Java

NAME :- BIKASH BISHNU CHARANA PRADHAN

ROLL NO.:- 114(class :-119)

1.write a program to display the star patter program.

**a.Right triangle pattern**

import java.util.Scanner;

public class righttri

{

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

System.out.print("Enter the value of x =");

int x=sc.nextInt();

System.out.println("Right triangle is =");

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

{

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

{

System.out.print("\*");

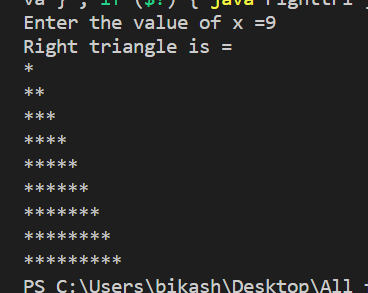
}

System.out.println();

}

}

}



B

import java.util.Scanner;

public class righttri

{

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

System.out.print("Enter the value of x =");

int x=sc.nextInt();

System.out.println("Right triangle is =");

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

{

for(int z=x-i;z>=0;z--)

{

System.out.print(" ");

}

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

{

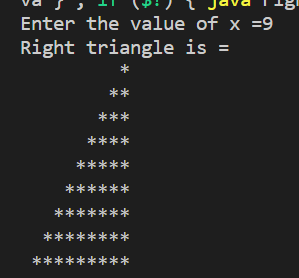
System.out.print("\*");

}

System.out.println();

}

}

}

**c.**

import java.util.Scanner;

public class inverttriangle

{

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

System.out.print("Enter the value of x =");

int x=sc.nextInt();

System.out.println("Right triangle is =");

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

{

for(int j=x-i;j>=1;j--)

{

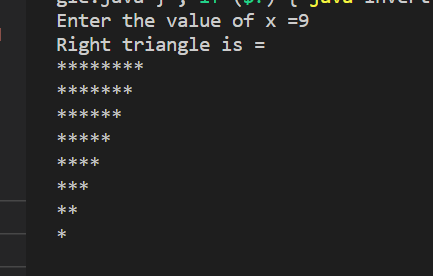
System.out.print("\*");

}

System.out.println();

}

}

}

**d.**

import java.util.Scanner;

public class inverttriangle

{

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

System.out.print("Enter the value of x =");

int x=sc.nextInt();

System.out.println("Right triangle is =");

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

{

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

{

System.out.print(" ");

}

for(int j=x-i;j>=1;j--)

{

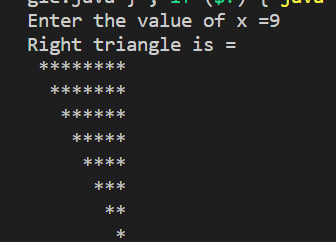
System.out.print("\*");

}

System.out.println();

}

}

}

**2.wap to print duplicate elements of an arrya.**

public class duplicate {

public static void main(String args[])

{

int a[]={10,100,30,40,50,40,100};

System.out.print("Removed duplication array is =");

for(int i=0;i<a.length;i++)

{

for(int j=i+1;j<a.length;j++)

{

if(a[i]==a[j])

{

// a[i]=-1;

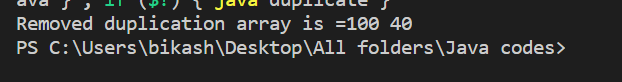
System.out.print(a[i]+" ");

}

}

}

}

}

**3. wap to remove duplicate elements of an arrya.**

public class duplicate {

public static void main(String args[])

{

int a[]={10,100,30,40,50,40,100};

for(int i=0;i<a.length;i++)

{

for(int j=i+1;j<a.length;j++)

{

if(a[i]==a[j])

{

a[i]=-1;

}

}

}

System.out.print("Removed duplication array is =");

for(int i=0;i<a.length;i++)

{

if(a[i]!=-1)

{

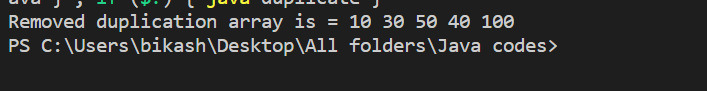
System.out.print(" "+a[i]);

}

}

}

}



**4.WAP TO CHECK A NUMBER IS PERFECT NUMBER OR NOT.**

import java.util.Scanner;

public class perfectno {

public static void main(String args[])

{

int sum=0;

int num;

System.out.print("Enter any number =");

Scanner sc = new Scanner(System.in);

num =sc.nextInt();

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

{

if(num%i==0)

{

sum=sum+i;

}

}

if(num==sum)

{

System.out.print("Perfect number.");

}

else

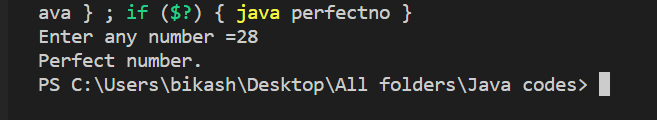
{

System.out.print("Not perfect number.");

}

}

}



**5.Write a program to display all string methods, stringbuilder methods, and string buffer methods.**

import java.io.\*;

class string

{

String st;

public string()

{

this("");

}

public string(String st)

{

this.st = st;

}

public void to\_String()

{

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

}

public void string\_length()

{

System.out.println("Length Of String = "+st.length());

}

public void string\_charat(int c)

{

char ch = st.charAt(c);

System.out.println("charAt Of String = "+ch);

}

public void string\_getchars(int start,int end)

{

char arr[] = new char[end-start];

st.getChars(start,end,arr,0);

System.out.print("GetChars of String = ");

System.out.println(arr);

}

public void string\_tochararray()

{

char arr[] = st.toCharArray();

System.out.print("toCharArray of String = ");

System.out.println(arr);

}

public void string\_equalsto(String str2)

{

System.out.println("EqualsTo of String = "+st.equals(str2));

}

public void start\_endwith()

{

System.out.println("Startwith of String (Java) = "+st.startsWith("Java"));

System.out.println("Endwith of String (s) = "+st.endsWith("s"));

