

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 equation formula. If the discriminant is  $b^2-4ac$  is negative, display a message stating that there are no real solutions.

```
import java.util.Scanner;
class Quad
{
    public static void main(String args[])
    {
        Scanner s1 = new Scanner(System.in);
        System.out.println("Enter the co-efficients of the
                           quadratic equation");
        double a = s1.nextDouble();
        double b = s1.nextDouble();
        double c = s1.nextDouble();
        double d = b*b - 4*a*c;
        if (d > 0)
        {
            System.out.println("Roots are real and distinct");
            System.out.println("Roots: " + "In r1 = " + (-b + Math.sqrt(d)) / (2*a));
            System.out.println("Roots: " + "In r2 = " + (-b - Math.sqrt(d)) / (2*a));
        }
        else if (d == 0)
        {
            System.out.println("Roots are real and equal");
            System.out.println("Roots: " + "In r1 = r2 = " + (-b / 2*a));
        }
    }
}
```

```

else
{
    System.out.println("Roots are real and imaginary");
    double r1 = (-b/2*a);
    double r2 = Math.abs(d);
    System.out.println("Roots:" + "ln r1=" + r1 + "+" + r2 + "i");
    System.out.println("Roots:" + "ln r2=" + r1 + "-" + r2 + "i");
}
}
}

```

Output:

javac l.java

> java Quad

Enter the co-efficients of the quadratic equation

1

2

1

Roots are real and equal

Roots:

$$r_1 = r_2 = -1.0$$

> java Quad

Enter the co-efficients of the quadratic equation

6

8

0

Roots are real and distinct.

Roots:

$$r_1 = 0.0000 \quad r_2 = -1.3333$$

> java Quad

Enter the co-efficients of the quadratic equation

2

3

4

Roots are real and imaginary

Roots:

$$\gamma_1 = -1.0 + 8.0i$$

$$\gamma_2 = -1.0 - 8.0i$$

Write a Java program to create a class Employee with members empid, empname, empnohrs, empbasic, emphra(%), empda(%), empit(%), empgross

```
import java.util.Scanner;  
Class Employee  
{  
    public static void main(String args[])  
    {  
        Scanner s1 = new Scanner(System.in);  
        int empid = s1.nextInt();  
        char empname = s1.nextChar();  
        float empnohrs = s1.nextFloat();  
        double emphra = s1.nextDouble();  
        double empbasic = s1.nextDouble();  
        double empit = s1.nextDouble();  
        double empda = s1.nextDouble();  
  
        System.out.println("Enter Employee's ID:");  
        System.out.println("In Employee ID: " + empid);  
        System.out.println("Enter Employee's name.");  
        System.out.println("In Employee's name: " + empname);  
        System.out.println("Enter  
        System.out.println("In EmployeeNo.of hours: ");  
        System.out.println("No.of hours: " + empnohrs);  
        System.out.println("In Enter emphra");  
        System.out.println("Enter emphra: " + emphra);  
        System.out.println("Enter empda: ");  
        System.out.println("empda: " + empda);  
        System.out.println("Enter employee's it: ");  
        System.out.println("Employee's it: " + empit);  
        System.out.println("Enter employee's basic: ");  
        System.out.println("Employee's basic: " + empbasic);  
    }  
}
```

```

empgross = new Double();
empgross = empbasic + empbasic * emphra + empbasic *
            empda - empbasic * empit;
System.out.println("The employee's gross salary :" + tempgross);

if(emphohrs > 200) {
    empgross += (emphohrs - 200) * 100;
    System.out.println("The employee's gross salary :" + tempgross);
} else {
    empgross += (200 - emphohrs) * 100;
    System.out.println("The employee's gross salary :" + empgross);
}

```

Output:

Enter Employee's ID:

Employee ID: 001

Enter Employee's name:

Employee's name: Rinku

Enter no. of hours:

No. of hours: 250

Enter emphra:

emphra: 16

Enter empda:

empda: 17

Enter employee's it:

Employee's it: 18

Enter employee's basic salary:

Employee's basic salary: 70000

The employee's gross salary: 85500.0

The employee's gross salary: 90500.0

\* Develop a java program to create a class Student with members usn, name, an array credits and an array marks. Include methods accept & display details and method to calculate SGPA of a student.

```
import java.util.Scanner;
```

```
class Student {
```

```
    int USN;
```

```
    String Name;
```

```
    int credits[];
```

```
    int marks[];
```

```
    int subcou;
```

```
    Scanner s1 = new Scanner(System.in);
```

```
    void accept()
```

```
{
```

```
    System.out.println("Enter the USN of the student:");
```

```
    USN = s1.nextInt();
```

```
    System.out.println("Enter the name of the student:");
```

```
    Name = s1.next();
```

```
    credits = new int[subcou];
```

```
    marks = new int[subcou];
```

```
    for (int i = 0; i < subcou; i++)
```

```
{
```

```
    System.out.println("Enter the credits of each sub:");
```

```
    credits[i] = s1.nextInt();
```

```
    System.out.println("Enter the marks of each subject");
```

```
    marks[i] = s1.nextInt();
```

```
}
```

```
}
```

```
void display()
```

```
{
```

System.out.println("USN:" + USN);  
System.out.println("Name:" + Name);  
for (int i=0; i < subcou; i++)  
{  
 System.out.println("Marks are:" + marks[i]);  
}  
  
float gradcalc(int marks)  
{  
 if (marks >= 90) {  
 return 10;  
 }  
 if (marks >= 80 && marks < 90) {  
 return 9;  
 }  
 if (marks >= 70 && marks < 80) {  
 return 8;  
 }  
 if (marks >= 60 && marks < 70) {  
 return 7;  
 }  
 if (marks >= 50 && marks < 60) {  
 return 6;  
 }  
 if (marks < 50) {  
 return 5;  
 }  
 return 0;  
}

```
float calc() {
    int totalgradepoints=0;
    int totalcredits=0;
    float sgpa;
    for(int i=0; i<subcou; i++) {
        totalgradepoints+=grade calc(marks[:,i])*credits[i];
        totalcredits+=credits[i];
    }
    return (float)totalgradepoints/total credits;
}
```

```
public static void main(String args[]) {
    Student s2 = new Student();
    s2.accept();
    s2.display();
    System.out.println("The SGPA of the student: "
        + s2.calc());
}
```

Output:

Enter the USN of student:

138

Enter the name of the student:

Raja

Enter no of subjects:

4

Enter the credits of each subject:

4

Enter the marks of each subject:

92

Enter the credits of each subject:

3

Enter the marks of each subject:

89

Enter the credits of each subject:

3

Enter the marks of each subject:

78

Enter the credits of each subject:

2

Enter the marks of each subject:

95

USN: 138

Name: Raja

Marks are: 92

Marks are: 89

Marks are: 78

Marks are: 95

The SGPA of the student is: 9.25

111124

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 object using toString() method for n books.

```
import java.util.Scanner;  
class Book  
{  
    String name;  
    String author;  
    double price;  
    int no-pages;  
    Book(String name, String author, double price,  
        int noPages) {  
        this.name = name;  
        this.author = author;  
        this.price = price;  
        this.no-pages = no-pages;  
    }  
    String getName() {  
        return name;  
    }  
    String getAuthor() {  
        return author;  
    }  
    double getPrice() {  
        return price;  
    }  
}
```

```
double int "no-pages" {  
    get  
    return no-pages;  
}  
  
void setName(String name) {  
    this.name = name;  
}  
  
void setAuthor(String author) {  
    this.author = author;  
}  
  
void set Price(double price) {  
    this.price = price;  
}  
  
void set No pages(int no-pages) {  
    this.no-pages = no-pages;  
}  
  
public String toString()  
{  
    return "Name:" + name + "Author:" + author + "Price :" +  
        price + "No-pages:" + no-pages;  
}  
  
public static void main(String args[])  
{  
    Scanner sc = new Scanner(System.in);  
    System.out.println("Enter the number of books:");  
    int n = sc.nextInt();  
    Book[] books = new Book[n];  
    for (int i = 0; i < n; i++)  
    {  
        System.out.println("Enter the details for book" + (i + 1));  
        System.out.println("Enter name");  
        String name = sc.next();  
        books[i] = new Book(name);  
        System.out.println("Enter author");  
        String author = sc.next();  
        books[i].setAuthor(author);  
        System.out.println("Enter price");  
        double price = sc.nextDouble();  
        books[i].set Price(price);  
        System.out.println("Enter no-pages");  
        int no-pages = sc.nextInt();  
        books[i].set No pages(no-pages);  
    }  
    for (int i = 0; i < n; i++)  
    {  
        System.out.println(books[i]);  
    }  
}
```

```
System.out.println("Enter author:");
String author = sc.next();
System.out.println("Enter price:");
double price = sc.nextDouble();
System.out.println("Enter no of pages:");
int no_pages = sc.nextInt();
books[i] = new Book(name, author, price,
no_pages);
}
for (int i=0, i<n; i++)
{
    System.out.println("Details of book " + (i+1));
    System.out.println(books[i].toString());
}
}
```

