

Q) program to find roots of quadratic equation.

⇒ public class Quadratic

{
public static void main (String args []) :

double a = 2.0;

double b = 4.0;

double c = -5.0;

double root1, root2;

double determinant = b * b - 4 * a * c;

if (determinant > 0)

{

System.out.println ("

root1 = (-b + Math.sqrt(determinant)) / (2 * a);

root2 = (-b - Math.sqrt(determinant)) / (2 * a);

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

else if (determinant == 0)

{

root1 = root2 = - b / (2 * a);

System.out.println ("root1 = root2 = % 2f (% 2f, root1);

else

{ double real part = - b / (2 * a);

double imaginary part = Math.sqrt(-determinant) / (2 * a);

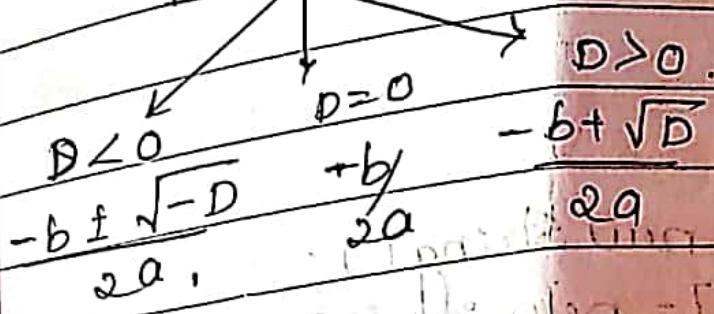
System.out.println ("real part = % 2f + i % 2f imaginary part = % 2f");

`System.out.println("root1 = " + realpart + " + " + imaginarypart + "i")
and root2 = " + realpart - " + imaginarypart + "i, realpart,
imaginary part, realpart, imaginarypart);`

Algorithm: quadratic determinant: $b^2 - 4ac$

where a, b, c are variable

$$b^2 - 4ac = D$$



output:

$$\text{root } 1 = -0.87 + 1.30i \text{ and}$$

$$\text{root } 2 = -0.87 - 1.30i$$

```
1 public class main
2 {
3     public static void main(String[] args) {
4
5         double a = 2.3, b = 4, c = 5.6;
6         double root1, root2;
7
8         double determinant = b * b - 4 * a * c;
9
10        // condition for real and different roots
11        if(determinant > 0) {
12            root1 = (-b + Math.sqrt(determinant)) / (2 * a);
13            root2 = (-b - Math.sqrt(determinant)) / (2 * a);
14
15            System.out.format("root1 = %.2f and root2 = %.2f", root1 , root2);
16        }
17        // condition for real and equal roots
18        else if(determinant == 0)
19        {
20            root1 = root2 = -b / (2 * a);
21
22            System.out.format("root1 = root2 = %.2f;", root1);
23        }
24        // If roots are not real
25        else {
26            double realPart = -b / (2 * a);
27            double imaginaryPart = Math.sqrt(-determinant) / (2 * a);
28
29            System.out.format("root1 = %.2f+%.2fi and root2 = %.2f-%.2fi", realPart,
30                            imaginaryPart, realPart, imaginaryPart);
31        }
32    }
33 }
```

```
$javac main.java
$java -Xmx128M -Xms16M main
root1 = -0.87+1.30i and root2 = -0.87-1.30i
```

Waiting for...

Lab program 2 :

Develop a java program to create a class Student with members, U.S.N, Name, and marks & credits & an arraymarks. Include methods to accept & display details of a method to calculate SGPA of a student.

import java.util.Scanner;

/import java.lang.Math;

class student

{

int USN;

String name = new String();

int credits[] = new int[5];

int marks[] = new int[5];

float SGPA()

{

float sum = 0;

for (int i=0; i<5; i++)

{

sum = sum + (credits[i] * marks[i]);

}

return sum/5;

}

}

public class main

{

public static void main(String args[])