}

public void string\_compare(String str2)

{

System.out.println("CompareTo of String = "+st.compareTo(str2));

}

public void string\_indexof(char ch1)

{

System.out.println("Index of String = "+st.indexOf(ch1));

}

public void string\_lastindexof(char ch2)

{

System.out.println("Last Index of String = "+st.lastIndexOf(ch2));

}

public void string\_substr(int end)

{

System.out.println("Substring of String = "+st.substring(end));

}

}

public class string\_methods

{

public static void main(String args[])

{

int start = 2;

int end = 4;

char ch1 = 'J';

char ch2 = 'o';

string str = new string("HewlloWorld Java Programs");

string str2 = new string("hello");

System.out.print("String 1 = ");

str.to\_String();

String s2 = str2.toString();

System.out.print("String 2 =");

str2.to\_String();

str.string\_length();

str.string\_charat(2);

str.string\_getchars(start,end);

str.string\_tochararray();

str.string\_equalsto(s2);

str.start\_endwith();

str.string\_compare(s2);

str.string\_indexof(ch1);

str.string\_lastindexof(ch2);

str.string\_substr(end);

}

}

class string\_buffer

{

public static void main(String args[])

{

StringBuffer sb = new StringBuffer("HewlloWorld Java Programs");

System.out.println(sb);

System.out.println("String Buffer Length = "+sb.length());

System.out.println("String Buffer capacity = "+sb.capacity());

sb.setLength(10);

System.out.println("String Buffer setlength (10) = "+sb);

sb.append("Java").append("Language");

System.out.println("Append String (Java,Language) = "+sb);

sb.insert(10," Programs");

System.out.println("Insert String at 10 index ( Programs) = "+sb);

sb.reverse();

System.out.println("Reverse String = "+sb);

sb.delete(1,6);

System.out.println("Delete String from(1,6) = "+sb);

sb.replace(5,7," Hello ");

System.out.println("Replace String from(5,7) add (Hello) = "+sb);

}

}

class string\_builder

{

public static void main(String args[])

{

StringBuilder sb = new StringBuilder("HewlloWorld Java Programs");

System.out.println(sb);

System.out.println("String Buffer Length = "+sb.length());

System.out.println("String Buffer capacity = "+sb.capacity());

sb.setLength(10);

System.out.println("String Buffer setlength (10) = "+sb);

sb.append("Java").append("Language");

System.out.println("Append String (Java,Language) = "+sb);

sb.insert(10," Programs");

System.out.println("Insert String at 10 index ( Programs) = "+sb);

sb.reverse();

System.out.println("Reverse String = "+sb);

sb.delete(1,6);

System.out.println("Delete String from(1,6) = "+sb);

sb.replace(5,7," Hello ");

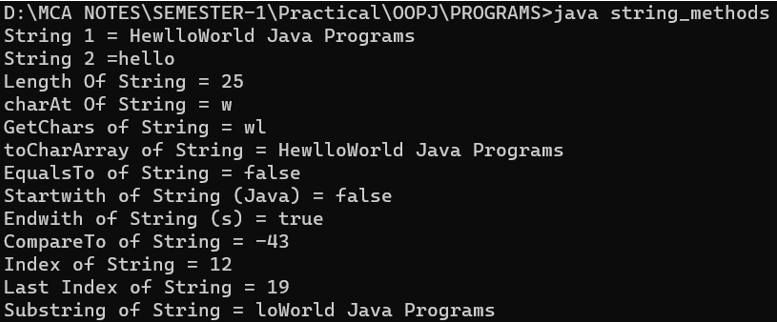
System.out.println("Replace String from(5,7) add (Hello) = "+sb);

}

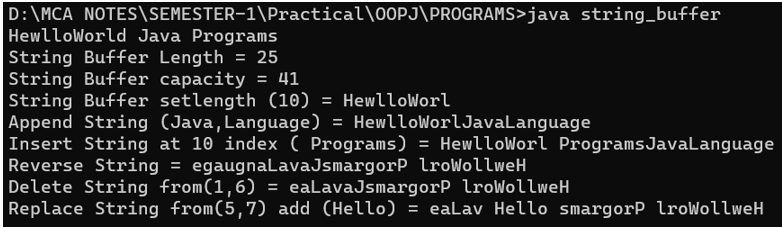
}

**Output :-**

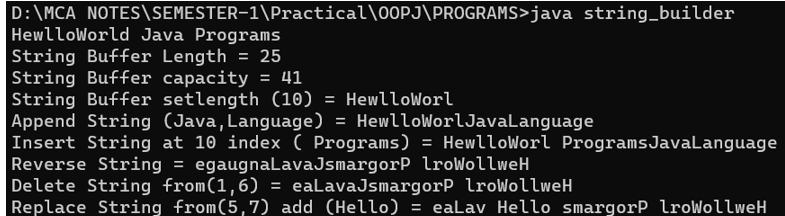
String



String buffer



String builder



**7.Write a program to calculate the area of a rectangular using a procedural way and object-oriented way (using class & methods). (Three Ways).**

**a.Procedural Way**

public class rectangearea

{

public static void main(String []args)

{

int length=50;

int width=40;

int area = length\*width;

System.out.println("Arear of the rectangel is ="+area+" sq.");

}

}

**b.Class inside class**

class rectangle

{

int length;

class rectanglecalculate

{

int breath;

}

}

public class rectangleClassInsideClass

{

public static void main(String args[])

{

rectangle r = new rectangle();

rectangle.rectanglecalculate r1 = r.new rectanglecalculate();

r.length = 40;

r1.breath = 60;

System.out.println("Area Of rectangle = " +r.length\*r1.breath);

}

}

**c.Object Oriented Way**

public class rectangearea

{

public static void main(String []args)

{

Rectangle r=new Rectangle(10,20);

r.Area(40,50);

}

}

class Rectangle

{

int length;

int width;

public Rectangle(int len,int wid)

{

this.length=len;

this.width=wid;

//this (0,0);

}

public int Area(int len,int wid)

