2,L3,L4;
    String s; Division e;
    public void init()
    {
        e=this;
        t1=new TextField(10);
        t2=new TextField(10);
        t3=new TextField(10);
        L1=new Label("Enter num1");
        L2=new Label("Enter num2");
        L3=new Label("Result is");
        L4=new Label("Division of 2 numbers");
        b=new Button("Divide");
        clear=new Button("clear");
        add(L4);
        add(L1);
        add(t1);
        add(L2);
        add(t2);
        add(L3);
        add(t3);
        add(b);
        add(clear);
        b.addActionListener(this);
        clear.addActionListener(this);
    }
    public void actionPerformed(ActionEvent ae)
    {

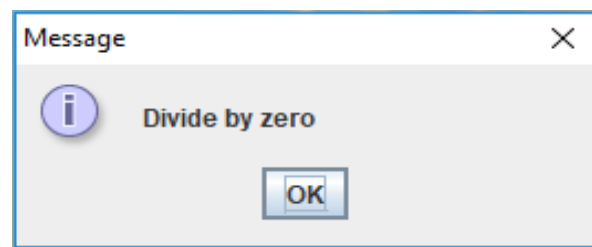
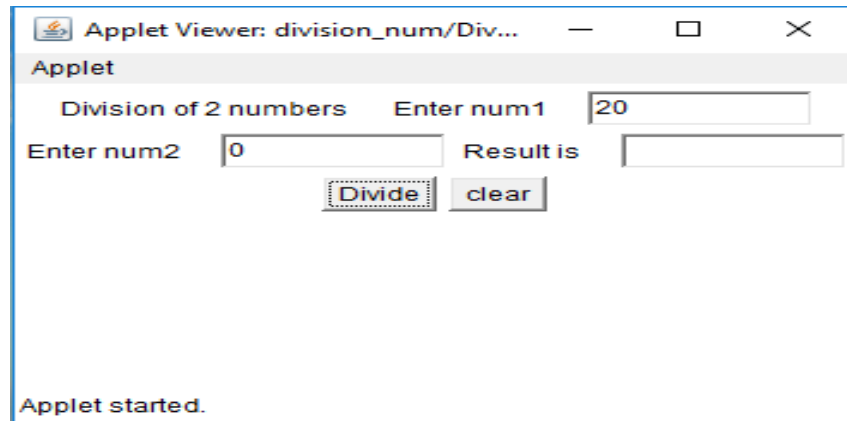
```