{

Scanner in = new Scanner(System.in);

```
Student stud1 = new Student();
System.out.println ("ENTER DETAILS");
System.out.println ("NAME : ");
System.out.println ("Enter the credits");
for (int j=0 ; j<5 ; j++)
{
    System.out.println ("Subject " + (j+1));
    int cred = in.nextInt();
    stud1.credits[j] = cred;
}
```

```
System.out.println ("Enter the marks");
for (int j=0 ; j<5 ; j++)
{
    System.out.println ("Subject " + (j+1));
    int mark = in.nextInt();
    stud1.marks[j] = mark;
}
```

```
System.out.println ("student Details");
System.out.println ("Name : " + stud1.name);
System.out.println ("USN : " + stud1.usn);
System.out.println ("SGPA : " + stud1.SGPA);
}
```

ENTER DETAILS

NAME:

SHWETA

USN:

21

Enter the Credits

subject 1

4

subject 2

4

subject 3

4

subject 4

4

subject 5

4

Enter the Marks

subject 1

9

subject 2

6

subject 3

7

DB subject 4

7

subject 5

Enter the Marks

Subject 1

Subject 2

Subject 3

Subject 4

Subject 5

Student Details :

Name : SHWETA

Usn : 24

SGPA : 7.4

**...Program finished with exit code 0
Press ENTER to exit console.**

Lab-4 program

```
import java.util.*;  
public class lab-program{  
    public static String name;  
    public static String author;  
    public static double price;  
    public static int no_of_page;  
    public static void main (String [] args){  
    }
```

```
Scanner sc = new Scanner (System.in);  
int n;  
System.out.println ("Enter the no. of books");  
int [] array = new int [n];  
for (int i=0; i<n; i++)  
{  
    System.out.println ("Enter the name of the book");  
    name = sc.next();  
    System.out.println ("Enter the author of the book");  
    author = sc.next();  
    System.out.println ("Enter the price of book");  
    price = sc.nextDouble();  
    System.out.println ("Enter the number of pages  
    of the book");  
    no_of_page = sc.nextInt();  
    for (int i=0; i<n; i++)  
    {  
        System.out.println ("Displaying the details of book");  
        System.out.println ();  
        System.out.println (array[i]);  
    }
```

Output:

Enter no. of books 2

Enter the name of Book & Show

Enter the author of book 1 of C.

Enter the price of the book \$ 90.

Enter the number of pages of book 1. 800.

Enter the name of book 2. swati

Enter the author of book2 or

Enter the price of book2 398.00000000000004

Enter the no.of pages of book 2 600.

Displaying the details of the book.

Displaying the details of book 2.

Algol -

- initializing a class lab program
 - initializing static variable (name, author etc.)
 - taking input from the user using Scanner
 - initializing array to limit books (details)
 - printing the details using for loop

```
3  
sheer  
ok  
90  
900  
SMA  
or.  
398  
600  
SAK  
SXX  
70  
500
```

```
$javac lab_program3.java  
$java -Xmx128M -Xms16M lab_program  
Enter the number of books  
Enter the name of the book 1  
Enter the author of the book 1  
Enter the price of the book 1  
Enter the number of pages of book 1  
Enter the name of the book 2  
Enter the author of the book 2  
Enter the price of the book 2  
Enter the number of pages of book 2  
Enter the name of the book 3  
Enter the author of the book 3  
Enter the price of the book 3  
Enter the number of pages of book 3  
Displaying the details of the book 2  
0
```

Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes containonly the method printArea() that prints the area of the given shape.

↳ Abstract class shape

double dim1;

double dim2;

Shape (double a, double b)

{

dim1 = a;

dim2 = b;

b

abstract double area()

{

↳ class Rectangle extends Shape

{

rectangle (double a, double b)

Super (a, b);

{

double area()

{

System.out.println ("inside area of rectangle")

return dim1 * dim2;

{

{

class triangle extends Shape

{

triangle (* double a, double b)

{

super (a, b);

{

double area()

```
1
System.out.println ("inside area of triangle");
return (dim1 + dim2)/12;
2
```

class circle extends shape

```
1
circle (double a, double b) {
    super(a,b);
}
2
```

double area()

```
1
System.out.println ("inside area of circle");
return dim1 * dim2 * 3.14;
2
```

```
3
```

public class Abstractclass

```
1
public static void main (String args[])
{
}
```

rectangle r1 = new rectangle (3,5);

triangle t = new triangle (10,6);

circle c = new circle (4,4);

shape shape;