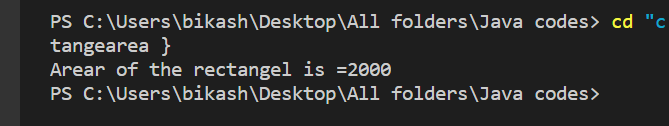
{

System.out.println("Arear of the rectangel is ="+ len\*wid);

return 0;

}

}



**8.WAP to check whether the number is palindrome or not. Input: 528 Output: It is not a palindrome number**

**Input: 545 Output: It is not a palindrome number**

class Check

{

int num;

Check(int num)

{

this.num=num;

}

public void view()

{

int rem;

int x,rev=0;

x=num;

if(num>0)

{

while(num!=0)

{

rem=num%10;

rev=(rev\*10)+rem;

num=num/10;

}

if(x==rev)

{

System.out.println(x+" is a paliandrome number.");

}

else

{

System.out.println(x+" is not a paliandrome number.");

}

}

else

{

System.out.println("Enter a valide number or positive number.");

}

}

}

public class paliandrome {

public static void main(String args[])

{

Check c = new Check(528);

Check c1= new Check(525);

Check c2= new Check(545);

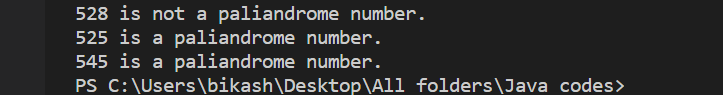
c.view();

c1.view();

c2.view();

}

}



**9.Write a program in Java to multiply two matrices. Declare a class Matrix where the 2D array is declared as an instance variable and the array should be initialized, within the class.**

public class multiplicationmatrix {

public static void main(String args[])

{

int a[][]={{2,3},{5,7}};

int b[][]={{3,7},{10,4}};

int c[][];

c =new int[2][2];

for(int i=0;i<a.length;i++)

{

for(int j=0;j<b.length;j++)

{

c[i][j]=0;

for(int k=0;k<2;k++)

{

c[i][j]=c[i][j]+a[i][k]\*b[k][j];

}

}

}

System.out.println("The mutliplication of the array is =");

for(int i=0;i<a.length;i++)

{

for(int j=0;j<b.length;j++)

{

System.out.print(c[i][j] +"\t");

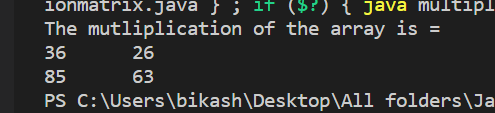
}

System.out.println();

}

}

}



**10.Write a Java application that takes several command line arguments, which are supposed to be names of students and prints output as given below: (Suppose we enter 3 names then the output should be as follows).**

**Number of arguments = 3**

**1.: First Student Name is = Arun**

**2.: Second Student Name is = Hiren**

**3.: Third Student’s Name is = Hitesh**

**(Hint: Initialize string array with “First”, “Second”, etc.)**

import java.util.Scanner;

public class command\_line

{

public static void main(String args[])

{

Scanner sc=new Scanner(System.in);

String [] str={"First","Second","Third","Fourth"};

System.out.println("Number of student = ");

int x=sc.nextInt();

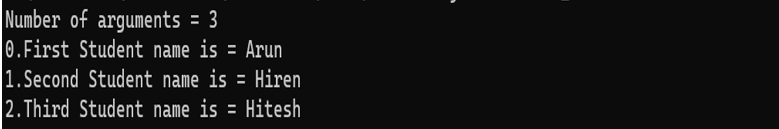
for(int i=0;i<x;i++)

{

System.out.println(i+1+"."+str[i]+" Student name is = "+args[i]);

}

}

}

**11. Write a Java application to count and display the frequency of letters and digits from the String given by the user as a command-line argument.**

public class frequency {

public static void main(String []args)

{

System.out.println("String Passed By Argumnet : "+args[0]);

int letters = 0, digits = 0;

String str = args[0];

str = str.toLowerCase();

for(int i=0; i < str.length(); i++)

{

char ch = str.charAt(i);

if(ch >= 'a' && ch <= 'z')

{

letters++;

}

else if(ch >= '0' && ch <= '9')

{

digits++;

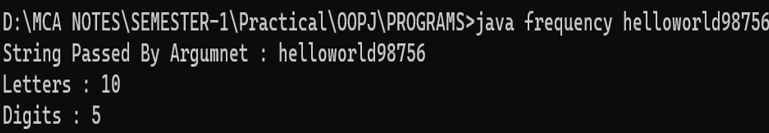
}

}

System.out.println("Letters : " +letters);

System.out.println("Digits : " +digits);

}}



**12.Create a class “Student” that would contain enrollment No, name, and gender and marks as instance variables and count as a static variable that stores the count of the objects; constructors, and display(). Implement constructors to initialize instance variables. Also, demonstrate constructor chaining. Create objects of class “Student” and displays all values of objects**

public class studentdetail {

public static void main(String []args)

{

Student s = new Student(100,"Bikash pradhan","Male",50);

s.display();

Student s1 = new Student(200,"Karishma pradhan","female",80);

s1.display();

}

}

class Student

{

int enroll\_no;

String name;

String gender;

int marks;

public static int count;

public Student()

{

}

public Student(int enroll\_no,String name,String gender,int marks)

{

this.enroll\_no = enroll\_no;

this.name = name;

this.gender = gender;

this.marks = marks;

}

public void display()

{

count++;

System.out.println("Counted Students : "+count);

System.out.println("Enrollment No:- "+enroll\_no);

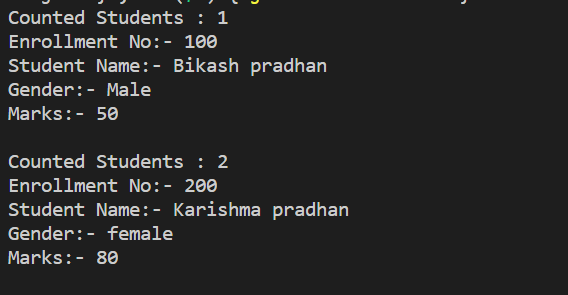
System.out.println("Student Name:- "+name);