Output:

Enter the number of books:

2

Enter details for book 1:

Enter name:

Jay

Enter author:

Fury

Enter price:

1000

Enter no of pages:

1132

Enter details for book 2:

Enter name:

Farhan

Enter author:

Nick

Enter price:

1250

Enter no of pages:

750

Details of Book 1:

Name: Jay

Author: Fury

Price: 1000

No. of pages: 1132

Details of Book 2:

Name: Farhan

Author: Nick

Price: 1250

No. of pages: 750

✓  
① 1124

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 class contains only one method printArea() that prints the area of the given shape.

```
import java.util.Scanner;  
abstract class Shape {  
    int len;  
    int bre;  
    abstract void printArea();  
}  
Class Rectangle extends Shape {  
    Rectangle(int l, int br) {  
        len = l;  
        bre = br;  
    }  
    void printArea() {  
        int area = len * bre;  
        System.out.println("The area of rectangle is: " + area);  
    }  
}  
class Triangle extends Shape {  
    Triangle(int b, int h) {  
        len = b;  
        bre = h;  
    }  
    void printArea() {  
        double area = 0.5 * len * bre;  
    }  
}
```

```
System.out.println("The area of triangle is: "+area);
}
}

class Circle extends Shape {
    Circle(int r) {
        len = r;
    }
    void printArea() {
        double area = 3.14*len*len;
        System.out.println("The area of circle is: "+area);
    }
}

class Areas {
    public static void main(String args[]) {
        Scanner s = new Scanner(System.in);
        Rectangle r = new Rectangle(s.nextInt(), s.nextInt());
        r.printArea();
        Triangle t = new Triangle(s.nextInt(), s.nextInt());
        t.printArea();
        Circle c = new Circle(s.nextInt());
        c.printArea();
    }
}
```

Output:

5

7

The area of rectangle is: 35

6

8

The area of triangle is: 24.0

6

The area of circle is  $113.0399$

~~11724.~~

Create a package CIE which has two classes - Student and Internals. The class Personal has members like usn, name, sem. The class internals has an array that stores the internal marks scored in five courses of the current semester of the student.

Create another package SEE which has the class External which is a derived class of Student. This class has an array that stores the SEE marks scored in five courses of the semester of the student.

```
import java.util.Scanner;  
package CIE;  
public class Student {  
    public int usn;  
    public String name;  
    public int sem;  
    public void accept() {  
        Scanner s1 = new Scanner(System.in);  
        System.out.println("Enter USN:");  
        usn = s1.nextInt();  
        System.out.println("Enter name:");  
        name = s1.next();  
        System.out.println("Enter semester:");  
        sem = s1.nextInt();  
    }  
    public void display() {  
        System.out.println("USN: " + usn + "Name: " +  
            name + "Semester: " + sem);  
    }  
}
```

```
package CIE;
import java.util.Scanner;
public class Internals extends Student {
    int marks[];
    public void acceptInternalMarks(int[] marks) {
        marks = new int[5];
        Scanner s1 = new Scanner(System.in);
        for (int i=0; i<5; i++) {
            System.out.println("Enter " + i + " subject marks:");
            marks[i] = s1.nextInt();
        }
    }
}
```

```
public void displayInternalMarks() {
    System.out.println("Internal Marks:");
    for (int i=0; i<5; i++) {
        System.out.println(marks[i]);
    }
}
```

```
package SEE;
import CIE.Student;
import java.util.Scanner;
public class Externals extends Student {
    int seeMarks[];
    public void acceptSeeMarks() {
        seeMarks = new int[5];
        Scanner s1 = new Scanner(System.in);
        for (int i=0; i<5; i++) {
            System.out.println("Enter subject marks:");
            seeMarks[i] = s1.nextInt();
        }
    }
}
```

```
    }  
    }  
  
    public void displaySeeMarks() {  
        System.out.println("SEE Marks:");  
        for (int i = 0; i < s; i++) {  
            System.out.println("seeMarks " + );  
        }  
    }
```

```
import CIE.*;  
import SEE.*;  
import java.util.Scanner;  
public class Demo {  
    public static void main(String[] args) {  
        int n;  
        Scanner s1 = new Scanner(System.in);  
        System.out.println("Enter no of students");  
        n = s1.nextInt();
```

```
    Internal[] internalArray = new Internal[n];  
    External[] externalArray = new External[n];  
    for (int i = 0; i < n; i++) {  
        internalArray[i] = accept(new Internal());  
        externalArray[i] = accept(new External());  
        internalArray[i].acceptInternalMarks(5);  
        System.out.println("Enter details for student " + (i + 1));  
        internalArray[i].accept();  
        internalArray[i].acceptInternalMarks();  
        externalArray[i].accept();  
        externalArray[i].acceptSeeMarks(5);  
    }
```

```
    System.out.println("In Details of students:");
```

```
for (int i=0; i<n; i++) {  
    System.out.println("Student "+(i+1));  
    internalsArray[i].display();  
    internalsArray[i].displayInternalMarks();  
    externalsArray[i].displaySecMarks();  
}
```

