

Q. Develop a Java program that prints all real solutions to the quadratic equation $ax^2 + bx + c = 0$. Read in a, b, c and use the quadratic formula. If the discriminant, $b^2 - 4ac$ is negative, display a message stating that there are no real solutions.

```
import java.util.*;  
public class Main {  
    public static void main (String [] args) {  
        double a, b, c;  
        double root1, root2, discriminant;  
        Scanner sc = new Scanner (System.in);  
        System.out.print ("Enter co-Efficients a, b, c : \n");  
        a = sc.nextDouble();  
        b = sc.nextDouble();  
        c = sc.nextDouble();  
        discriminant = (b * b) - (4 * a * c);  
        if (discriminant > 0)  
            root1 = (-b + Math.sqrt(discriminant)) / (2 * a);  
            root2 = (-b - Math.sqrt(discriminant)) / (2 * a);  
            System.out.println ("Two distinct real roots are :  
                root1 = %.4f and root2 = %.4f", root1, root2);  
        else if (discriminant == 0)  
            root1 = & root2 = -b / (2 * a);  
            System.out.printf ("Two equal roots : %.4f", root1);  
    }  
}
```

else if (discriminant < 0) {
 cout << " Roots are not real";
}
else {
 cout << " Roots are real";
}

ENTER CO-EFFICIENTS a,b,c :

1
4
4

TWO EQUAL ROOTS: -2.0000

... Program finished with exit code 0

Press ENTER to exit console. □

```
import java.util.*;  
class Student {  
    private String usn;  
    private String name;  
    private int cred[];  
    private int marks[];  
    private int n;  
    void accept()  
{
```

```
Scanner s = new Scanner (System.in);  
System.out.println ("Enter Student details");  
System.out.println (" USN of Student:");  
usn = s.next();  
System.out.println ("Name of student:");  
name = s.next();  
System.out.println ("Enter the number of subjects:");  
n = s.nextInt();
```

```
cred = new int[n];  
marks = new int[n];  
System.out.println ("Enter credits and marks attained  
by the student in each subject out of 100");  
for (int i=0; i<n; i++)  
{
```

```
    cred[i] = s.nextInt();  
    marks[i] = s.nextInt();  
}
```

```
}  
void display()  
{
```

```
System.out.println ("Student details:");  
System.out.println ("USN : " + usn);
```

```
System.out.println ("Name : " + name);  
System.out.println ("Marks in each subject : ");  
for (int i = 0; i < n; i++)  
{  
    System.out.println ("Subject " + (i + 1) + ":" + marks[i]);  
}
```

```
double calculate()
```

```
{
```

```
int tp = 0, tc = 0;  
for (int i = 0; i < n; i++)  
{
```

```
    tc = tc + cred[i];
```

```
    if (marks[i] >= 50)  
{
```

```
        tch = tch + (((marks[i] / 10) + 1) * cred[i]);  
    }
```

```
else if (marks[i] >= 40 && marks[i] < 50)  
{
```

```
    tch = tch + (5 * cred[i]);  
}
```

```
return (double) tch / tc;
```

```
}
```

```
class Main
```

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

```
    Student s1 = new Student();
```

```
    s1.accept ();
```

```
    s1.display ();
```

Date _____
Page _____

System.out.println ("SGPA : " + s1.calculate());

3

KETAN

Enter the number of subjects:

5

Enter credits and marks attained by the student in each subject (out of 100)

5

90

4

80

4

78

3

40

4

86

Student details:

USN: 18M19C9076

Name: KETAN

Marks in each subject:

Subject 1: 90

Subject 2: 80

Subject 3: 78

Subject 4: 40

Subject 5: 86

SGPA: 8.3

Program finished with exit code 0

Lab Program 3

Create a class Book which contains four members: name, author, price, num - pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a `toString()` method that could display the complete details of the book. Develop a Java program to create a book objects.

```

import java.util.*;
import java.lang.*; // for Scanner
class Main {
    String name;
    String author;
    double price;
    int num - pages;
    void getdetails() {
        Scanner x = new Scanner (System.in);
        System.out.println ("Enter book Details");
        System.out.println ("Enter book name:");
        name = x.next();
        System.out.println ("Enter the Author");
        author = x.next();
        System.out.println ("Enter the price of the book");
        num - pages price = x.nextDouble();
        System.out.println ("Enter the number of pages in
                           the book");
    }
}
  
```

num_pages = x.nextInt();

{

public String toString()

{

```
return ("BOOK DETAILS* " + "\n The name of the book: " + name + "\n The author of the book: " + author + "\n The price of the book: " + price + "\n Number of pages in book: " + num_pages);
```

