**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.**

package prime;

import java.util.Scanner;

public class Prime

{

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");

}

}

}

}

**Output1:**

Enter a number

5

Given number is prime

BUILD SUCCESSFUL (total time: 2 seconds)

**Output2:**

Enter a number

6

Given number is not prime

BUILD SUCCESSFUL (total time: 1 second)

**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).**

package lcm\_gcd;

import java.util.Scanner;

public class LCM\_GCD {

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;

}

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:**

run:

Enter the two numbers:

5

36

The GCD of two numbers is:1

The LCM of two numbers is:180

BUILD SUCCESSFUL (total time: 3 seconds)

**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).**

package fact;

import java.util.\*;

public class Fact

{

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.println(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=fact\*i;

}

return fact;

}

}

**OUTPUT:**

Enter the value for n

5

Enter the value for k

3

5C3=

10

BUILD SUCCESSFUL (total time: 4 seconds)

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

package queue;

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.Enqueue");

System.out.println("2.Dqueue");

System.out.println("3.Display");

System.out.println("4.Exit");

System.out.println("Enter your option:");

option=input.nextInt();

switch(option)

{

case 1: System.out.println("Enter the value");

int val=input.nextInt();

Enqueue(val);

break;

case 2:Dqueue();

break;

case 3:Display();

break;

case 4:System.exit(0);

break;

default:System.out.println("Invalid option");

break;

}

}

}

public static void Enqueue(int val)

{

if(rear>=MAX)

{

System.out.println("Queue is full");

System.out.println(" ");

return;

}

queue[rear]=val;

rear++;

}

public static void Dqueue()

{

System.out.println(" ");

if(rear<=0)

{

System.out.println("Queue is empty");

System.out.println(" ");

return;

}

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

for(int i=front;i<rear-1;i++)

{

int temp=queue[i+1];

queue[i]=temp;

}

rear--;

}

public static void isplay()

{

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:**

run:

1.Enqueue

2.Dqueue

3.Display

4.Exit

Enter your option:

1

Enter the value

2

1.Enqueue

2.Dqueue

3.Display

4.Exit

Enter your option:

1

Enter the value

3

1.Enqueue

2.Dqueue

3.Display

4.Exit

Enter your option:

3

Element in the queue are

2

3

1.Enqueue

2.Dqueue

3.Display

4.Exit

Enter your option:

2

Delted:2

1.Enqueue

2.Dqueue

3.Display

4.Exit

Enter your option:

2

Delted:3

1.Enqueue

2.Dqueue

3.Display

4.Exit

Enter your option:

3

queue is empty

1.Enqueue

2.Dqueue

3.Display

4.Exit

Enter your option:

4

BUILD SUCCESSFUL (total time: 31 seconds)

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

package freq;

import java.util.\*;

public class freq

{

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:**

run:

Enter the line

i love to be what i am

i : Freq = 2

love : Freq = 1

to : Freq = 1

be : Freq = 1

what : Freq = 1

am : Freq = 1

Total Characters: 22

BUILD SUCCESSFUL (total time: 38 seconds)

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

package student;

public class STUDENT

{

int id;

String name;

int age;

STUDENT(int i,String n)

{

id=i;

name=n;

}

STUDENT(int i,String n,int a)

{

id=i;

name=n;

age=a;

}

void display()

{

System.out.println(id+" "+name+" "+age);

}

public static void main(String[] args)

{

STUDENT s1=new STUDENT(111,"karan");

STUDENT s2=new STUDENT(222,"aryan",25);

s1.display();

s2.display();

}

}

**OUTPUT:**

run:

111 karan 0

222 aryan 25

BUILD SUCCESSFUL (total time: 0 seconds)

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

package inheritance;

public class employee

{

protected int emp\_no,salary;

protected String name;

public employee(int empno,String nam,int sal)

{

emp\_no=empno;

name=nam;

salary=sal;

}

public void emp\_data()

{

System.out.println("Employee number="+emp\_no);

System.out.println("Employee name="+name);