```
try
{
    int num1=Integer.parseInt(t1.getText());
    int num2=Integer.parseInt(t2.getText());
    s=""+(num1/num2);
    t3.setText(s);
}
catch(ArithmeticException a)
{
    JOptionPane.showMessageDialog(null,"Divide by zero");
}
catch(NumberFormatException b)
{
    JOptionPane.showMessageDialog(null,"NumberFormateException");
}
if(ae.getSource()==clear)
{
    t1.setText("");
    t2.setText("");
    t3.setText("");
}
}
```

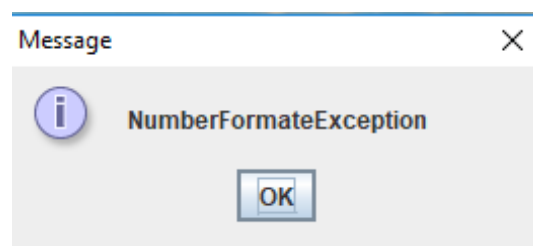
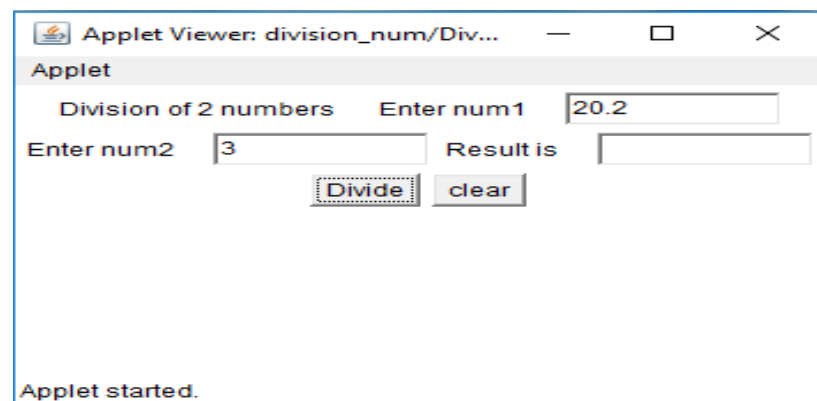
Output:01



Output:02



Output:03

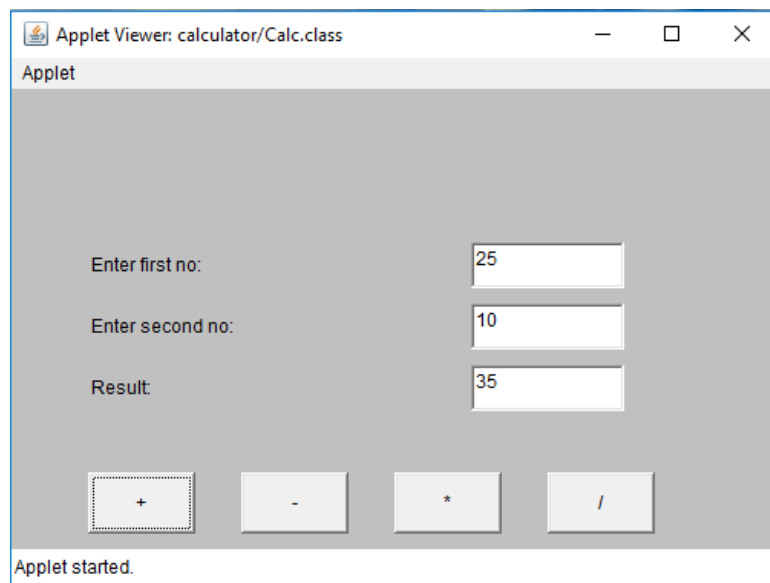


17. Write a Java program to illustrate basic calculator using grid layout manager.

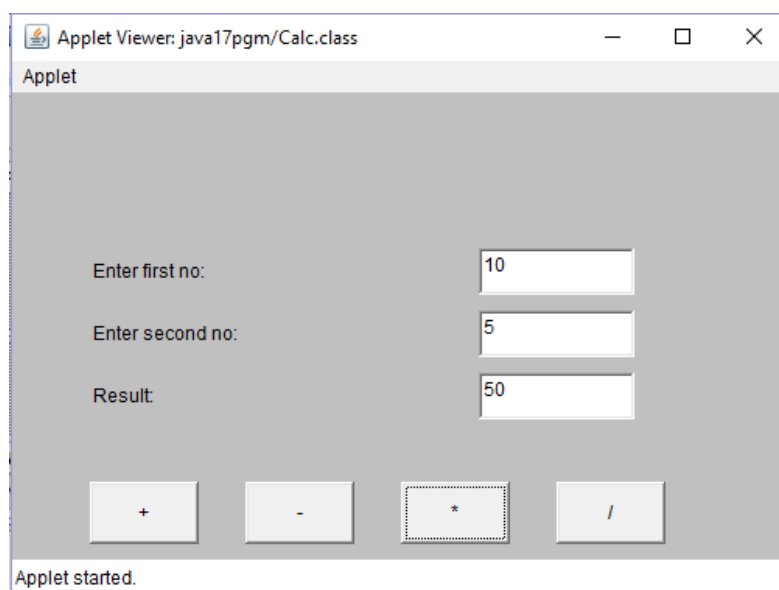
```
package calculator;
import java.awt.*;
import java.applet.*;
import java.awt.event.*;
import java.awt.Color;
public class Calc extends Applet implements ActionListener
{
    Label l1,l2,l3;
    TextField t1,t2,t3;
    Button add,sub,mul,div;
    public void init()
    {
        setBackground(Color.lightGray);
        setLayout(null);
        l1=new Label("Enter first no:");
        add(l1);
        l2=new Label("Enter second no:");
        add(l2);
        l3=new Label("Result:");
        add(l3);
        t1=new TextField(10);
        add(t1);
        t2=new TextField(10);
        add(t2);
        t3=new TextField(10);
        add(t3);
        add=new Button("+");
        add(add);
        add.addActionListener(this);
        sub=new Button("-");
        add(sub);
        sub.addActionListener(this);
        mul=new Button("*");
        add(mul);
        mul.addActionListener(this);
        div=new Button("/");
        add(div);
        div.addActionListener(this);
        setSize(500,300);
        l1.setBounds(50,100,100,30);
        l2.setBounds(50,140,100,30);
        l3.setBounds(50,180,100,30);
        t1.setBounds(300,100,100,30);
        t2.setBounds(300,140,100,30);
        t3.setBounds(300,180,100,30);
        add.setBounds(50,250,70,40);
        sub.setBounds(150,250,70,40);
        mul.setBounds(250,250,70,40);
        div.setBounds(350,250,70,40);
    }
}
```

