

Q) program to find roots of quadratic equation.

⇒ public class Quadratic

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

double a = 2.3 ;

double b = 4 ;

double c = -5.6 ;

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 imaginarypart = math.sqrt(-determinant) / (2 \* a) ;

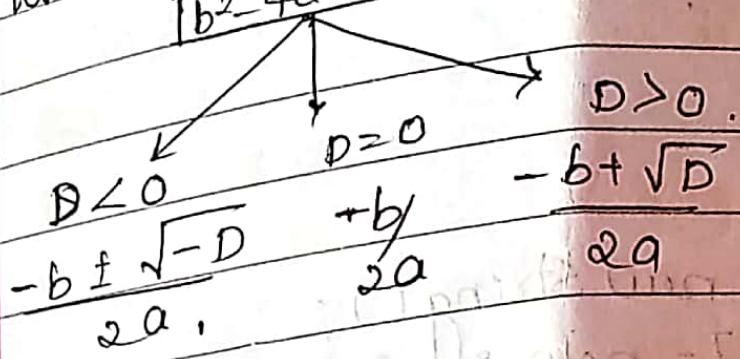
System.out.println ("The roots are complex and differ by sign.  
Real part = % 2f and Imaginary part = % 2f i", real part, imaginarypart);

`System.out.println ("root1 = " + .2f + "%i. 2fi)  
and root2 = " + .2f - "%i. 2fi, realpart,  
imaginary part, realpart, imaginarypart);`

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

where a, b, c are variables

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



Output :

$$\text{root1} = -0.87 + 1.30i \text{ and}$$

$$\text{root2} = -0.87 - 1.30i$$

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

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

### Lab program 2 :

Develop a java program to create a class Student with members, USN, Name, and marks & credits & an arraymarks. Include methods to accept & display details & 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:

SHREYA

USN:

24

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**

**7**

**Subject 4**

**7**

**Subject 5**

**8**

**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.lang.*;  
public class lab-program3{  
    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 D) array = new int [n];  
        for (int i=0, i<n; it+)  
    {
```

```
        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; it+)
```

{

```
        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 book1 & hwe

Enter the author of book1 olc

Enter the price of the book1 90.

Enter the number of pages of book1 900.

Enter the name of book2 swati

Enter the author of book2 or

Enter the price of book2 398

Enter the no.of pages of book2 600.

Displaying the details of the book1

Displaying the details of book2

Algol-

- initializing a class lab-program;
- initializing static variable name,author etc.
- taking input from the user using scanner
- initializing array to limit book's detail.
- & printing the details using for loop

```
3  
shmc  
ok  
90  
900  
SMA  
or.  
398  
600  
sak  
exx  
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 1  
8
```

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 contain only 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;

}

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()

{

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

}

}

{

circle (double a, double b) {  
super(a, b);

}

double area()

{

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

}

}

public class Abstractclass

{

public static void main (String args [])

{

rectangle r = new rectangle (9,5);

triangle t = new triangle (10,6);

circle c = new circle (4,4);

shape shape;

System.out.println ("area of " +

shape.area());

shapef = t;

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

shapef = c;

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

}

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(4,5);
        triangle t=new triangle(10,5);
        circle x=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=x;
        System.out.println("area of circle "+sharef.area());
    }
}

```

```

javac abstractareas.java
java -Djava.awt.headless=true abstractareas
inside area of rectangle
area of rectangle=10.0
inside area of triangle
area of triangle=25.0
inside area of circle
area of circle=50.0

```

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.next();

System.out.println ("enter Acc-num");

Acc num = s.next();

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

Acc-type = s.next();

}

Void display()

{

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

System.out.println ("Account number : " + Acc\_num);

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

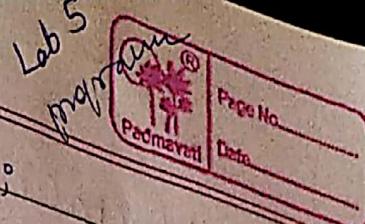
}

class curr\_acct extends Bank {

int updated\_balance;

int Afterwithdrawn;

int updated\_last\_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 (After_withdraw) <= (2000)
```

```
}
```

updated lost cbalance = (After\_withdraw) - 2000;  
System.out.println ("you have minimum balance below  
2000 so u have lost 200 rupees");

```
return updated_lost_cbalance;
```

```
}
```

```
else
```

```
return After_withdraw;
```

```
}
```

```
class sal_acct extends bank
```

```
{
```

```
int updated_balance;
```

```
int After_withdraw;
```

```
int updated_lost_sbalance;
```

```
int compound_interest;
```

```
int sdepo bal()
{
    updated balance = Balance + deposit balance;
    return updated balance;
}

int interest()
{
    double r = 0.08;
    int n = 12;
    int t = 5;
    compound interest = (int)((updated balance) * (math.pow(1 + (r/n)), (n*t)));
    return compound interest;
}
```

```
int swith bal()
{
    after withdrawn = (compound interest) - withdrawl;
    return after withdrawn;
}
```

```
int minimum()
{
    if (after withdrawn) <= (1000)
    {
        updated lost sbalance = (after swithdrawn) - (100);
        return updated lost sbalance;
    }
    else
    {
        return after swithdrawn;
    }
}
```

```
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  
            RS - " + ca.edepo_bal());  
        System.out.println ("enter money to be withdrawn");  
        ca.withdrawn_balance = r.nextInt();  
        System.out.println ("After your withdrawal of "  
            ca.withdrawn_balance + " now your total balance  
            is RS - " + ca.rwdepobal());  
        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 "  
            deposit_balance + " now your total balance  
            RS - " + SA.edepo_bal());  
        System.out.println ("After interest or updated balance  
            is RS - " + SA.interestr());
```

System.out.println ("Entered 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));

Problems • Issues • Declaration  Console

terminated> Transactions [Java Application] C:\Users\shivamshu\Downloads\javaprojects\javaprojects\src\main\java\com\javaproj\17\_03\_04\_1402\Transactions.java

Enter the customer's name

shivamshu

Enter the Account Number

111

Enter the Account type

current

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

1000

CUSTOMER NAME : shivamshu

ACCOUNT NUMBER : 111

ACCOUNT TYPE : current

After your deposition of 1000

Now your total balance is RS-20000

Enter the money you want to withdraw in rupees

500

After your withdrawal of 500

Now your total balance is RS-15000

After checking if u have minimum balance are not your updated total balance is RS-20000

Enter the customer's name

shivamshu

Enter the Account Number

111

Enter the Account type

savings

Enter the money u want to deposit in savings account

10000

CUSTOMER NAME : shivamshu

ACCOUNT NUMBER : 111

ACCOUNT TYPE : savings

After your deposition of 10000

Now your total balance is RS-17000

After interest ur updated balance is RS-16116

Enter the money you want to withdraw in Savings account

316

After your withdrawal of RS-316

Now your total balance is RS-157934