System.out.println("Employee salary="+salary);

}

}

package inheritance;

public class manager extends employee

{

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 number="+emp\_no);

System.out.println("Employee name="+name);

System.out.println("Employee salary="+salary);

System.out.println("Reward="+reward);

}

}

package inheritance;

public class scientist extends employee

{

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 number="+emp\_no);

System.out.println("Employee name="+name);

System.out.println("Employee salary="+salary);

System.out.println("perks="+perks);

}

}

package inheritance;

import java.lang.\*;

import java.util.Scanner;

public class Inheritance

{

public static void main(String[] args)

{

employee emp=new employee(1,"Varsha",20000);

emp.emp\_data();

manager man=new manager(2,"Ganga",50000,1000);

man.managerdata();

scientist scient=new scientist(3,"Pragati",60000,50000);

scient.scientistdata();

}

}

**OUTPUT:**

run:

Employee number=1

Employee name=Varsha

Employee salary=20000

Employee number=2

Employee name=Ganga

Employee salary=50000

Reward=1000

Employee number=3

Employee name=Pragati

Employee salary=60000

perks=50000

BUILD SUCCESSFUL (total time: 1 second)

**Program No 08: 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.**

**firstPackage.java**

package firstpackage;

import java.util.Scanner;

public class Firstpackage

{

public static void main(String[] args)

{

Scanner in=new Scanner(System.in);

evenodd o=new evenodd();

System.out.println("Enter a number:");

int num=in.nextInt();

o.displayevenodd(num);

}

}

**evenodd.java**

package firstpackage;

public class evenodd

{

public void displayevenodd(int n)

{

if(n%2==0)

System.out.println(n+" is even number");

else

System.out.println(n+" is odd number");

}

}

**secondpackage.java**

package secondpackage;

import java.util.Scanner;

import firstpackage.evenodd;

public class secondpackage

{

public static void main(String args[])

{

Scanner in=new Scanner(System.in);

evenodd o=new evenodd();

System.out.println("Enter a number");

int num=in.nextInt();

o.displayevenodd(num);

}

}

**OUTPUT1:**

run:

Enter a number:

5

5 is odd number

BUILD SUCCESSFUL (total time: 2 seconds)

**OUTPUT2:**

run:

Enter a number:

6

6 is even number

BUILD SUCCESSFUL (total time: 4 seconds)

**Program No 09: Write a program to demonstrate use of interfaces for two different classes. Interface should also include constants along with function prototypes.**

**varsha25.java**

package varsha25;

public class varsha25 extends factorial25 impliments test25

{

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)

{

varsha25 obj=new varsha25();

obj.fact();

obj.fib();

}

}

**test25.java**

package varsha25;

public interface test25

{

final int n=3,x=5;

abstract public void fact();

abstract public void fib();

}

**factorial25.java**

package varsha25;

abstract class factorial25 implements test25

{

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);

}

}

**OUTPUT:**

run:

the factorial of 3 is:6

the fibonacci values of5are:

0

1

1

2

3

BUILD SUCCESSFUL (total time: 1 second)

**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.**

package exception;

import java.util.Scanner;

public class Exception

{

public static void main(String[] args)

{

int a,b,result;

Scanner input=new Scanner(System.in);

System.out.println("Input two integers");

a=input.nextInt();

b=input.nextInt();

int arr[]=new int[2];

try

{

result=a/b;

System.out.println("Result="+result);

arr[0]=0;

arr[1]=1;

arr[2]=2;

}

catch(ArithmeticException e)

{

System.out.println(e);

}

catch (ArrayIndexOutOfBoundsException e)

{

System.out.println("Exception caught;Divided by zero exception");

}

finally

{

System.out.println("This is final bolck");

}

}

}

**OUTPUT 1:**

run:

Input two integers

8

0

java.lang.ArithmeticException: / by zero

This is final bolck

BUILD SUCCESSFUL (total time: 6 seconds)

**OUTPUT 2:**

run:

Input two integers

4

2

Result=2

Exception caught;Divided by zero exception

This is final bolck

BUILD SUCCESSFUL (total time: 3 seconds)

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

package userdifined;

import java.util.Scanner;

class InvalidAgeException extends Exception

{

InvalidAgeException(String s)

{

super(s);

}

}

public class Userdifined

{

static void validate(int age)throws InvalidAgeException

{

if(age<18)

throw new InvalidAgeException("not valid");

else

System.out.println("Welcome to vote");

}

public static void main(String[] args)

{

Scanner s=new Scanner(System.in);

try

{

validate(20);

}

catch (InvalidAgeException m)

{

System.out.println("Exception occured:"+m);

}

System.out.println("rest of the code----");

}

}

**OUTPUT1:**

run:

Welcome to vote

rest of the code----

BUILD SUCCESSFUL (total time: 0 seconds)

**OUTPUT2:**

When validate is<18

run:

Exception occured: userdifined.InvalidAgeException: not valid

rest of the code----

BUILD SUCCESSFUL (total time: 0 seconds)

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

package pgm12;

class Pgm12 implements Runnable

{

@Override

public void run()

{

System.out.println("Thread is running");

}

public static void main(String[] args)

{

Pgm12 m=new Pgm12();

Thread t=new Thread(m);

t.start();

}

}

**OUTPUT:**

run:

Thread is running

BUILD SUCCESSFUL (total time: 0 seconds)

**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.**

package three;

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()+": Good Morning");

}

else if(t.getName().equals("Second Thread"))

{

Thread.sleep(5000);

System.out.println(t.getName()+": Hello");

}

else

{

Thread.sleep(10000);

System.out.println(t.getName()+": Welcome");

}

}

catch(InterruptedException e)

{

System.out.println(t.getName()+" is interrupted");

}

}

}

}

class Three

{

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:**

run:

First Thread: Good Morning

First Thread: Good Morning

First Thread: Good Morning

First Thread: Good Morning

Second Thread: Hello

First Thread: Good Morning

Second Thread: Hello

Third Thread: Welcome

Second Thread: Hello

Second Thread: Hello

Third Thread: Welcome

Second Thread: Hello

Third Thread: Welcome

Third Thread: Welcome

Third Thread: Welcome

BUILD SUCCESSFUL (total time: 50 seconds)

**Program No 14: Illustrate thread join concept**.

package Lab14;

class Threethreads implements Runnable

{

public void run()

{

try

{

Thread.sleep(500);

} catch(InterruptedException e)

{

System.out.println(e);

} System.out.println("Good morning");

}

}

class Lab14 extends Thread

{

public static void main(String args[])

{

Threethreads m=new Threethreads();

Thread t=new Thread(m);

Thread t1=new Thread(m);

Thread t2=new Thread(m);

t.start();

try

{

t.join(1000);

}

catch(InterruptedException e)

{

System.out.println(e);

}

System.out.println("join");

t1.start();

t2.start();

}

}

**OUTPUT:**

run:

Good morning

join

Good morning

Good morning

BUILD SUCCESSFUL (total time: 1 second)

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

package lab15;

import javax.swing.\*;

import java.awt.\*;

import javax.swing.event.\*;

import java.awt.event.\*;

class AA extends JFrame implements MouseListener

{

JLabel l1;

public AA()

{

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.RED);

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");

}

}

public class Lab15

{

public static void main(String args[])

{

AA a = new AA();

}

}

**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.**

package pgm16;

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

class A extends JFrame implements ActionListener

{

JLabel l1, l2, l3;

JTextField tf1, tf2, tf3;

JButton b1;

A()

{

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLayout(new FlowLayout());

l1 = new JLabel("Welcome");

setSize(800, 400);

l1 = new JLabel("Enter Number1");

add(l1);

tf1 = new JTextField(10);

add(tf1);

l2 = new JLabel("Enter Number2");

add(l2);

tf2 = new JTextField(10);

add(tf2);

l3 = new JLabel("Result");

add(l3);

tf3 = new JTextField(10);

add(tf3);

b1 = new JButton("Divide");

add(b1);

b1.addActionListener(this);

setVisible(true);