```
public void actionPerformed(ActionEvent ae)
{
    int x1=Integer.parseInt(t1.getText());
    int x2=Integer.parseInt(t2.getText());
    if(ae.getSource()==add)
    {
        t3.setText(String.valueOf(x1+x2));
    }
    if(ae.getSource()==sub)
    {
        t3.setText(String.valueOf(x1-x2));
    }
    if(ae.getSource()==mul)
    {
        t3.setText(String.valueOf(x1*x2));
    }
    if(ae.getSource()==div)
    {
        t3.setText(String.valueOf(x1/x2));
    }
}
}
```

Output:1



Output:2

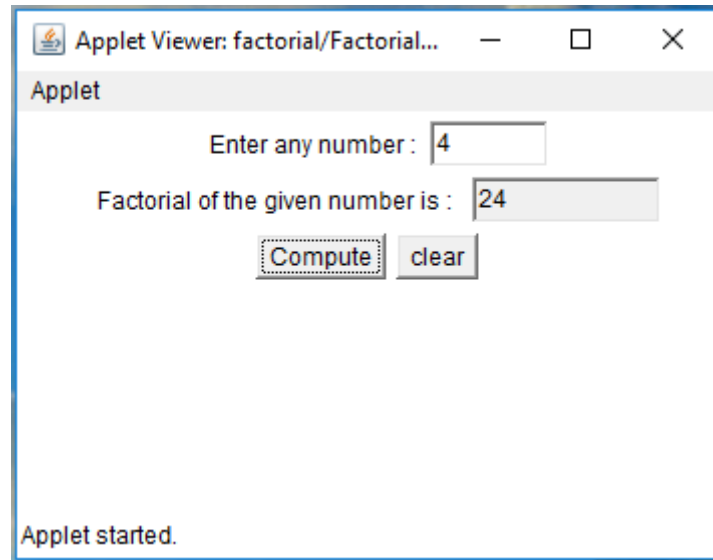


18. Develop an applet that receives an integer in one text field, and computes its factorial Value and returns it in another text field, when the button named “Compute” is clicked.

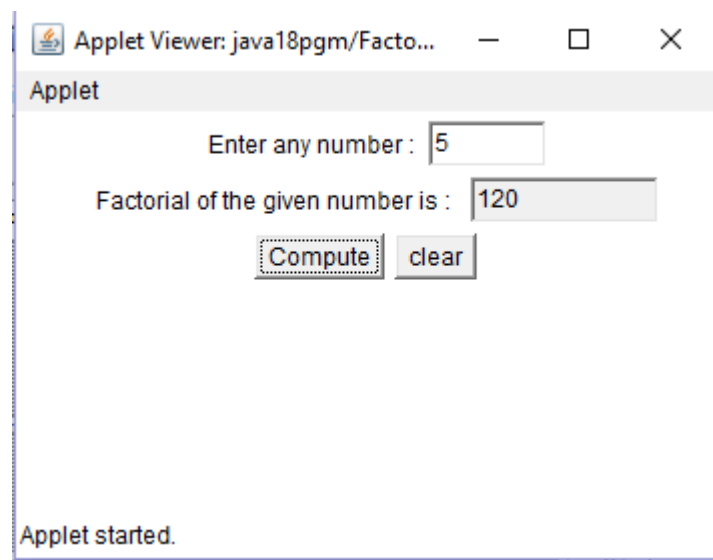
```
package factorial;
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
public class Factorial extends Applet implements ActionListener
{
    TextField input,output;
    Button compute,clear;
    int fact=0;
    public void init()
    {
        compute=new Button("Compute");
       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

```
}
```