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

System.out.println ("area of triangle t = " + t.area());

shapef = i

System.out.println("area of triangle"+shapef.i)

shapef = c;

System.out.println("area of circle"+shapef.area);
}

}

Output :-

inside area of rectangle

area of rectangle 45.0

inside area of triangle

area of triangle 30.0

inside area of circle

area of circle 50.24

```
class triangle extends shape
{
    triangle(double a,double b)
    {
        super(a,b);
    }
    double area()
    {
        system.out.println("inside area of triangle");
        return (dim1*dim2)/2;
    }
}

class circle extends shape
{
    circle(double a,double b)
    {
        super(a,b);
    }
    double area()
    {
        system.out.println("inside area of circle");
        return 3.14*dim1*dim2;
    }
}
public class abstractareas
{
    public static void main(string args[])
    {
        rectangle r = new rectangle(0,5);
        triangle t = new triangle(10,5);
        circle c = new circle(4,5);
        shape sharef;
        sharef = r;
        system.out.println("area of rectangle "+sharef.area());
        sharef = t;
        system.out.println("area of triangle "+sharef.area());
        sharef = c;
        system.out.println("area of circle "+sharef.area());
    }
}
```

Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed. Create a class Account that stores customer name, account number and type of account. From this derive the classes Curr-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:

- Accept deposit from customer and update the balance.
- Display the balance.
- Compute and deposit interest
- Permit withdrawal and update the balance
- Check for the minimum balance, impose penalty if necessary and update the balance

② import java.util.Scanner;

class Bank

{

int deposit, balance;

int withdraw, bal;

String customername;

String Acc_num;

String Acc_type;

int balance = 20,000;

void accept()

{

Scanner s = new Scanner (System.in);
System.out.println ("enter customername");
customername = s.nextLine();

System.out.println ("enter Acc_num");

Acc_num = s.nextInt();

System.out.println ("enter Acc-type/b");

Acc_type = s.nextLine();

}

void display()

{

System.out.println ("customername : " + customername);

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

System.out.println ("Account type : " + Acc_type);

}

class curr_acct extends Bank

int updated_balance;

int After withdrawn;

int updated_lost_balance;

```
int cdepo_bal()
```

```
{
```

updated balance = Balance + deposit balance;
return updated balance;

```
}
```

```
int cwithd_bal()
```

```
{
```

After withdraw = (updated balance) - withdraw
return After withdraw;

```
}
```

```
int minimum()
```

```
{
```

if (c_after.withdrawn) <= (2000)

updated lost cbalance = (c_after.withdrawn) - 2000;

System.out.println ("you have minimum balance below
2000 so u have lost 200 rupees");

return updated.lost_cbalance;

```
}
```

```
else
```

return After.cwithdrawn;

```
}
```

```
}
```

```
{
```

```
int updated_balance;
```

```
int after_withdrawn;
```

```
int updated_lost_sbalance;
```

```
compound_interest;
```

```
int sdeposit()
```

```
    updated balance = Balance + deposit - balance;
```

```
    return updated balance;
```

```
} int interest()
```

```
double rate = 0.08;
```

```
int n = 12;
```

```
int t = 5;
```

```
compound interest = (int)((updated balance) * (rate / 12))
```

```
+ (r(n)), cn + 1));
```

```
return compound interest;
```

```
}
```

```
int withdraw()
```

```
after withdraw = (compound interest) - withdraw balance;
```

```
return after withdraw.
```

```
}
```

```
int minimum()
```

```
{
```

```
if (after withdraw) <= (1000)
```

```
{
```

```
updated lost sbalance = (after withdraw) - (100);
```

```
return updated lost sbalance;
```

```
}
```

```
else
```

```
return after withdraw;
```

```
}
```

```

public class main{
    public static void main(String args[]){
        Scanner r = new Scanner(System.in);
        Curr acct ca = new curr_acct();
        ca.accept();
        System.out.println("Enter the money to be deposited");
        ca.deposit_balance = r.nextInt();
        ca.display();
        System.out.println("After your deposition of " + ca.deposit_balance + " now your total balance is " + (ca.deposit_balance + ca.withdraw_balance));
        System.out.println("Enter money to be withdrawn");
        ca.withdrawn_balance = r.nextInt();
        System.out.println("After your withdrawal of " + ca.withdraw_balance + " now your total balance is RS - " + ca.withdraw_balance);
        System.out.println("After checking if u have minimum balance or not your updated total balance is RS - " + ca.minimum());
        Sav acct SA = new Sav_acct();
        SA.accept();
        System.out.println("Enter money to be deposited");
        SA.deposit_balance = r.nextInt();
        SA.display();
        System.out.println("After your deposition of " + SA.deposit_balance + " now your total balance is RS - " + (SA.deposit_balance));
        System.out.println("After interest on updated balance is RS - " + SA.interest());
    }
}