}

@Override

public void actionPerformed(ActionEvent ae)

{

try

{

int a = Integer.parseInt(tf1.getText());

int b = Integer.parseInt(tf2.getText());

if(b==0)

throw new ArithmeticException(" Divide by Zero Error");

float c = (float) a / b;

tf3.setText(String.valueOf(c));

}

catch (NumberFormatException | ArithmeticException ex)

{

JOptionPane.showMessageDialog(this, ex.getMessage());

}

}

}

public class Pgm16

{

public static void main(String[] args)

{

A a = new A();

}

}

**OUTPUT1:**



**OUTPUT2:**





**OUTPUT3:**





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

import java.awt.\*;

import java.applet.\*;

import java.awt.event.\*;

import java.awt.Color;

public class pgm extends Applet implements ActionListener

{

Label l1,l2,l3;

TextField t1,t2,t3;

Button add,sub,mul,div;

public void init()

{

setBackground(Color.lightGray);

//setColor(Color.White);

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);

//Labels...

l1.setBounds(50,100,100,30);

l2.setBounds(50,140,100,30);

l3.setBounds(50,180,100,30);

//TextFields...

t1.setBounds(300,100,100,30);

t2.setBounds(300,140,100,30);

t3.setBounds(300,180,100,30);

//Buttons...

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:**









**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.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

public class Pgm18 extends Applet implements ActionListener

{

Label l1,l2;

Button b1;

TextField t1,t2;

public void init()

{

setLayout(null);

l1 = new Label("Enter the number");

l2 = new Label("The Factorial of given number=");

t1 = new TextField();

t2 = new TextField();

b1 = new Button("Compute");

add(t1);

add(t2);

add(l1);

add(l2);

add(b1);

l1.setBounds(50,100,150,30);

l2.setBounds(50,150,150,30);

t1.setBounds(250,100,150,30);

t2.setBounds(250,150,150,30);

b1.setBounds(230,240,100,30);

setSize(500,500);

b1.addActionListener(this);

}

public void actionPerformed(ActionEvent e)

{

int a = Integer.parseInt(t1.getText());

int fact =1;

for (int i= 1; i<=a; i++)

{

fact=fact\*i;

}

t2.setText(String.valueOf(fact));

}}

**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.**

import java.awt.\*;

import java.applet.\*;

import java.awt.event.\*;

public class pgm19 extends Applet implements ActionListener

{

Label lblTitle,lblRegNo,lblName,lblJava,lblSE;

TextField txtRegNo,txtName,txtJava,txtSE;

Button ButtonClick;

int total;

float avg;

@Override

public void init()

{

setLayout(null);

lblTitle=new Label("Enter Student’s Details");

lblRegNo=new Label("Reg. No:");

lblName=new Label("Name:");

lblJava=new Label("Java:");

lblSE=new Label("SE:");

txtRegNo=new TextField(10);

txtName=new TextField(25);

txtJava=new TextField(3);

txtSE=new TextField(3);

setSize(900,800);

ButtonClick=new Button("View Student Result");

lblTitle.setBounds(100,0,200,20);

lblRegNo.setBounds(0,50,100,20);

txtRegNo.setBounds(120,50,100,20);

lblName.setBounds(0,75,100,20);

txtName.setBounds(120,75,250,20);

lblJava.setBounds(0,100,100,20);

txtJava.setBounds(120,100,40,20);

lblSE.setBounds(0,125,100,20);

txtSE.setBounds(120,125,40,20);

ButtonClick.setBounds(100,225,150,30);

add(lblTitle);

add(lblRegNo);

add(txtRegNo);

add(lblName);

add(txtName);

add(lblJava);

add(txtJava);

add(lblSE);

add(txtSE);

add(ButtonClick);

ButtonClick.addActionListener(this);

}

@Override

public void actionPerformed(ActionEvent ae)

