



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.Scanner;  
  
public class LAB {  
    public static void main(String args[]) {  
        Scanner sc = new Scanner(System.in);  
        int a, b, c;  
        double d, s1, s2;  
        System.out.println("Enter values of  
        a, b, c");  
        a = sc.nextInt();  
        b = sc.nextInt();  
        c = sc.nextInt();  
        sc.close();  
  
        d = (double)((b * b) - (4 * a * c));  
        if (a == 0) {  
            System.out.println("Invalid");  
            return;  
    }  
}
```

PTO

```

if (a*d<0){
    System.out.println("No real solutions");
} else if (d==0){
    s1 = (double)((-b+Math.sqrt(d))/(2*a));
    s2 = (double)((-b-Math.sqrt(d))/(2*a));
    System.out.printf("Roots are Real
and equal: %.4f and %.4f", s1, s2);
} else {
    s1 = (double)((-b+Math.sqrt(d))/(2*a));
    s2 = (double)((-b-Math.sqrt(d))/(2*a));
    System.out.printf("Roots are Real
and distinct: %.4f and %.4f, s1, s2);
}

```

```

srava@LAPTOP-NRIKOIFA ~\Documents\JavaStuff
> cd "c:\Users\srava\Documents\JavaStuff\" ; if ($?) { javac LAB.java } ; if ($?) { java LAB }
Enter values of a,b,c of a quadratic eqn: 1 1 1
No real solutions!
srava@LAPTOP-NRIKOIFA ~\Documents\JavaStuff
> cd "c:\Users\srava\Documents\JavaStuff\" ; if ($?) { javac LAB.java } ; if ($?) { java LAB }
Enter values of a,b,c of a quadratic eqn: 1 2 1
Roots are Real and Equal: -1.0000 and -1.0000
srava@LAPTOP-NRIKOIFA ~\Documents\JavaStuff
> cd "c:\Users\srava\Documents\JavaStuff\" ; if ($?) { javac LAB.java } ; if ($?) { java LAB }
Enter values of a,b,c of a quadratic eqn: 1 -1 -6
Roots are Real and Disinct: 3.0000 and -2.0000

```

Lab Program 2:

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

```
import java.util.Scanner;

class Student {
    Scanner sc = new Scanner(System.in);
    String USN;
    String Name;
    int credits[] = new int[5];
    int marks[] = new int[5];
    int SGPA;
    int totalCredits=0;

    void getDetails(){
        System.out.println("Enter student USN");
        USN = sc.nextLine();
        System.out.println("Enter name:");
        Name = sc.nextLine();
        for(int i=0; i<5; i++){
            System.out.print("Enter credits for subject " + (i+1) + ": ");
            credits[i] = sc.nextInt();
            totalCredits += credits[i];
            System.out.print("Enter marks for subject " + (i+1) + ": ");
            marks[i] = sc.nextInt();
        }
    }

    void displayDetails(){
        System.out.println("USN: " + USN);
        System.out.println("Name: " + Name);
        System.out.println("Total Credits: " + totalCredits);
        System.out.println("SGPA: " + SGPA);
    }
}
```

```
marks[i] = sc.nextFloat();
```

```
}
```

```
}
```

```
Void ShowDetails(){
```

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

```
System.out.println("Student Name:");
```

```
for(i=0; i<5; i++) {
```

```
System.out.println("Subject " +(i+1) + "-
```

```
credits: " + credits[i] + "- Marks: "
```

```
+ Marks[i]);
```

```
}
```

```
}
```

```
Void CalcSGPA(){
```

```
int points[] = new int[5];
```

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

```
if(Marks[i] < 0) {
```

```
System.out.println("Error: Marks  
below 0");
```

```
return;
```

```
}
```

```
PTO
```

```
{elseif (marks[i]<40){  
    points[i]=0;  
}  
{else if (marks[i]>40 && marks[i]<=50){  
    points[i]=4;  
}  
{else if (marks[i]>51 && marks[i]<=60){  
    points[i]=5;  
}  
{else if (marks[i]>61 && marks[i]<=70){  
    points[i]=7;  
}  
{else if (marks[i]>71 && marks[i]<=80){  
    points[i]=8;  
}  
{else if (marks[i]>81 && marks[i]<=90){  
    points[i]=9;  
}  
{else if (marks[i]>91 && marks[i]<=100){  
    points[i]=10;  
}  
{else {  
    System.out.println("Error marks  
alone 100");  
}}}
```

```
    SGPA += (marks[i] * credits[i]);  
}  
  
System.out.println("SGPA of " + Name +  
    " is: " + (float)(SGPA / totalCredits));  
}  
}
```

```
public class Lab2 {  
    public static void main(String args[]) {  
        Student stu1 = new Student();  
        stu1.getDetails();  
        stu1.showDetails();  
        stu1.calcSGPA;  
    }  
}
```

```
Enter Credits for Subject 5:  
3  
Enter Marks for Subject 5:  
75  
Student USN: 1BM19CS068  
Enter student name: Kalle Venata Sravan Dhira  
Subject 1 - Credits: 5 - Marks: 90.0  
Subject 2 - Credits: 4 - Marks: 78.0  
Subject 3 - Credits: 4 - Marks: 83.0  
Subject 4 - Credits: 4 - Marks: 81.0  
Subject 5 - Credits: 3 - Marks: 75.0  
SGPA of Kalle Venata Sravan Dhira is: 8.9
```

Week 5 Lab prog

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 n book objects.