```

System.out.println ("Enter money to be withdrawn
from my saving account");

SA.withdraw balance = g.nextInt();

System.out.println ("After your withdrawal of
RS - " + SA.withdraw balance + " now your total
balance is RS - " + SA.withdraw balance);

System.out.println ("After checking if u have
minimum balance are not your updated total
balance is RS - " + SA.minimum));

}

1. Problems • Invalid ID Declaration Correct

2. Determined Transcations [Java Application] C:\Users\shivanshu\Downloads\javaprojects\src\java\com\javatpoint\Bank\Bank.java At line 22 cols 64, 1402, v02202116021

Enter the customer name

shivanshu

Enter the Account Number

3: Enter the Account type

current

Enter the money u want to deposit in current account in rupees

1000

CUSTOMER NAME : shivanshu

ACCOUNT NUMBER : 911

ACCOUNT TYPE : current

After your deposition of 1000

Now your total balance is RS-23800

Enter the money you want to withdraw in rupees

500

After your withdrawal of 500

Now your total balance is RS-23300

After checking if u have minimum balance are not your updated total balance is RS-28000

Enter the customer name

shivanshu

Enter the Account Number

4: Enter the Account type

savings

Enter the money u want to deposit in Saving account

10000

CUSTOMER NAME : shivanshu

ACCOUNT NUMBER : 911

ACCOUNT TYPE : savings

After your deposition of 10000

Now your total balance is RS-37800

After interest or updated balance is RS-36111

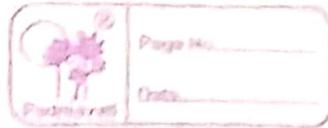
Enter the money you want to withdraw in Saving account

312

After your withdrawal of RS-312

Now your total balance is RS-36000

```
import java.util.*;
import java.io.*;
class demo
{
    public static void main (String [] args)
    {
        int n; Scanner sc=new Scanner (System.in);
        System.out.println ("Enter number of students");
        n= sc.nextInt ();
        Student [] stud= new Student [n];
        Interval [] im= new Interval [n];
        int total [][] [] = new int [n] [5];
        Score [] scr= new Score [n]; int temp=0;
        for (int i=0; i<n; i++)
        {
            im [i]= new Interval ();
            scr [i]= new Score ();
        }
        for (int j=0; j<n; j++)
        {
            im [j].getcl ();
            scr [j].getscr ();
            for (int k=0; k<5; k++)
            {
                total [j] [k] = im [j].marks (k) + scr [j].marks (k);
            }
            System.out.println ("marks in subject " + (j+1) + " by " + im [j].name + " is " + total [j] [1]);
        }
    }
}
```



```
for (int i=0; i<5; i++)  
    temp = temp + total[i][j];  
System.out.println("total mark by " + total[j].name  
+ " is " + temp); temp = 0;  
}  
}
```

* Internals

```
package CIE;  
import java.util.*;  
public class Internals extends Student  
{  
    public void getdata()  
    {
```

```
        super.getdata(); System.out.println("marks  
        marks in Internals:");
```

```
        for (int i=0; i<5; i++)  
            marks[i] = sc.nextInt();  
    }
```

```
}
```