3

3

Output:

Enter the no of students:

2

Enter USN:

1

Enter Name:

Ram

Enter Semester:

3

Enter subject1 marks:

47

Enter subject 2 marks:

48

Enter subject 3 marks:

46

Enter subject 4 marks:

49

Enter subject 5 marks:

50

Enter USN:

2

Enter name:

Ramesh

Enter Semester :

3

Enter subject 1 marks:

48

Enter subject 2 marks:

49

Enter subject 3 marks:

50

Enter subject 4 marks:

50

Enter subject 5 marks:

49

Enter <sup>USN</sup> name :

I

Enter name:

Ram

Enter ~~as~~ semester:

3

Enter subject 1 marks:

48

Enter subject 2 marks:

49

Enter subject 3 marks:

50

Enter subject 4 marks:

50

~~Courses:~~ Enter subject 5 marks:

49

Enter USN:

2

Enter Name:

Ramesh

Enter semester:

3

Enter subject 1 marks:

47

Enter subject 2 marks:

49

Enter subject 3 marks:

49

Enter subject 4 marks:

50

Enter subject 5 marks:

48

Final Marks:

USN1, Name: Ram, Semester: 3, Final Marks:

95 97 96 99 99

USN: 2, Name: Ramesh, Semester: 3, Final Marks:

94 96 99 96 97

X  
201121

Write a program that demonstrates handling of exceptions in inheritance tree.

```
import java.util.Scanner;

class wrongAge extends Exception {
    public wrongAge(String message) {
        super(message);
    }
}

class Father {
    private int age;
    public Father(int age) throws wrongAge {
        if (age < 0) {
            throw new wrongAge("Age cannot be negative");
        }
        this.age = age;
    }
    public int getAge() {
        return age;
    }
}

class Son extends Father {
    private int sonAge;
    public Son(int fatherAge, int sonAge) throws wrongAge {
        super(fatherAge);
        if (sonAge >= fatherAge) {
            throw new wrongAge("Son's age should be less than Father's age");
        }
    }
}
```

```
this.sonAge = sonAge;
```

```
}
```

```
public int getSonAge() {
```

```
    return sonAge;
```

```
}
```

```
}
```

```
public class Exception {
```

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

```
        try {
```

```
            Father father = new Father(40);
```

```
            Son son = new Son(father.getAge(), 56);
```

```
            System.out.println("Father's age: " + father.
```

```
                .getAge());
```

```
            System.out.println("Son's age: " + son.getSonAge());
```

```
        } catch (WrongAgeException e) {
```

```
        }
```

```
        System.out.
```

```
            .println("Exception: " + e.getMessage());
```

```
}
```

```
}
```

Output:

Exception: Son's age should be less than  
Father's age.