{

{

class Main

{

public static void main(String [] args)

{

int i, n;

Scanner x = new Scanner (System.in);

System.out.println ("Enter the number of books");

n = x.nextInt();

Main1 b[] = new Main1[n];

for (i=0; i<n; i++)

{

b[i] = new Main1();

b[i].getDetails();

3. (Scanner input.nextLine());

for (i=0; i<n; i++)

3. (System.out.println(b[i]));

3. (System.out.println("Total price of books is " + sum));

3. (sum = sum + b[i].price);

3. (sum = sum + b[i].price);

Enter the number of books

1

Enter Book Details

Enter Book Name:

java

Enter the Author

reemathareja

Enter the Price of the book

680

Enter the number of pages in the book

1058

BOOK DETAILS*

The name of the book : java

The author of the book : reemathareja

The price of the book: 680.0

Number of pages in book :1058

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

import java.util.Scanner;
abstract class Shape

{

int d1;

int d2;

Shape (int a, int b)

{

d1 = a;

d2 = b;

{

abstract void printArea();

{

class Rectangle extends Shape

{

Rectangle (int a, int b)

{

super (a, b);

{

void printArea()

{

double area = d1 * d2;

System.out.println ("Area of the rectangle : "+area);

{

class Triangle extends Shape

{

Triangle (int a, int b)

{

super (a, b);

{

void printarea()

{

double area = d1 * d2 / 2;

System.out.println ("Area of the triangle : "+ar

{

{

class Circle extends Shape

{

Circle (int a, int b)

{

super (a, b);

{

void printarea()

{

double area = 3.14 * d1 * d1;

System.out.println ("Area of the circle : "+area);

{

{

class Main

{

public static void main (String args [])

{

System.out.println("Enter the dimensions of rectangle");
Scanner sc = new Scanner(System.in);
int x = sc.nextInt();
int y = sc.nextInt();
System.out.println("Enter dimensions of triangle");
int s = sc.nextInt();
int w = sc.nextInt();
System.out.println("Enter the radius of circle");
int f = sc.nextInt();
Rectangle r = new Rectangle(x, y);
Triangle t = new Triangle(s, w);
Circle c = new Circle(f, f);
r.printarea();
t.printarea();
c.printarea();

Enter the dimensions of rectangle

10

30

Enter the dimensions of triangle

9

8

Enter the radius of circle

5

Area of the rectangle :300.0

Area of the triangle :36.0

Area of the circle :78.5

...Program finished with exit code 0

Press ENTER to exit console.

ecut....pdf



lab 2 output (1).pdf



1B

here to search



Lab-5

Develop a Java program to create 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 withdraw facilities but no cheque book facility.

import java.util.Scanner;
abstract class Account

{

String cust_name;

long acc_no;

String acc_type;

double balance;

double min_bal = 1000.0;

Account (String cust_name, long acc_no, String acc_type, double balance)

{

this.cust_name = cust_name;

this.acc_no = acc_no;

this.acc_type = acc_type;

this.balance = balance;

}

abstract void deposit (double amount);

abstract void display ();

abstract void withdraw (double amount);

{

class Curr_acct extends Account

{

double penalty = 100.0;

curr_act (String cust_name, long acc_no,
String acc_type, double balance)

{
super(cust_name, acc_no, acc_type, balance);
System.out.println ("Name of the customer: ");

System.out.println ("Account Number: " + acc_no);

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

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

void deposit (double amount)

{
this.balance = this.balance + amount;

}

void withdraw (double amount)

{
this.balance = this.balance - amount;

impose penalty();

System.out.println ("The current balance is " + balance);
}

void imposePenalty ()

{
if (this.balance < min_bal) {

if (this.balance < min_bal) {
}

this.balance = this.balance - penalty;

System.out.println ("The balance amount is
insufficient, the penalty imposed= 100Ru");
}

void display()

{
System.out.println ("Customer Name: " + cust_name);
System.out.println ("Account Number: " + acc_no);
System.out.println ("Account Type: " + acc_type);
System.out.println ("Balance: " + balance);
}

```
System.out.println("Balance is :" + this.balance);
```

```
}
```

```
class Sav acct extends Account
```

```
{
```

```
Sav_acct (String cust_name, long acc_no,  
String acc_type, double balance)
```

```
{
```

```
super(cust_name, acc_no, acc_type, balance);
```

```
System.out.println ("Name of the customer :" + cust_name);
```

```
System.out.println ("Account Number :" + acc_no);
```

```
System.out.println ("Account Type : " + acc_type);
```