```
import java.util.Scanner  
class Book {  
    String name, author;  
    double price;  
    int num_pages;  
    Scanner sc = new Scanner(System.in);  
    Book() {}  
    Book(String name, String author, double price, int pages) {  
        this.name = name;  
        this.author = author;  
        this.price = price;  
        this.num_pages = pages  
    }  
    void setDetails() {  
        System.out.println("Enter the name of the book");  
        this.name = sc.nextLine();  
        System.out.println("Enter author of the book");  
        this.author = author;  
        System.out.println("Enter price: ");  
        this.price = sc.nextDouble();  
    }  
    String toString() {  
        return "Book{" + "name=" + name + ", author=" + author + ", price=" + price + ", num_pages=" + num_pages + '}';  
    }  
}
```

```
System.out.print("Enter number of pages:");
    } thunum-pages = sc.nextInt();
void getDetails(){
    System.out.println("Name:" + Name);
    System.out.println("Author:" + Author);
    System.out.println("Price:" + price);
    System.out.println("Pages:" + pages);
}
```

```
public String toString(){
    return ("Name: " + name + "\n Author:" + author
        + "\n Price:" + price + "\n pages:" +
        pages);
}
```

```
Public class Labprog 3{
    public static void main(String args[]){
        int n;
        Book books[];
        Book myBook = new Book("Chamber of
            secrets", "J.K Rowling", 35.55, 867);
        Scanner sc = new Scanner(System.in);
```

```
System.out.print("Enter number of books");
```

```
n = sc.nextInt();
```

```
books = new Book[n];
```

```
for (int i=0; i<books.length; i++) {
```

```
System.out.print(" Details of book" + i + ")");
```

```
books[i] = new Book();
```

```
books[i].setDetails();
```

```
}
```

```
for (i=0; i<books.length; i++) {
```

```
System.out.print(" Details of book" +  
(i+1) + " : ");
```

```
books[i].getDetails();
```

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

```
}
```

```
System.out.print(n(my Book));
```

```
}
```

```
Enter name of the book: Goblet Of Fire
```

```
Enter the author of the book: J.K Rowling
```

```
Enter the price: 35.55
```

```
Enter the number of pages: 354
```

```
Details of book 1:
```

```
Name: Order Of Phoenix
```

```
Author: J.K Rowling
```

```
Price: 35.55
```

```
Number of pages: 402
```

```
Details of book 2:
```

```
Name: Goblet Of Fire
```

```
Author: J.K Rowling
```

```
Price: 35.55
```

```
Number of pages: 354
```

```
Name: Chamber Of Secrets
```

```
Author: J.K Rowling
```

```
Price: 35.55
```

```
Number of pages: 367
```

Week 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.

1) abstract class Shape{
 int b,h;
 Shape (int b, int h){
 this.b = b;
 this.h = h;
 }
 void printArea();
}

class Rectangle extends Shape{
 Rectangle (int b, int h){
 super(b,h);
 }
 void printArea(){
 System.out.println("Area of the rectangle is "+
 (b*h));
 }
}

class Triangle extends Shape{
 Triangle (int b, int h){
 super(b,h);
 }
 void printArea(){
 System.out.println("Area of
 the triangle is " + ((b*h)/2));
 }
}

Class Circle Extends Shape {

Circle (int r) {

super(r, r);

}

Void print Area() {

System.out.println ("Area of the circle is " +

(Math.PI * r * r));

}

}

public class Prog1 {

public static void main (String args[]) {

Rectangle rect = new Rectangle (10, 5);

Triangle tri = new Triangle (10, 5);

Circle c = new Circle (10);

rect.print Area();

tri.print Area();

c.print Area();

}

```
cd "c:\Users\sjava\Documents\JavaStuff\Webs"
a Prog1 }
Area of the rectangle is 50
Area of the triangle is 25
Area of the circle is 314.1592653589793
```

}

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 abstract class Account {  
    String cName, accNum, acctype;  
    Scanner sc = new Scanner(System.in)  
    Account (String name, String accNo,  
             String acctype) {  
        this.cName = name;  
        this.accNum = accNo;  
        this.acctype = acctype;  
    }  
    Account () {}  
}  
  
class Current Acc extends Account {  
    private double balance = 5000, rate = 0.05;  
    private boolean conWithdraw = false;  
    Current Acc (String name, String accNo, String acctype) {  
        super(name, accNo, acctype);  
        System.out.println("Welcome "+cName);  
    }  
    void getBalance () {  
        System.out.format("Your balance : %.2f",  
                         balance);  
    }  
}
```

```
void deposit(double amount){  
    char choice;  
    System.out.println("Deposit Account holder :");  
    System.out.println("Name + " Amount: " + amount);  
    System.out.println("Approve Deposit (Y/N):");  
    choice = sc.next().charAt(0);  
    if (choice == 'y' || choice == 'Y') {  
        balance += amount;  
        System.out.println("Deposit approved. Updated  
balance: " + balance);  
    } else {  
        System.out.println("Deposit not approved.");  
    }  
}  
void withdraw(double amount){  
    System.out.println("This account cannot  
withdraw only funds");  
}  
void checkMinAmount(){  
    if (balance < 3000) {  
        balance -= 500;  
        System.out.print("Balance is below level ");  
        System.out.print("Penalty imposed");  
    }  
}
```

class Savings Acc extends Account {

private double balance = 5000, rate = 0.06;

