

## Program - 2

Develop a Java program that prints all real solution to the quadratic equations  $ax^2 + bx + c = 0$ . 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;
class quad
{
    double Disc(double a, double b, double c)
    {
        return b*b - 4*a*c;
    }
    void root(double a, double b, double c)
    {
        double D = Disc(a,b,c);
        if (D < 0)
        {
            double realpart = -b/(2*a);
            double imaginarypart = Math.sqrt(Math.abs(D))/2*a;
            System.out.println("The quadratic
Equation has conjugate imaginary Roots:");
            System.out.print("Root 1: %.5f + %.5fi\n",
                            realpart, imaginarypart);
            System.out.print("Root 2: %.5f - %.5fi
%.n", realpart, imaginarypart);
        }
    }
}
```

```
else if (D > 0)
```

```
{
```

```
    system.out.println("The quadratic Equation has  
two Distinct Real Roots:");
```

```
    double r1 = (-b + Math.sqrt(D)) / (2 * a);
```

```
    double r2 = (-b - Math.sqrt(D)) / (2 * a);
```

```
    system.out.printf("Root 1 : %.5f\n", r1);
```

```
    system.out.printf("Root 2 : %.5f\n", r2);
```

```
{
```

```
else
```

```
{
```

```
    system.out.println("The quadratic Equation  
has equal and Real Root :");
```

```
    double r1 = (-b) / (2 * a);
```

```
    system.out.printf("Both Root 1 and Root 2 :  
%.5f\n", r1);
```

```
{
```

```
{
```

```
class QuodEqn
```

```
{
```

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

```
{
```

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

```
    System.out.println ("Enter the coefficients  
of Quadratic Equation :");
```

```
    double a = s1.nextDouble();
```

```
    double b = s1.nextDouble();
```

```
    double c = s1.nextDouble();
```

```
if (a == 0)
```

```
{
```

System.out.println ("since the coefficient  
of  $x^2$  is zero, it's not a Quadratic  
Equation");

```
}
```

```
else
```

```
{
```

```
Quod
```

Quadratic = new Quod();

Quadratic.Disc(a, b, c);

Quadratic.root(a, b, c);

```
}
```

```
{
```

```
{
```

```
{
```

Output

```
> java QuadEqn.java
> java Quod Eqn
```

Enter the coefficient of Quadratic Equation:

23

24

25

The quadratic Equation has conjugate  
imaginary Root.

Root<sub>1</sub> :-

- 0.52174 + 0.92640i;

Root<sub>2</sub> :-

- 0.52174 - 0.92640i

Develop Java program

### 3) S GPA

```
import java.util.Scanner;
class Student
{
    int usn;
    String name;
    int []marks = new int[8];
    int [] credit = new int[8];
    int [] Gcredit = new int[8];
    Scanner s1 = new Scanner (System.in);
    void accept_details()
    {
        System.out.println("Enter student usn:");
        usn = s1.nextInt();
        System.out.println("Enter student name");
        name = s1.next();
        System.out.println("Enter marks in order
                           of credit:");
        for (int i=0; i<8; i++)
        {
            marks[i] = s1.nextInt();
        }
        System.out.println("Enter order of credits");
        for (int i=0; i<8; i++)
        {
            credit[i] = s1.nextInt();
        }
    }
}
```

```
void calculate()
{
    for (int i=0; i<8; i++)
    {
        if (marks[i] >= 90)
        {
            credit_points[i] = 10 * credit[i];
        }
        else if (marks[i] >= 80)
        {
            credit_points[i] = 9 * credit[i];
        }
        else if (marks[i] >= 70)
        {
            credit_points[i] = 8 * credit[i];
        }
        else if (marks[i] >= 60)
        {
            credit_points[i] = 7 * credit[i];
        }
        else if (marks[i] >= 50)
        {
            credit_points[i] = 6 * credit[i];
        }
        else if (marks[i] >= 40)
        {
            credit_points[i] = 5 * credit[i];
        }
    }

    int sum = 0;
    int count = 0;
    double GPA;

    for (int j=0; j<8; j++)
    {
        sum = sum + credit_points[j];
        count = count + credit[j];
    }
}
```

```
SerPA = sum I count ;  
System.out.println("SerPA is "+SerPA);
```

{  
}  
}  
}

class Main

{  
}

public static void main (String args[])

{  
}

System.out.println ("Enter <sup>no</sup> of Student")

Scanner S=new Scanner (System.in) ;  
int n= S.nextInt();

Student ST []=new Student [n];

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

{  
}

ST [i]=new Student ();

ST [i].accept\_details();

ST [i] → calculate();

{  
}  
}

③ import java.util.Scanner;

```
class Book {  
    String name;  
    String author;  
    double price;  
    int numPages;
```

```
    public Book (String name, String author,  
                double price, int numPages) {
```