System.out.println("Gender:- "+gender);

System.out.println("Marks:- "+marks+"\n");

}

}



**13.Create a class “Rectangle” that would contain length and width as an instance variable and count as a static variable. Define constructors [constructor overloading (default, parameterized, and copy)] To initialize variables of objects. Define methods to find area and to display variables’ Value of objects which are created. [Note: define initializer block, static initializer block, and the static variable and method. Also, demonstrate the sequence of execution of the initializer block and static initialize block].**

public class Rectangel{

public static void main(String args[])

{

Rect r = new Rect();

r.display();

Rect r1 = new Rect(12,30);

r1.display();

Rect r2 = new Rect(r);

r2.display();

}

}

class Rect

{

private int length;

private int width;

public static int count = 0;

public Rect()

{

this(0,0);

System.out.println("Default constructor .");

}

public Rect(int len,int wid)

{

length = len;

width = wid;

System.out.println("Parameterized Constructor.");

}

public Rect(Rect r)

{

this.length = r.length;

this.width = r.width;

System.out.println("Copy Constructor");

}

public int area\_rect(int length,int breath)

{

return length\*breath;

}

public void display()

{

count++;

System.out.println("Number Of Rectangle = "+count);

System.out.println("Length Of Rectangle = "+length);

System.out.println("Breath Of Rectangle = "+width);

System.out.println("Area Of Rectangle = "+area\_rect(length,width)+"\n");}}

**14.Create a class "Vehicle" with instance variable vehicle\_type. Inherit the class in a class called "Car" with instance model\_type, company name etc. display the information of the vehicle by defining the display() in both super and sub class [ Method Overriding]**

public class VehicelP{

public static void main(String args[])

{

Car v= new Car("SUV X200","Maruti",4 );

v.display();

Car v1= new Car("Ferrari Enzo","ferrari",3 );

v1.display();

}

}

class Vehicle {

private int v\_type;

public Vehicle(int v\_type)

{

this.v\_type=v\_type;

}

public void display()

{

System.out.println("vehicel type is = "+v\_type+" wheeler.");

}

}

class Car extends Vehicle

{

private String m\_type;

private String company\_name;

public Car(String m\_typ,String company\_name,int v\_type)

{

super(v\_type);

this.m\_type=m\_typ;

this.company\_name=company\_name;

}

public void display()

{

super.display();

System.out.println(" Model type of the car is = "+m\_type);

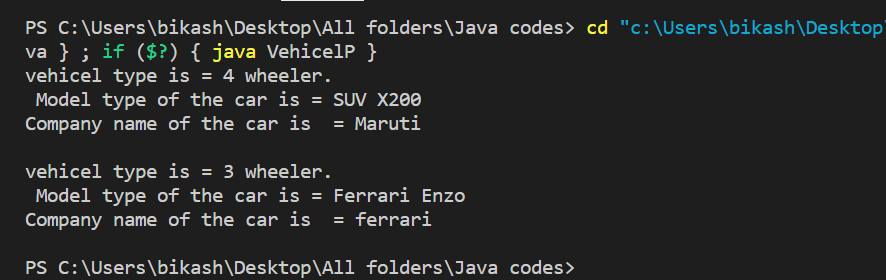
System.out.println("Company name of the car is = "+company\_name);

System.out.println();

}

}

**OUTPUT:**

****

**15.Create a class “Account” containing accountNo, and balance as an instance variable. Derive the Account class into two classes named “Savings” and “Current”. The “Savings” class should contain an instance variable named interestRate, and the “Current” class should contain an instance variable called overdraftLimit. Define appropriate methods for all the classes to enable functionalities to check balances, deposit, and withdraw amounts in Savings and Current accounts. [Ensure that the Account class cannot be instantiated.]**

class Account

{

long ano;

double bal;

Account(long ano,double bal)

{

this.ano=ano;

this.bal=bal;

}

public double chkbal()

{

return bal;

}

}

class Saving extends Account

{

int irate;

public Saving(long ano,double bal,int irate)

{

super(ano,bal);

this.irate=irate;

}

public double chkbal()

{

System.out.println("Account number: "+ super.ano);

double totalAmount;

totalAmount = (super.bal\*irate)/100;

return totalAmount+super.bal;

}

public void withdraw(int amount)

{

if(amount<super.bal)

{

super.bal = super.bal-amount;

System.out.println("Savings Account Balance after withdrawl "+amount+" : " +super.bal);

}

else

{

System.out.println("not avilable Balance");

}

}

public void deposit(int amount)

{

super.bal = super.bal+amount;

System.out.println("Savings Account Balance after Deposit "+amount+" : " +super.bal);

}

}

class Current extends Account

{

int olt;

public Current(long ano,double bal,int olt)

{

super(ano,bal);

this.olt=olt;

}

public double chkbal()

{

System.out.println("Account number: "+ super.ano);

return super.bal;

}

public void deposit(int amount)

{

super.bal = super.bal+amount;

System.out.println("Current Account Balance after Deposit "+amount+" : " +super.bal);

}

public void withdraw(int amount)

{

if(amount<olt)

{

super.bal= super.bal-amount;

System.out.println("Current Account Balance after withdrawl "+amount+" : " + super.bal);

}

else

{

System.out.println("Withdrawl amount exceeds Overdraft limit");

}

}

}

class Acc

{

public static void main(String[] args)

{

System.out.println("ACCOUNT DETAILS ");

System.out.println("\n\n\*\*\*\*\*\*\* SAVING ACCOUNT \*\*\*\*\*\*\*\n\n");

Saving s=new Saving(981234802,100000,10);

System.out.println("Total Balance: " + s.chkbal());

s.withdraw(5000);

s.deposit(10000);

System.out.println("\n\n\*\*\*\*\*\*\* CURRENT ACCOUNT \*\*\*\*\*\*\*\n\n");

Current c=new Current(923479234,150000,5000);