Output:1



Output:2



19. Write a java program to create student report using applet, read the input using text boxes and display the o/p using buttons.

```
package student_report;
import java.awt.*;
import java.applet.*;
import java.awt.event.*;
public class Student_report extends Applet implements ActionListener
{
    Label Title,RegNo,Name,Java,SE,UNIX,WEB;
    TextField txtRegNo,txtName,txtJava,txtSE,txtUNIX,txtWEB;
    Button cmdReport;
    int total;
    float avg;
    public void init()
    {
        setLayout(null);
        Title=new Label("Enter Student Details");
        RegNo=new Label("RegNo:");
        Name=new Label("Name:");
        Java=new Label("Java:");
        SE=new Label("SE:");
        UNIX=new Label("UNIX:");
        WEB=new Label("WEB:");
        txtRegNo=new TextField(10);
        txtName=new TextField(25);
        txtJava=new TextField(3);
        txtSE=new TextField(3);
        txtUNIX=new TextField(3);
        txtWEB=new TextField(3);
        cmdReport=new Button("View Student Result");
        Title.setBounds(100,0,200,20);
        RegNo.setBounds(0,50,100,20);
        txtRegNo.setBounds(120,50,100,20);
        Name.setBounds(0,75,100,20);
        txtName.setBounds(120,75,250,20);
        Java.setBounds(0,100,100,20);
        txtJava.setBounds(120,100,40,20);
        SE.setBounds(0,125,100,20);
        txtSE.setBounds(120,125,40,20);
        UNIX.setBounds(0,150,100,20);
        txtUNIX.setBounds(120,150,40,20);
        WEB.setBounds(0,175,100,20);
        txtWEB.setBounds(120,175,40,20);
        cmdReport.setBounds(100,225,150,30);
        add(Title);
        add(RegNo);add(txtRegNo);
        add(Name);add(txtName);
        add(Java);add(txtJava);
        add(SE);add(txtSE);
        add(UNIX);add(txtUNIX);
```

```
        add(WEB);add(txtWEB);
        add(cmdReport);
        cmdReport.addActionListener(this);
    }
    public void actionPerformed(ActionEvent ae)
    {
        try
        {
            int java=Integer.parseInt(txtJava.getText());
            int se=Integer.parseInt(txtSE.getText());
            int unix=Integer.parseInt(txtUNIX.getText());
            int web=Integer.parseInt(txtWEB.getText());
            total=(java+se+unix+ web);
            avg=total/4;
        }
        catch(NumberFormatException e)
        {

        }
        repaint();
    }
    public void paint(Graphics g)
    {
        g.drawString("STUDENT REPORT",100,275);
        g.drawString("Reg. No.: "+txtRegNo.getText(),0,300);
        g.drawString("Name : "+txtName.getText(),0,325);
        g.drawString("Java: "+txtJava.getText(),0,350);
        g.drawString("SE : "+txtSE.getText(),0,375);
        g.drawString("UNIX : "+txtUNIX.getText(),0,400);
        g.drawString("WEB: "+txtWEB.getText(),0,425);
        g.drawString("Total: "+total,0,475);
        g.drawString("Average: "+avg,0,500);
    }
}
```


Output:

Applet Viewer: java19pgm/Student_report.class

Applet

Enter Student Details

RegNo:	<input type="text" value="m100"/>
Name:	<input type="text" value="VIJAYKUMAR"/>
OS:	<input type="text" value="88"/>
JAVA:	<input type="text" value="85"/>
DBMS:	<input type="text" value="86"/>
CN:	<input type="text" value="82"/>