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

```
}
```

```
void deposit (double amount)
```

```
{
```

```
this.balance = this.balance + amount;
```

```
interest();
```

```
}
```

```
void interest ()
```

```
{
```

```
intrate = 10, time = 1;
```

```
float ci = (float)
```

```
(this.balance * Math. pow((1+rate)/100.0, time) -  
(1+rate) * this.balance);
```

```
System.out.println ("The interest amount added to  
balance is " + ci);
```

```
this.balance = this.balance + ci;
```

```
}
```

```
void withdraw (double amount)
```

this.balance = this.balance - amount;

 System.out.println ("The current balance is " + balance);

 void display ()

 System.out.println ("Balance is " + this.balance);

}

class AccountMain

 public static void main (String [] args)

 Scanner xx = new Scanner (System.in);

 Double amount;

 int flag = 0;

 while (flag == 0)

 System.out.println ("Enter the type of Account : Int:");

 Current account 1 or 2: Savings account);

 int choice = xx.nextInt();

 switch (choice)

 case 1: System.out.println ("1 is Current account : ");

 System.out.println ("Enter the name of account holder");

 String f = xx.next();

 System.out.println ("Enter the account number");

 long g = xx.nextLong();

 System.out.println ("Enter the balance amount");

 double h = xx.nextDouble();

Curv -> act c = new Curv. acc (f, g, "current", h);
int flag1 = 0;
while (flag1 == 0)
{

System.out.println ("Enter your choice.\n1: Deposit
amount\n2: Display Balance\n3: Withdraw");
int choice1 = xx.nextInt();
switch (choice1)

{

case 1:

System.out.println ("Enter amount to be deposited");
amount = xx.nextDouble();
c.deposit(amount);
break;

case 2:

c.display();

break;

case 3:

System.out.println ("Enter amount you want to
withdraw");

amount = xx.nextDouble();

c.withdrawl(amount);

break;

default:

flag1 = 1;

}

}

break;

case 2 : System.out.println ("In Savings Account:\n")

System.out.println ("Enter name of account holder");

String h = xx.nextLine();

```
long long q = xx. nextLong();  
System.out.println("Enter the balance amount");  
double r1 = xx. nextDouble();  
New account = new NewAccount(p, q, "Savings", r1);  
int flag2 = 0;  
while (flag2 == 0)
```

{

```
System.out.println("Enter your choice. 1 n 1: Deposit  
amount | n 2: Display Balance | n 3: Withdraw");
```

```
int choice2 = xx. nextInt();
```

```
switch (choice2) {
```

```
case 1: System.out.println("Enter amount to be  
deposited:");
```

```
amount = xx. nextDouble();
```

```
break;
```

```
case 2:
```

```
s. display();
```

```
break;
```

```
case 3:
```

```
System.out.println("Enter amount you want to  
withdraw:");
```

```
amount = xx. nextDouble();
```

```
s. withdraw(amount);
```

```
break;
```

```
default:
```

```
flag2 = 1;
```

```
3
```

```
break;
```

```
default: flag = 1;
```

```
3
```

```
3
```

```
enter customer's account type 1.savings account 2.current account
1
-----enter account details-----
enter customer name
karthik
enter customer account number
34576897
enter customer's account type 1.savings account 2.current account
1
enter customer's balance amount in account
30000
-----customer's account details-----
customer name karthik
customer account number 34576897
customer's account type 1
customer's balance amount in account 30000.0
enter 1.deposit 2.withdrawal
1
enter amount to be deposited
560
enter rate and time period
```



1

enter customer's balance amount in account

30000

-----customer's account details-----

customer name karthik

customer account number 34576897

customer's account type 1

customer's balance amount in account 30000.0

enter 1.deposit 2.withdrawal

1

enter amount to be deposited

560

enter rate and time period

2

2

compound interest - 1234.6239999999998

customer's balance amount in account 31794.624

...Program finished with exit code 0

Press ENTER to exit console.