System.out.println("Total Balance: " + c.chkbal());

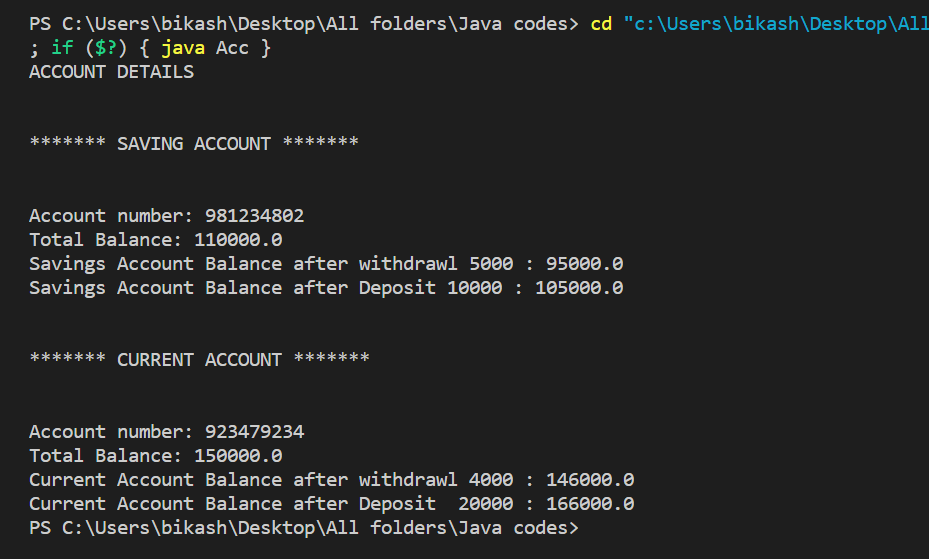
c.withdraw(4000);

c.deposit(20000);

}

}

OUTPUT:



**16 .Describe an abstract class called Shape, which has three subclasses: Triangle, Rectangle, and Circle. Define one method area() in the abstract class and override this area() in these three subclasses to calculate for a specific object i.e. area() of the Triangle subclass should calculate the area of the triangle etc. Same for Rectangle and Circle.**

abstract class Shape

{

private String name;

Shape(String name)

{

this.name=name;

}

public String getname()

{

return name;

}

abstract double area();

}

class Rectangle extends Shape

{

private int h;

private int w;

Rectangle(int h,int w,String name)

{

super(name);

this.h=h;

this.w=w;

}

public double area()

{

return h\*w;

}

}

class Triangle extends Shape

{

private int h;

private int w;

Triangle(int h,int w,String name)

{

super(name);

this.h=h;

this.w=w;

}

public double area()

{

return 0.5\*h\*w;

}

}

class Circle extends Shape

{

private int redius;

Circle(int redius,String name)

{

super(name);

this.redius=redius;

}

public double area()

{

return Math.PI\*redius\*redius;

}

}

class ShapeP

{

public static void main(String args[])

{

Shape s = new Rectangle(20,10,"Rectangle");

System.out.println(""+s.getname()+" in Area:"+s.area());

Shape s1 = new Triangle(50,25,"Triangle");

System.out.println("\n"+s1.getname()+" in Area:"+s1.area());

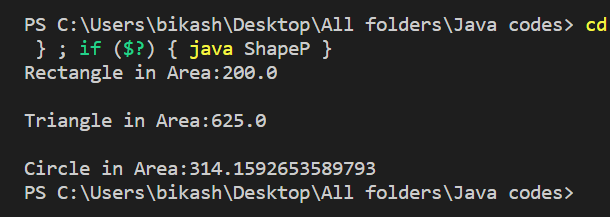
Shape s2=new Circle(10,"Circle");

System.out.println("\n"+s2.getname()+" in Area:"+s2.area());

}

}

OUTPUT:



**17.Define a class A in package apack. In class A, three variables are defined access modifiers protected, private, and public. Define class B in package bpack which extends A and write a display method that accesses variables of class A. Define class C in package cpack which has one method display() in that creates one object of class A and displays its variables. Define class ProtectedDemo in package dpack in which write main () method. Create objects of class B and C and the class display method for both these objects.**

package apack;

public class A

{ private int i1;

public int i2;

protected int i3;

public A()

{

i1=0; i2=0; i3=0;

} public A(int i1,int i2,int i3)

{

this.i1=i1;

this.i2=i2;

this.i3=i3;

} public int getI1()

{

return i1;

} public int getI3()

{

return i3;

}

}

package bpack; import apack.A; public class B extends A

{

public B(){}

public B(int i1,int i2,int i3)

{ super(i1,i2,i3);

}

public void dislay()

{

System.out.println("\*\*\* Inside B CLASS \*\*\*");

System.out.println("private i1 : "+getI1());

System.out.println("public i2 : "+i2);

System.out.println("protected i3 : "+i3);

}

}

package cpack; import apack.A;

public class C

{

public C(){}

public void display()

{

A a1 = new A(1,2,3);

System.out.println("\*\*\* Inside C CLASS \*\*\*");

System.out.println("private i1 : "+a1.getI1());

System.out.println("public i2 : "+a1.i2);

System.out.println("protected i3 : "+a1.getI3());

} } package dpack; import bpack.B; import cpack.C; public class ProtectedDemo

{

public static void main(String[] args)

{

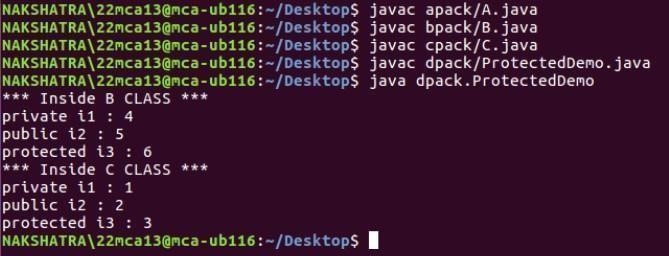
B bb = new B(4,5,6); bb.dislay(); C c1 =new C();

c1.display();