STUDENT REPORT

Reg. No.: m100
Name : VIJAYKUMAR
Java: 88
SE : 85
UNIX : 86
WEB: 82

Total: 341
Average: 85.0

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

```
package tictac;
import java.util.Scanner;
public class Tictac
{
    public static final int EMPTY = 0;
    public static final int CROSS = 1;
    public static final int NOUGHT = 2;
    public static final int PLAYING = 0;
    public static final int DRAW = 1;
    public static final int CROSS_WON = 2;
    public static final int NOUGHT_WON = 3;
    public static final int ROWS = 3, COLS = 3;
    public static int[][] board = new int[ROWS][COLS];
    public static int currentState;
    public static int currentPlayer;
    public static int currntRow, currentCol;
    public static Scanner in = new Scanner(System.in);
    public static void main(String[] args)
    {
        initGame();
        do
        {
            playerMove(currentPlayer);
            updateGame(currentPlayer, currntRow, currentCol);
            printBoard();
            if (currentState == CROSS_WON)
            {
                System.out.println("'X' won! Bye!");
            }
            else if (currentState == NOUGHT_WON)
            {
                System.out.println("'O' won! Bye!");
            }
            else if (currentState == DRAW)
            {
                System.out.println("It's a Draw! Bye!");
            }
            currentPlayer = (currentPlayer == CROSS) ? NOUGHT : CROSS;
        }
        while (currentState == PLAYING);
    }
    public static void initGame()
    {
        for (int row = 0; row < ROWS; ++row)
        {
```

```
        for (int col = 0; col < COLS; ++col)
        {
            board[row][col] = EMPTY;
        }
    }
    currentState = PLAYING;
    currentPlayer = CROSS;
}
public static void playerMove(int theSeed)
{
    boolean validInput = false;
    do
    {
        if (theSeed == CROSS)
        {
            System.out.print("Player 'X', enter your move (row[1-3] column[1-3]): ");
        }
        else
        {
            System.out.print("Player 'O', enter your move (row[1-3] column[1-3]): ");
        }
        int row = in.nextInt() - 1;
        int col = in.nextInt() - 1;
        if (row >= 0 && row < ROWS && col >= 0 && col < COLS && board[row][col] == EMPTY)
        {
            currntRow = row;
            currentCol = col;
            board[currntRow][currentCol] = theSeed;
            validInput = true;
        }
        else
        {
            System.out.println("This move at (" + (row + 1) + "," + (col + 1) + ") is not valid. Try again...");
        }
    }
    while (!validInput);
}
public static void updateGame(int theSeed, int currentRow, int currentCol)
{
    if (hasWon(theSeed, currentRow, currentCol))
    {
        currentState = (theSeed == CROSS)?CROSS_WON : NOUGHT_WON;
    }
    else if (isDraw())
    {
        currentState = DRAW;
    }
}
```

```
    }
}
public static boolean isDraw()
{
    for (int row = 0; row < ROWS; ++row)
    {
        for (int col = 0; col < COLS; ++col)
        {
            if (board[row][col] == EMPTY)
            {
                return false;
            }
        }
    }
    return true;
}
public static boolean hasWon(int theSeed, int currentRow, int currentCol)
{
    return (board[currentRow][0] == theSeed
    && board[currentRow][1] == theSeed
    && board[currentRow][2] == theSeed
    || board[0][currentCol] == theSeed
    && board[1][currentCol] == theSeed
    && board[2][currentCol] == theSeed
    || currentRow == currentCol
    && board[0][0] == theSeed
    && board[1][1] == theSeed
    && board[2][2] == theSeed
    || currentRow + currentCol == 2
    && board[0][2] == theSeed
    && board[1][1] == theSeed
    && board[2][0] == theSeed);
}
public static void printBoard()
{
    for (int row = 0; row < ROWS; ++row)
    {
        for (int col = 0; col < COLS; ++col)
        {
            printCell(board[row][col]);
            if (col != COLS - 1)
            {
                System.out.print("|");
            }
        }
        System.out.println();
        if (row != ROWS - 1)
        {
            System.out.println("-----");
        }
    }
}
```

```
        }
        System.out.println();
    }
    public static void printCell(int content)
    {
        switch (content)
        {
            case EMPTY:
                System.out.print("  ");
                break;
            case NOUGHT:
                System.out.print(" O ");
                break;
            case CROSS:
                System.out.print(" X ");
                break;
        }
    }
}
```

Output:

run:

Player 'X', enter your move (row[1-3] column[1-3]): 3

1

| |

| |

X | |

Player 'O', enter your move (row[1-3] column[1-3]): 1

1

O | |

| |

X | |

Player 'X', enter your move (row[1-3] column[1-3]): 2

2

O | |

| X |

X | |

Player 'O', enter your move (row[1-3] column[1-3]): 1

3

O | | O

| X |

X | |

Player 'X', enter your move (row[1-3] column[1-3]): 1

2

O | X | O

| X |

X | |

Player 'O', enter your move (row[1-3] column[1-3]): 3

2

O | X | O

| X |

X | O |

Player 'X', enter your move (row[1-3] column[1-3]): 2

1

O | X | O

X | X |

X | O |

Player 'O', enter your move (row[1-3] column[1-3]): 2

3

O | X | O

X | X | O

X | O |

Player 'X', enter your move (row[1-3] column[1-3]): 3

3

O | X | O

X | X | O

X | O | X

It's a Draw! Bye!

BUILD SUCCESSFUL (total time: 1 minute 3 seconds)