Lab-6

```
package CTF;
import java.util.*;
public class Student
{
    public String usn, name;
    public int sem;
    public void input()
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("---- enter student details ---");
        System.out.println("name:");
        name = sc.nextLine();
        System.out.print("usn:");
        usn = sc.nextLine();
        System.out.print("sem:");
        sem = sc.nextInt();
        System.out.println();
    }
    public void display()
    {
        System.out.println("---- student details ----");
        System.out.println("name:" + name);
        System.out.println("usn:" + usn);
        System.out.println("sem:" + sem);
    }
}
```

```
package CIF;
import java.util.*;
public class Internal
```

```
{  
    public int cie_marks [] = new int [5];
    public void input()
```

```
{  
    Scanner sc = new Scanner (System.in);
    System.out.println ("Enter CIE marks in 5 courses");
    for (int i=0; i<5; i++)
        cie_marks [i] = sc.nextInt();
```

```
}  
public void display()
```

```
{  
    System.out.println ("CIE marks : ");
    for (int i=0; i<5; i++)
        System.out.print (cie_marks [i] + " ");
    System.out.println ();
```

```
package CIF;
```

```
import CIF.*;
import java.util.*;
public class External CIF.Student
```

```
{  
    public int sec_marks [] = new int [5];
    public void input()
```

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

Scanner sc ("enter see marks in Scowser");

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

see_marks[i] = sc.nextInt();

}

public void display()

{

System.out.println("See marks:");

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

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

System.out.println();

}

import CIE.*;

import SFF.*;

import java.util.*;

class main

{

int final_marks[] = new int[5];

public static void main (String args [])

{

Scanner sc = new Scanner (System.in);

System.out.println("Enter no. of students");

int n = sc.nextInt();

CIE.Student[] o1 = new CIE.Student[n];

CIE.Internal[] o2 = new CIE.Internal[n];

SFF.External[] o3 = new SFF.External[n];

main[] obj = new main[n];

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

{

o1[i] = new CIE.Student();

o2[i] = new CIE.Internal();

```

o3[i] = new SFE.External
obj[i] = new main();
o1[i].input();
o2[i].input();
o3[i].input();
for (int j = 0; j < 5; j++)
    obj[i].final_marks[j] = o2[i].marks[j] + (o3[i].se
                                                marks[j])
{
    o1[i].display();
    o2[i].display();
    o3[i].display();
}
System.out.println("Final marks in 5 courses");
for (int j = 0; j < 5; j++)
    System.out.print(obj[i].final_marks[j] + " ");
System.out.println();
}

```

3

(i) If a method is forward and has no return
 statement, it is called a "function", which
 does not return any value.
 (ii) If a method has a return value, it is called
 a "function with a return value".
 Note that a function = a global function (AF)
 A function AF can be used in another AF.
 (i) Function = obj / function
 (ii) If a function uses other functions

(i) Function AF can be used in another AF.

(ii) Global AF can be used in another AF.

name : neha

usn : 123

sem : 3

enter cie marks in 5 courses :

45 40 45 40 50

enter see marks in 5 courses :

90 95 90 95 100

---enter student details---

name : nikhil

usn : 456

sem : 3

enter cie marks in 5 courses :

40 40 50 40 40

enter see marks in 5 courses :

80 80 100 80 80

---student details---

name : neha

usn : 123

sem : 3

cie marks :

45 40 45 40 50

see marks :

90 95 90 95 100



11-120

Week 10Lab 7

generics.

- 7) Write a program to demonstrate, with multiple object parameters.

```
class Generics < T, U, S >
```

{

```
    T obj1,
```

```
    U obj2;
```

```
    S obj3;
```

```
Generics (T obj1, U obj2, S obj3)
```

{

```
    this.obj1 = obj1;
```

```
    this.obj2 = obj2;
```

```
    this.obj3 = obj3;
```

{

```
public void print()
```

{

```
    System.out.println(obj1);
```

```
    System.out.println(obj2);
```

```
    System.out.println(obj3);
```

{

{

```
class Main
```

{

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

{

```
    Generics < String, Integer, String > obj =
```

```
        new Generics < String, Integer, String >
```

```
        ("Week", 7, "Lab-Program")
```

```
    obj.print();
```

{

WEEK

7

LAB-PROGRAM

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

6 (2).docx



lab 4 and 5 output....pdf



bookdata

Lab 8

Write a program that demonstrates handling of exceptions 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 and throws the exception ~~WrongAge()~~ WrongAge() when the input age = father's age.

```
import java.util.*;  
class FatherAgeException extends Exception {  
    public String toString() {  
        return ("Father's age is less than 0");  
    }  
}
```

```
class SonAgeException extends Exception {  
    int a;  
    SonAgeException (int age) {  
        a = age;  
    }  
    public String toString() {  
        if (a < 0)  
            return ("Son's age is less than 0");  
        else  
            return ("Son's age is more than or equal to  
                    father's age");  
    }  
}
```

```
class Father {  
    public int age;  
}
```