* Scorey

```
package SEE;  
import java.util.*;  
import CIE.*;  
public class Scorey // extends student  
{
```

```
    Scanner sc = new Scanner (System.in);
```

```
System.out.println("Enter marks internal:  
for(int i=0;i<5;i++)  
    marks[i]=sc.nextInt();  
}  
}
```

```
package CDF;  
import java.util.*;  
public class student {  
    public String USN;  
    public String name;  
    public int sum;  
    Scanner sc = new Scanner(System.in);
```

```
public void getdata() {  
    System.out.println("Enter your USN :");  
    USN = sc.next();  
}
```

```
System.out.println("Enter your name :");  
name = sc.nextLine();
```

```
System.out.println("Enter the semester :");  
sum = sem = sc.nextInt();  
}  
}
```

2. ROLL NO/USN:

131

3. COURSE HANDLED BY PROFESSOR :

JAVA

2. DEPT NO OF FACULTY:

123

FACULTY-ENTER IN FOLLOWING ORDER 1. NAME:

DR. KUMAR SWAMY

THE DETAILS OF STUDENT IS NAME AND ROLL NO : SHIVANSHU
151

THE DETAILS OF FACULTY IS NAME & DEPT NO: DR. PALLAVI
111

THE DETAILS OF STUDENT IS NAME AND ROLL NO : SAMRTHCSHETTY
131

THE DETAILS OF FACULTY IS NAME & DEPT NO: DR. KUMAR
123

program - 7 :

1. write a program to demonstrate generics with multiple object parameters.

```
class Type<T> {
    T ob1;
    class TwoGen<T, V> {
        Gen<T> {
            T ob1;
            V ob2;
            TwoGen(T o1, V o2) {
                ob1 = o1;
                ob2 = o2;
            }
        }
    }
}
```

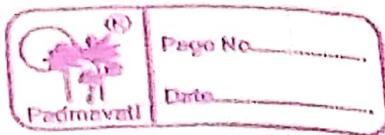
```
void showTypes() {
    System.out.println("Type of T is " + ob1.getClass().getName());
    System.out.println("Type of V is " + ob2.getClass().getName());
}
```

```
T getOb1() {
    return ob1;
}
```

```
V getOb2() {
    return ob2;
}
```

```
class SimpleGen {
```

```
    public static void main (String args[])
    }
```



```
Two Gen<int, String> tObj = new Two  
Gen<int, String>(88, "Generics");  
tObj.setObj1(1);  
int v = tObj.getObj1();  
System.out.println("value : "+v);  
String str = tObj.getObj2();  
System.out.println("value : "+str);
```

Result

```
$javac SimpGen.java
```

```
$java -Xmx128M -Xms16M SimpGen
```

Type of T is java.lang.Integer

Type of V is java.lang.String

value: 88

value: Generics

55) write a program that demonstrates handling of exception in inheritance tree.
create a base class called 'Father' and derived class called "son" which extends the base class. In father class implement a constructor which takes the age & throws the exception wrong age(). When input age < 0, In son class implement a constructor that cases both father & son's age & throws an exception if son's age is \geq father's age
 \Rightarrow class wrong age extends exception

d
public String toString()

L
return "Please enter the right age.";

b

class Father

d

int age;

Father(int age)

f

age = age();

System.out.println ("Father age: " + age);

b

class son extends Father

f

```
super (age);
System.out.println ("son age: " + age);
}
class final
public static void main (String args[]) throws
    wrongAge
{
    int i = args.length;
    int j = Integer.parseInt (args[0]);
    int k = Integer.parseInt (args[1]);
    if (i < 0 || j <= k)
        throw new wrongAge ();
    else
    {
        Father f = new Father (j);
        Son s = new Son (k);
        f
    }
}
```

week-11

MULTIPLE THREAD

1) class NewThread implements Runnable

```

    Thread t;
    NewThread() {
        t = new Thread(this, "SSE Thread");
        t.start();
    }
    public void run() {
        try {
            for(int i=5; i>0; i--) {
                System.out.println("SSE (" + i + ")");
                Thread.sleep(2000);
            }
        } catch(InterruptedException e) {
            System.out.println("SSE interrupted");
        }
    }
}