{

try

{

int java=Integer.parseInt(txtJava.getText());

int se=Integer.parseInt(txtSE.getText());

total=(java+se);

avg=total/2;

}

catch(NumberFormatException e)

{

}

repaint();

}

@Override

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("Software Engineering : "+txtSE.getText(),0,375);

g.drawString("Total: "+total,0,475);

g.drawString("Average: "+avg,0,500);

}

}

**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**](http://en.wikipedia.org/wiki/Tic_tac_toe) **You are required to implement this game with two classes, TicTacToeGame and TicTacToeTester**

import java.util.Scanner;

public class Pgm200

{

public static void main(String[ ] args) {

TicTacToe t = new TicTacToe();

Scanner s = new Scanner(System.in);

int x=0,y=0;

do

{

System.out.println(t.player==t.X?"Player X turn":"Player O turn");

System.out.println("Enter x and y places");

x=s.nextInt();

y=s.nextInt();

t.putSign(x, y);

System.out.println(t.toString());

System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

t.displayWinner();

}while(t.isEmpty);

}

}

class TicTacToe

{

public static final int X = 1, O = -1;

public static final int EMPTY = 0;

public int player = X;

private int[][] board = new int[3][3];

public boolean isEmpty = false;

/\*\* Puts an X or O mark at position i,j. \*/

public void putSign(int x, int y)

{

if(x<0 || x>2 || y<0 || y>2)

{

System.out.println("Invalid board position");

return;

}

if(board[x][y] != EMPTY)

{

System.out.println("Board position occupied");

return;

}

board[x][y] = player; // place the mark for the current player

player = -player; // switch players (uses fact that O = - X)

}

/\*\* Checks whether the board configuration is a win for the given player. \*/

public boolean isWin(int player)

{

return ((board[0][0] + board[0][1] + board[0][2] == player\*3) ||

(board[1][0] + board[1][1] + board[1][2] == player\*3) ||

(board[2][0] + board[2][1] + board[2][2] == player\*3) ||

(board[0][0] + board[1][0] + board[2][0] == player\*3) ||

(board[0][1] + board[1][1] + board[2][1] == player\*3) ||

(board[0][2] + board[1][2] + board[2][2] == player\*3) ||

(board[0][0] + board[1][1] + board[2][2] == player\*3) ||

(board[2][0] + board[1][1] + board[0][2] == player\*3));

}

/\*\*display the winning player or indicate a tie (or unfinished game).\*/

public void displayWinner()

{

if(isWin(X))

{

System.out.println("\n X wins...!!");

isEmpty=false;

}

else if(isWin(O))

{

System.out.println("\n O wins...!!");

isEmpty=false;

}

else

{

if(!isEmpty)

{

System.out.println("its a tie");

}

}

}

public String toString()

{

StringBuilder s = new StringBuilder();

isEmpty = false;

for(int i=0;i<3;i++)

{

for(int j=0;j<3;j++)

{

switch(board[i][j])

{

case X:

s.append(" X ");

break;

case O:

s.append(" O ");

break;

case EMPTY:

s.append(" ");

isEmpty=true;

break;

}

if(j<2)

{

s.append("|");

}

}

if(i<2)

{

s.append("\n-----------\n");

}

}

return s.toString();

}

}

**OUTPUT:**

run:

Player X turn

Enter x and y places

1

1

| |

-----------

| X |

-----------

| |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Player O turn

Enter x and y places

0

0

O | |

-----------

| X |

-----------

| |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Player X turn

Enter x and y places

0

2

O | | X

-----------

| X |

-----------

| |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Player O turn

Enter x and y places

2

0

O | | X

-----------

| X |

-----------

O | |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Player X turn

Enter x and y places

2

1

O | | X

-----------

| X |

-----------

O | X |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Player O turn

Enter x and y places

1

0

O | | X

-----------

O | X |

-----------

O | X |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

O wins...!!

BUILD SUCCESSFUL (total time: 1 minute 14 seconds)