Scanner s = new Scanner(System.in);
father () {

System.out.print("Enter father's age: ");
age1 = s.nextInt();

void exc1 () throws FatherAgeException {

if (age1 < 0)

throw new FatherAgeException();

class son extends father {

public int age2;

son () {

System.out.print("Enter son's age: ");

age2 = s.nextInt();

void exc2 () throws SonAgeException {

if (age2 > 0 || age2 >= super.age1)

throw new SonAgeException(age2);

System.out.println("No logical errors in the
entered data");

class Main {

public static void main (String args []) {

son s = new son();

try {

s.exc1();

catch (FatherAgeException e) {

System.out.println(e);

try {

s.exc2();

}

catch (sonAgeException e) {

System.out.println(e);

}

}

}

Enter father's age: 60
Enter son's age: 58
No logical error in the entered data

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

Lab - 9

Write a program which creates two threads, one thread displaying "BMS College Of Engineering" once every ten seconds and another displaying "CSE" once every two seconds.

class NewThread implements Runnable

{

String name;

String identity;

Thread t;

long time;

NewThread (String threadname, long time)

{

time = time; 1;

name = threadname;

t = new Thread (this, name);

System.out.println ("First Thread" + t);

t.start();

{

public void run ()

{

try

{

for (int i=0; i<20; i++)

{

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

Thread.sleep (time);

{

catch (InterruptedException e)

{

System.out.println(name + " Interrupted");

System.out.println(name + " exiting.");

{

class Main {

public static void main (String args [])

new NewThread ("B.M.S. College of Engineering", 10000),

new NewThread ("CSE", 2000);

First Thread Thread[BMS College of Engineering, 5, main]

First Thread Thread[CSE, 5, main]

BMS College of Engineering

CSE

CSE

CSE

CSE

CSE

BMS College of Engineering

CSE

CSE

CSE

CSE

BMS College of Engineering

CSE

CSE

CSE

CSE

CSE

BMS College of Engineering

CSE

CSE

CSE

- GDB

H!

Lab program 10

Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, 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 number format exception. If num2 were zero, the program would throw an ArithmeticException. Display the exception in a message dialog box.

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class IntegerDivision extends Frame implements
ActionListener
```

{

```
TextField n1, n2, res;
```

```
Label lm1, lm2, lres;
```

```
Button b;
```

```
public IntegerDivision()
```

{

```
setLayout(new FlowLayout());
```

```
Label lm1 = new Label("Number 1", Label.RIGHT);
```

```
Label lm2 = new Label("Number 2", Label.RIGHT);
```

```
Label lres = new Label("RESULT", Label.RIGHT);
```

```
n1 = new TextField(12);
```

```
n2 = new TextField(8);
```

```
res = new JTextField(10);
b = new JButton("DIVISION");
add(lm1);
add(n1);
add(m2);
add(n2);
add(b);
add(brs);
add(res);
b.addActionListener(this);
addWindowListener(new WindowAdapter() {
    }
```

```
    public void actionPerformed(ActionEvent e) {
        if (ae.getSource() == b)
```

```
            try {
                int num1 = Integer.parseInt(n1.getText());
                int num2 = Integer.parseInt(n2.getText());
                int num3 = num1 / num2;
                res.setText(String.valueOf(num3));
            }
```

```
        catch (NumberFormatException e) {
            JOptionPane.showMessageDialog(this, "DIVISION BY ZERO ERROR", "ERROR", JOptionPane.ERROR_MESSAGE);
        }
```

```
        catch (ArithmaticException e) {
            JOptionPane.showMessageDialog(this, "DIVISION BY ZERO ERROR", "ERROR", JOptionPane.ERROR_MESSAGE);
        }
```

```
    }
```

{
3
3
3public static void main (String args []) {
}integerdivision i = new integerdivision ();
i.setSize (new Dimension (400,400));
i.setTitle ("INTEGER DIVISION OF TWO NUMBERS");
i.setVisible (true);
}class WindowAdapter1 extends WindowAdapter
{public void windowClosing (WindowEvent we) {
}System.exit(0);
}

3

3 An application window - window

3 A window frame - window

3 window frame - window



INTEGER DIVISION OF TWO NUMBERS

-



NUMBER 1

NUMBER 2

DIVISION

RESULT



INTEGER DIVISION OF TWO NUMBERS

-



NUMBER 1

NUMBER 2

DIVISION

RESULT

ERROR



java.lang.NumberFormatException: For input string: "20.5"

OK



INTEGER DIVISION OF TWO NUMBERS



NUMBER 1

NUMBER 2

DIVISION

RESULT

DIVISION BY ZERO ERROR



java.lang.ArithmeticException: / by zero

OK