Savings Acc (String name, String accNo, String
accType) {

super(name, accNo, accType);

}
System.out.println("Welcome " + name);

void getBalance () {

System.out.format("Your Balance : %.2f\n",
balance);

void deposit (double amount) {

char choice ;

System.out.println ("Deposit Options
Deposit : Account holder : " + name + " Amount
+ amount);

System.out.println ("Approve Deposit ? (Y/N) ! ");

choice = sc.next().charAt(0);

if (choice == ~~sc.next().charAt(0)~~; 'Y' || choice == 'y') {
balance += amount;

System.out.println (" Deposit approved .
Updated Balance : " + balance);

```
calc Interest();  
check Min amount();  
}  
else {  
    System.out.println(ANSI "Deposit not  
void calc Interest(){  
    double CI;  
    CI = balance => * (Math.pow((tRate / 100), 2));  
    balance += CI;  
    System.out.println("Interest Added.");  
void withdraw(double Amount){  
    char choice;  
    if (balance < amount){  
        System.out.println("Account Balance  
is lower than Amount to be Withdrawn");  
    }  
    return;  
}  
System.out.println("Approve " + cName + " Is  
request for withdrawal?(Y/N) ");  
choice = sc.next().charAt(0);  
if (choice == 'Y' || choice == 'y') {  
    balance +- amount
```

System.out.println ("Withdrawal approved.

calc Interest(); Updated Balance: "+balance);
check min Amount();

} else {

System.out.println ("Withdrawal not
approved");

}

}

void check min Amount(){

int min amount = 3000, penalty = 500;
if (balance < min amount){
balance -= penalty;

}

System.out.println ("Bal is under minimum
amount to be maintained");

System.out.println ("Penalty imposed");

}

}

```
public class Bank {  
    public static void main (String [] args) {  
        int c;  
        double temp;  
        String name, accNo, accType;  
        Scanner sc = new Scanner (System.in);  
        System.out.println ("Enter Name: ");  
        name = sc.nextLine ();  
        System.out.println ("Enter Account number: ");  
        accNo = sc.nextLine ();  
        System.out.println ("Enter type: ");  
        accType = sc.nextLine ();  
        if (accType.charAt (0) == 'C') {  
            CurrentAcc a = new CurrentAcc (  
                name, accNo, accType);  
            while (true) {  
                System.out.println ("1. deposit\n" +  
                    "2. Withdraw Money\n" +  
                    "3. Display Money\n" +  
                    "4. Exit");  
                c = sc.nextInt();  
                if (c == 1)  
                    a.deposit ();  
                else if (c == 2)  
                    a.withdraw ();  
                else if (c == 3)  
                    a.display ();  
                else if (c == 4)  
                    break;  
            }  
        }  
    }  
}
```

switch(c) {

case 1: {

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

temp = sc.nextDouble();

a. deposit (temp);
break;

}

case 2: { System.out.println ("Enter
amount to be withdrawn : ");

temp = sc.nextDouble();

a. withdraw (temp);
break;

}

case 3: {

a. get Balance ();

break;

}

case 4: {

System.exit (0);

break;

default: System.out.println ("Enter correct option");

}

}

else if (acctype.chatat(0) == 'S') {

Savings Acc a = new SavingsAcc (name, accNo,
acctype);

while (true) {

c = sc.nextInt();

switch (c) {

case 1 : {

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

temp = sc.nextDouble();

a.deposit (temp);

break;

}

case 2 : {

System.out.println ("Enter amount
to be withdrawn : ");

temp = sc.nextDouble();

a.withdraw (temp);

break;

3

case 3: {

a. getBalance();
break;

}

case 4: {

System.exit(0);
break;

}

default: System.out.println("Enter the
correct options");

}

}

} else {

System.out.println("Enter valid type... Exiting");

}

}

```
4. Exit  
1  
Enter amount to be deposited:  
2000
```

```
Deposit. Account holder: Sravan Amount: 2000.0  
Approve Deposit?(Y/S):  
Y
```

```
Deposit approved. Updated balance: 7000.0
```

```
Interest added. Updated balance: 14008.40252
```

- 1. Deposit money
- 2. Withdraw money
- 3. Display money
- 4. Exit

```
2
```

```
Enter amount to be withdrawn:
```

```
15000
```

```
Account balance is lower than amount to be withdrawn
```

- 1. Deposit money
- 2. Withdraw money
- 3. Display money
- 4. Exit