```

class NewThreads implements Runnable {

```

    Thread t;
    NewThreads() {
        t = new Thread(this, "BMS Thread");
        t.start();
    }
    public void run() {
        try {
            for(int i=5; i>0; i--) {
                System.out.println("BMS College of Engg
                    neering (" + i + ")");
                Thread.sleep(10000);
            }
        } catch(InterruptedException e) {
    }
}

```



```
1
System.out.println ("BMS interrupted.");
}
}
}

public class Main
{
    public static void main (String args[])
    {
        new NewThread2 ();
        new NewThread1 ();
    }
}
```

Output:

BMS College of Engineering (S) i
CSE (5)
CSE (4)
CSE (3)
CSE (2)
CSE (1)

```
$javac Main.java
```

```
$java -Xmx128M -Xms16M Main
```

BMS College Of Engineering(5)

CSE(5)

CSE(4)

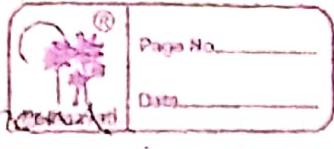
CSE(3)

CSE(2)

CSE(1)

18/12/20

Synchronized program



Page No.

Date

```
* import java.util.Scanner;
public class tables {
    public static void main (String args[])
    {
        cal c = new cal();
        Scanner sc = new Scanner (System.in);
        System.out.println ("input the number of elements");
        int n = sc.nextInt();
        table obj = new table (n, c, s);
        table obj1 = new table (n, c, koo);
        try {
            obj.t.join();
            obj1.t.join();
        } catch (Exception e)
        {
            System.out.println ("exception caught");
        }
    }
}
```

class table implements Runnable

{

int n, tab;

Thread t;

cal tar;

table (int a, cal c, int tab)

{

tab = tab

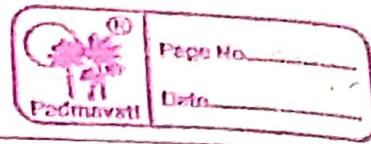
tar = c

this.n = n;

t = new Thread (this);

t.start();

}



```
public void run()
{
    synchronized (tar) {
        tar.cals(n, tab1);
    }
}
Class cal {
    void cals(int n, int ta)
    {
        for (int i = 1; i < n; i++)
            System.out.println (ta + "x" + i + "=" + (ta * i));
    }
}
```

Input the number of elements

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

maximum height

16/1/2021

Applet

```
10) import java.awt.*;
import javax.swing.event.*;
import java.awt.*;
import java.applet.*;
class division extends Applet implements
    ActionListener {
    String msg;
    Textfield num1, num2, res;
    Label l1, l2, l3;
    Button div;
    public void init() {
        l1 = new Label("Dividend");
        l2 = new Label("Divisor");
        l3 = new Label("Result");
        num1 = new TextField(10);
        num2 = new TextField(10);
        res = new TextField(10);
        div = new Button("divide");
        div.addActionListener(this);
        add(l1);
        add(num1);
        add(l2);
        add(num2);
        add(l3);
        add(res);
        add(div);
    }
```

```
public void actionPerformed(ActionEvent e)
```

```
{  
    String org = e.getActionCommand();
```

```
int num1=0, num2=0;
if (arg.equals ("click")) {
    if ((this, num1, getText(), is Empty() || this, num2,
        getText(), is Empty()))
        msg = "Enter the valid numbers";
    repaint();
}
else {
    try {
        num1 = integer.parseInt(this, num1, getText());
        num2 = integer.parseInt(this, num2, getText());
        int num3 = num1 / num2;
        res, setText (String Value of (num3));
        msg = "Operation Successful";
        repaint();
    }
    catch (NumberFormatException ex) {
        System.out.println (ex);
        res, setText ("");
        msg = "can't be divided by zero";
        repaint();
    }
}
public void paint (Graphics g) {
    g.drawString (msg, 30, 70);
}
```