}

}

OUTPUT:-



**18.Write a program in Java to demonstrate throw, throws, and finally, multiple try blocks and multiple catch exceptions.**

import java.util.Scanner;

public class ExceptionHandle

{

public static void main(String[] args)

{

int a ,b ,div;

Scanner sc = new Scanner(System.in);

System.out.print("Enter a first value =");

a = sc.nextInt();

System.out.print("Enter a seconde value =");

b = sc.nextInt();

try{

if(a==0)

{

throw new Exception("First value is not valide .");

}

else{

div=a/b;

System.out.println("Div ="+div);

}

}

catch(ArithmeticException e){

System.out.println("ArithmeticException :"+e);

}

catch(Exception e)

{

System.out.println("Exception:"+e);

}

finally{

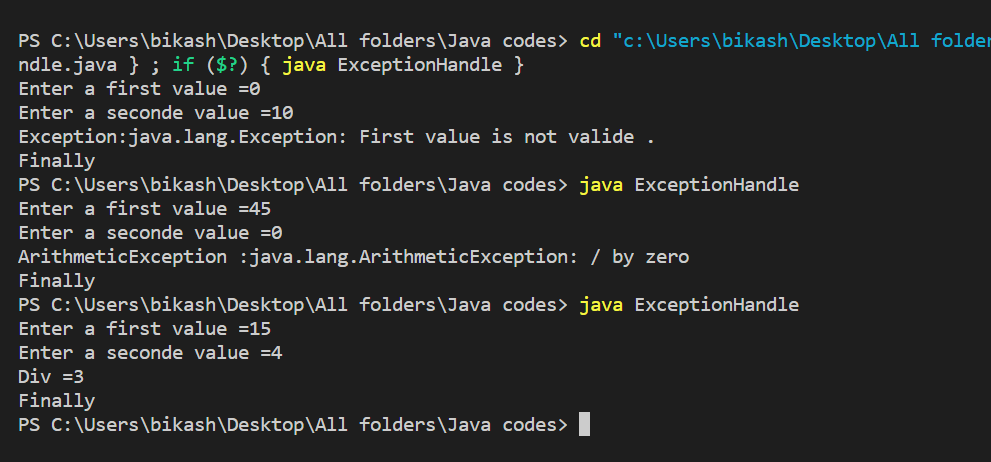
System.out.println("Finally ");

}

}

}

**OUTPUT:**

****

**19.Write a small application in Java to develop a Banking Application in which the user deposits the amount of Rs 1000.00 and then starts withdrawing Rs 400.00, and Rs 300.00 and it throws the exception "Not Sufficient Fund" when the user withdraws Rs. 500 thereafter.**

public class AccountDemo

{

public static void main(String args[])

{

Account s = new Saving(1101,0);

System.out.println("\n\*\*\*\*\* Saving Account Details \*\*\*\*\*");

s.depositBalance(1000);

s.withdrawBalance(400);

s.withdrawBalance(300);

s.withdrawBalance(500);

}

}

abstract class Account

{

private long accountNo=0;

protected double balance=0.0;

public Account()

{

this.accountNo=0;

this.balance=0.0;

}

public Account(long x, double y)

{

this.accountNo = x;

this.balance = y;

}

public void checkBalance(){

System.out.println("\nTotal Balance: "+ balance);

}

public void depositBalance(double x)

{

System.out.println("\nDeposited Ammount: "+ x);

balance = balance + x;

checkBalance();

}

public abstract void withdrawBalance(double x);

}

class Saving extends Account

{

public Saving()

{

this(0,0.0);

}

public Saving(long a,double b)

{

super(a,b);

}

public void withdrawBalance(double x)

{

System.out.println("\nWithdrawn Ammount: "+ x);

if((super.balance-x) > 0)

{

super.balance = super.balance - x;

checkBalance();

}

else

{

String msg="";

msg+="\nYou can not withdraw "+x;

msg+="\ninsufficient balance in your account"; msg+="\nSaving Account balance :"+super.balance; msg+="\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

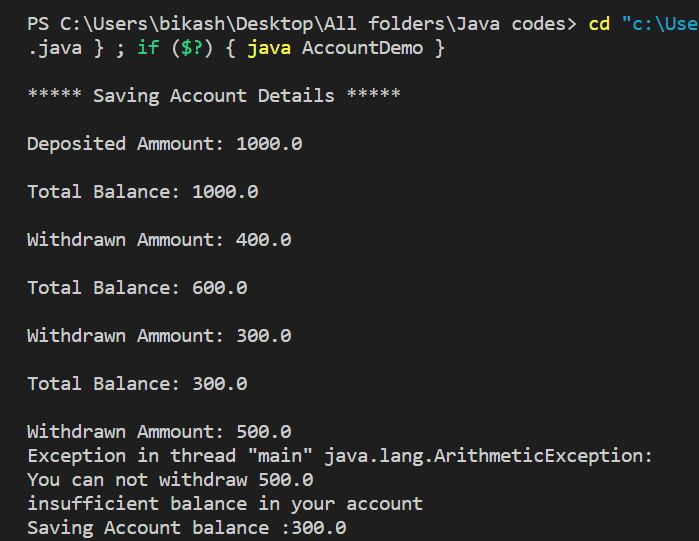
throw new ArithmeticException(msg);

}

}

}

OUTPUT:



**20.Write a program to implement the concept of threading by extending the “Thread” Class.**

class ThreadT extends Thread

{

public ThreadT()

{

super("Child Thread");

start();

}

public void run()

{

for(int i=0;i<=10;i++)

{

System.out.println("In Thread " +Thread.currentThread().getName() + " : i = " + i);

}

}

}

class ThreadClass

{

public static void main(String[] args)

{

ThreadT tt = new ThreadT();

for(int i=0;i<=10;i++)