Develop a java program to create a class Bank that maintains two kinds of account from its customer, one called savings and other current account.

```
import java.util.Scanner;

class Account {
    String customer_name;
    int account_no;
    String account_type;
    double balance;
    Account(String name, int number, String type) {
        customer_name = name;
        account_no = no;
        account_type = type;
        balance = 0.0;
    }
    void deposit(double amount) {
        balance += amount;
        System.out.println("Updated balance : " + balance);
    }
    void display() {
        System.out.println("Current balance : " + balance);
    }
}

class sav_acct extends Account {
    double rate;
    sav_acct(String name, int number) {
        super(name, number);
        rate = 0.05;
    }
}
```

```
}

void compute-interest() {
    double interest = balance * rate;
    deposit(interest);
    System.out.println("Updated balance: "+balance);
}

void withdraw(double amount) {
    if(balance >= amount) {
        balance -= amount;
        System.out.println("Updated balance: "+balance);
    } else {
        System.out.println("Insufficient funds for withdrawal");
    }
}

class curr_acct extends Account {
    double min-balance;
    double service-charge;
    curr-acct(String name,int number) {
        super(name,number);
        min-balance=1000;
        service-charge=20.0;
    }

    void service-charge() {
        if(balance < min-balance) {
            balance -= service-charge;
        }
        System.out.println("Updated balance: "+balance);
    }
}
```

```
void withdraw(double amount){  
    if (balance >= amount){  
        balance -= amount;  
        System.out.println("Updated balance: " + balance);  
        serviceCharge();  
    } else {  
        System.out.println("Insufficient funds for  
        withdrawal");  
    }  
}  
}
```

```
3  
class Bank {  
    public static void main(String args[]){  
        Scanner s1 = new Scanner(System.in);  
        System.out.println("Enter your name:");  
        customer.name = s1.next();  
        System.out.println("Enter your account no.");  
        account.no = s1.nextInt();  
  
        Sav-acct s = new Sav-acct();  
        cur-acct c = new cur-acct();  
        System.out.println("Enter deposit amount for  
        savings account");  
        double s-deposit = s1.nextDouble();  
        s.deposit(s-deposit);  
        System.out.println("Enter deposit amount for current account");  
        double c-deposit = s1.nextDouble();  
        c.deposit(c-deposit);  
        System.out.println("Enter withdrawal amount for savings  
        account");  
        double s-withdrawal = s1.nextDouble();  
        s.withdraw(s-withdrawal);  
    }  
}
```

```
System.out.println("Enter withdrawal amount for current account:");  
double c-withdrawal = s1.nextDouble();  
c.withdraw(c-withdrawal);  
  
s.display();  
c.display();  
}  
}
```

### Output:

```
Enter your name: Raj  
Enter your account no: 4001  
Enter deposit amount for savings account: 8000  
Updated balance: 8000.0  
Enter deposit amount for current account: 13000  
Updated balance: 13000.0  
Enter withdrawal amount for savings account: 2000  
Updated balance: 6000.0  
Enter withdrawal amount for current account: 3000  
Updated balance: 10000.0  
Current balance: 6000.0  
Current balance: 10000.0
```

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 BMS implements Runnable {  
    public void run() {  
        try {  
            while(true) {  
                System.out.println("BMS College of Engineering");  
                Thread.sleep(10000);  
            }  
        } catch (InterruptedException e) {  
            System.out.println("There is an interruption  
occurred");  
        }  
    }  
}
```

```
class CSE implements Runnable {  
    public void run() {  
        try {  
            while(true) {  
                System.out.println("CSE");  
                Thread.sleep(2000);  
            }  
        } catch (InterruptedException e) {  
            System.out.println("There is an interruption  
occurred");  
        }  
    }  
}
```

```
public class Main {  
    public static void main(String [] args){  
        BMS bms = new BMS();  
        Thread t1 = new Thread(bms);  
        t1.start();  
        CSE cse = new CSE();  
        Thread t2 = new Thread(cse);  
        t2.start();  
    }  
}
```

Output:

BMS College of Engineering  
CSE  
CSE  
CSE  
CSE  
BMS College of Engineering  
CSE  
CSE  
CSE  
CSE

g (2) 2024

## AWT programs - Report

- 1) buttondrag.java - It opens a button game window with  $3 \times 3$  tiles with numbers and provides user with buttons for reset, start and restart. Once clicked on start, user can click on two tiles and simultaneously clicked tiles get swapped. User can click on reset to start the game again.
- 2) ButtonList.java - It opens a buttonlist window with three buttons - yes, no, undecided and has a default text HELLO. Once clicking on yes, the window writes the text - you pressed yes. Similarly on clicking No, → you pressed NO, → & on clicking undecided → you pressed undecided.
- 3) ButtonListD.java - It opens buttonlistD window with three buttons - yes, no, undecided. On clicking yes, a Dialog window appears with message you pressed yes and an OK button. Similarly on clicking NO, you pressed NO & on clicking undeciding, you pressed undecided in separate dialog box.
- 4) DivisionMain.java - It opens Divisionof Integers window with fields to enter 2 numbers & a button RESULT, on clicking which the two numbers and quotient will appear on the

window after Result: It gives & intakes numbers in float datatype.

- 5) DivisionMain.java - It opens a Division of Integers similar to last one. But the result provided will be integers & the quotient is float.
- 6) TextFieldDemo.java - It opens a TF-Label Demo window with fields for entering name & password. On clicking enter after entering the name, the text appears after Name. Similarly, on clicking enter after entering the password, it appears next to password. On selecting some letters in name field & clicking enter, the selected text appears next to select text in name. Password receives the & input & displays it after encrypting it to.