{

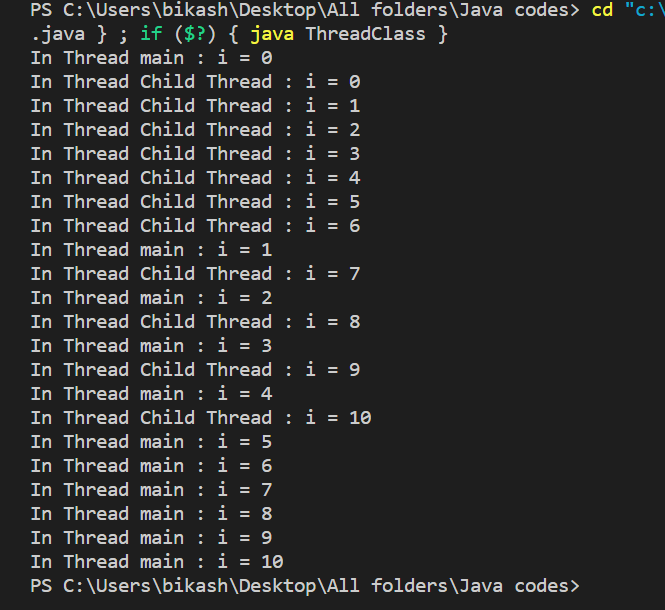
System.out.println("In Thread " +Thread.currentThread().getName() + " : i = " + i);

}

}

}

OUTPUT:



**21.Write a program to implement the concept of threading by implementing the “Runnable” Interface.\**

class ThreadR implements Runnable

{

public ThreadR() {

Thread t = new Thread(this);

t.start();

}

public void run() {

for (int i=0; i<10; i++) {

System.out.println("In Thread " + Thread.currentThread().getName() + " :

i = " + i);

}

}

}

class SampleThreadR

{

public static void main(String[] args)

{

ThreadR tr = new ThreadR();

for (int i=0; i<10; i++) {

System.out.println("In Thread " + Thread.currentThread().getName() + " :

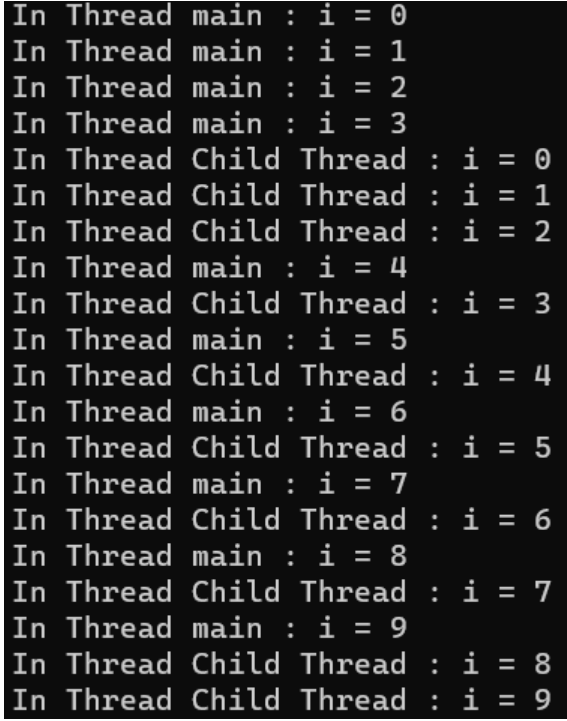
i = " + i);

}

}

}

OUTPUT:



**22.Write a program that executes two threads. One thread displays “Thread1” every 2,000 milliseconds, and the other displays “Thread2” every 4,000 milliseconds.**

class ThreadT extends Thread

{

private int miliseconds;

public ThreadT(String s, int miliseconds)

{

super(s);

this.miliseconds = miliseconds;

start();

}

public void run()

{

try{

Thread.sleep(miliseconds);

System.out.printf("\nIn " +Thread.currentThread().getName());

}

catch(Exception e){}

}

}

class MultiThread1

{

public static void main(String[] args)

{

System.out.println("In Thread " +Thread.currentThread().getName());

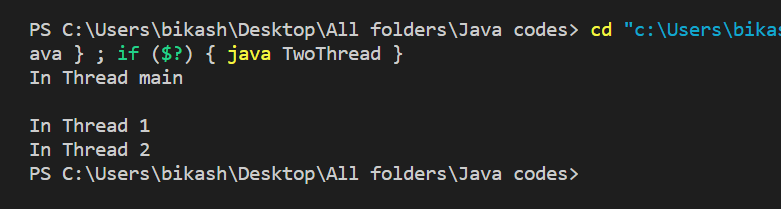
ThreadT t1= new ThreadT("Thread 1",2000);

ThreadT t2= new ThreadT("Thread 2",4000);

}

}

**OUTPUT:**

****

**23.Write a program that creates and executes at least 2 threads. Each of the threads is trying to deposit and withdraw money from the same Account.**

public class SampleThreadSync implements Runnable

{

ThreadAccount ta = null;

String threadName = null;

public SampleThreadSync(ThreadAccount ta, String name) {

this.ta = ta;

this.threadName = name;

Thread t = new Thread(this, name);

t.start();

}

public static void main(String[] args) {

ThreadAccount ta1 = new ThreadAccount("A", 1000);

ThreadAccount ta2 = new ThreadAccount("B", 2000);

SampleThreadSync t1 = new SampleThreadSync(ta1, "Thread1");

SampleThreadSync t2 = new SampleThreadSync(ta2, "Thread2");

}

public void run() {

if (threadName.equals("Thread1")) {

ta.deposit(500);

} else if(threadName.equals("Thread2")) {

ta.withdraw(200);

}

}

}

class ThreadAccount

{

private String name;

private double balance;

public ThreadAccount(String name, double balance) {

this.name = name;

this.balance = balance;

}

public synchronized void deposit(double amt) {

balance += amt;

for (int i = 0; i < 10; i++) {

System.out.println("In thread " + Thread.currentThread().getName() + "

deposit method: i = " + i);

}

}

public synchronized void withdraw(double amt) {

balance -= amt;

for (int i = 0; i < 10; i++) {

System.out.println("In thread " + Thread.currentThread().getName() + "

withdraw method: i = " + i);

}

}

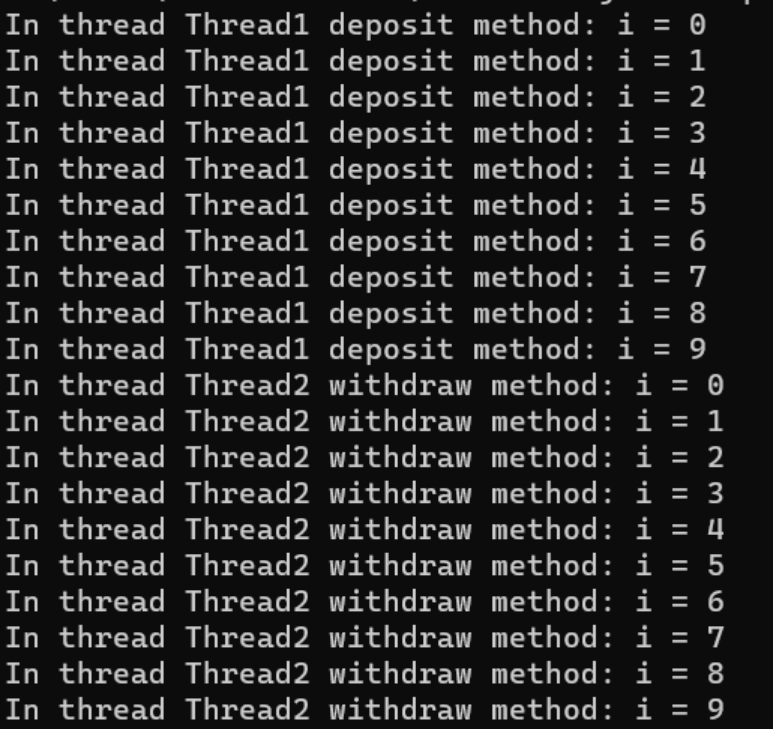
public String toString() {

return "Name = " + name + " and balance = " + balance;

}

}

OUTPUT:



**24.Write a Java program to create a Frame that includes the Student’s name, Student Marks, and Out of Marks. Create a button to calculate the percentage. Clicking the button should display the percentage in another Percentage textfield which is disabled. Users should not be able to enter characters in the Marks textfield. Use Key Listener to check.**

import java.awt.\*;

import java.awt.event.\*;

class SampleFrame implements ActionListener, WindowListener, TextListener

{

private Frame f = null;

private Button b1 = null;

private TextField t1,t2,t3,t4 = null;

private Label l1,l2,l3,l7 = null;

private String s1,s2,s3,result;

public static void main(String []args)

{

SampleFrame sf = new SampleFrame();

}

public SampleFrame()

{

f = new Frame("Student Marks Calculate");

f.setLayout(new FlowLayout());

f.addWindowListener(this);

f.setSize(600,400);

f.setResizable(true);

f.setVisible(true);

l1 = new Label("Student Name");

f.add(l1);

t1 = new TextField(50);

t1.addTextListener(this);

f.add(t1);

l2 = new Label("Student Marks");

f.add(l2);

t2 = new TextField(50);

t2.addTextListener(this);

f.add(t2);

l3 = new Label("Out Of Marks");

f.add(l3);

t3 = new TextField(50);

t3.addTextListener(this);

f.add(t3);

b1 = new Button("Calculate Percentage");

b1.addActionListener(this);

f.add(b1);

l7 = new Label("Percentage");

l7.setVisible(false);

f.add(l7);

t4 = new TextField(50);

t4.setEditable(false);

t4.setVisible(false);

f.add(t4);

}

public void actionPerformed(ActionEvent ae)

{

if(ae.getSource() == b1)

{

s1 = t1.getText();

s2 = t2.getText();

s3 = t3.getText();

double marks = Double.parseDouble(s2);

double outofmarks = Double.parseDouble(s3);

double calculate\_marks = marks\*100/outofmarks;

result = String.valueOf(calculate\_marks);

t4.setText(result);

l7.setVisible(true);

t4.setVisible(true);

System.out.println("Button clicked");

}

}

public void textValueChanged(TextEvent e)

{

s1 = t1.getText();

s2 = t2.getText();

s3 = t3.getText();

}

public void windowClosing(WindowEvent we)

{

System.out.println("In windowClosing");

f.dispose();

}

public void windowClosed(WindowEvent we)

{

System.out.println("In windowClosed");

System.exit(0);

}

public void windowOpened(WindowEvent we)

{

System.out.println("In windowOpened");

}

public void windowActivated(WindowEvent we)

{

System.out.println("In windowActivated");

}

public void windowDeactivated(WindowEvent we)

{

System.out.println("In windowDeactivated");

}

public void windowIconified(WindowEvent we)

{

System.out.println("In windowIconified");

}

public void windowDeiconified(WindowEvent we)

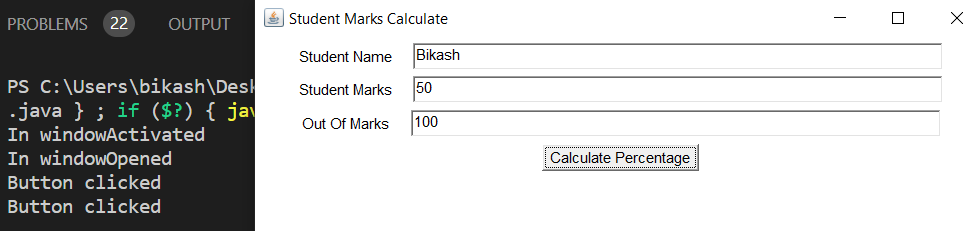
{

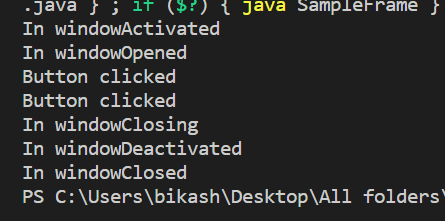
System.out.println("In windowDeiconified");

}

}

OUTPUT:





**END**