```
        this.name = name;  
        this.author = author;  
        this.Price = price;  
        this.numPages = numPages;
```

```
}
```

```
    public void setDetails() {
```

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

```
        System.out.print("Enter book name:");  
        this.name = scanner.nextLine();
```

```
        System.out.print("Enter author name:");
```

```
        this.author = scanner.nextLine();
```

```
        System.out.print("Enter price:");
```

```
        this.price = scanner.nextDouble();
```

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

```
        this.numPages = scanner.nextInt();
```

```
    public void getDetails()
```

```
{
```

```
    System.out.println("Book Name: " +  
                      name);
```

```
    System.out.println("Author: " + author);
```

```
        System.out.println("Price: $" + price);
        System.out.println("Number of pages:"
                           + numPages);
    }
```

```
public String toString()
```

```
{  
    return "Book Details: \n" +  
           "Name: " + Name + "\n" +  
           "Author: " + author + "\n" +  
           "Price: $" + price + "\n" +  
           "Number of pages: " +  
           numPages;  
}
```

```
}
```

```
{
```

```
public class Main {
```

```
    public static void main(String[] args) {  
        Scanner scanner = new  
        Scanner(System.in);  
  
        System.out.print("Enter the number of  
books: ");  
        int n = scanner.nextInt();
```

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

```
        for (int i = 0; i < n; i++) {  
            System.out.println("\nEnter details  
for Book " + (i+1) + ":");
```

```
            books[i] = new Book("", "", 0, 0, 0);  
            books[i].setDetails();
```

```
}
```

```
        System.out.println("\nDetails of all  
books: ");
```

```
for (int i=0; i<n; i++) {  
    System.out.println ("In Book " + (i+1) + ":")  
    books[i].getDetails ();
```

{

in complete details  
of all books :");

```
for (int i=0; i<n; i++) {  
    System.out.println ("In Book " + (i+1) + ":")  
    + ":" + books[i].toString () ;
```

{

{

{

public static printArea()

## D shaped

import java.util.Scanner;

abstract class Shape {

int a;

int b;

abstract void printArea();

class Rectangle extends Shape {

void printArea()

{ int c = a \* b;

System.out.println("area of  
rectangle = " + c);

}

}

class Triangle extends Shape {

void printArea()

{ float c = (float) 0.5 \* a \* b;

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

}

}

class Circle extends Shape {

void printArea()

{ float c = (float) 3.14 \* a \* a;

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

}

}

class main

{ public static void main (String [ ] args)

{ int a, b;

```

        int choice;
do {
    system.out.println("1. enter choice 1A);
    rectangleIn 2. triangle In 3. CircleIn
    4. exit");
    Scanner s = new Scanner (System.in);
    choice = s.nextInt();
    switch (choice) {
        case 1 : Rectangle t = new Rectangle();
                    system.out.println("enter length
                                         and breadth");
                    a = s.nextInt();
                    b = s.nextInt();
                    t.a = a;
                    t.b = b;
                    t.printArea(); break;
        case 2 : triangle t = new Triangle();
                    system.out.println("enter
                                         height and base");
                    a = s.nextInt();
                    b = s.nextInt();
                    t.a = a;
                    t.b = b;
                    t.printArea(); break;
        case 3 : circle c = new Circle();
                    system.out.println("enter radius
                                         of circle");
                    a = s.nextInt();
                    c.a = a;
                    c.printArea(); break;
        case 4 : system.exit(0); break;
    }
} while (true);
}

```

## Bank

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

```
class Bank {
```

```
    double accno;
```

```
    String name;
```

```
    String type;
```

```
    double balance;
```

```
    public Bank(double accno, String name, String type,  
               double balance) {
```

```
        this.accno = accno;
```

```
        this.name = name;
```

```
        this.type = type;
```

```
        this.balance = balance;
```

```
}
```

```
    public void display() {
```

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

```
}
```

```
    public void deposit(double deposit) {
```

```
        this.balance += deposit;
```

 ~~System.out.println("updated balance is  
" + balance);~~

```
}
```

```
    public void withdraw(double amount) {
```

```
        balance -= amount;
```

 ~~System.out.println("After withdrawing,  
the balance is :" + balance);~~

```
}
```

class Saving, extends Bank {  
double rate;  
int time;

public Saving(double accno, string name,  
double balance, int time,  
double rate) {

super (accno, name, "Saving", balance);  
this.time = time;  
this.rate = rate;

}

public void calculateInterest() {

balance += (balance \* time \* rate) / 100;

}

}

class Current extends Bank {  
double minBalance;

public Current(double accno, string name,  
double balance, double  
minBalance) {

super (accno, name, "Current", balance);  
this.minBalance = minBalance;

}

public void applyServiceCharge() {  
if (balance < minBalance) {

System.out.println("Service charge of  
5% is applied:");

balance = balance \* 0.05;

class main {

```
public static void main (String [] args) {
    double accno;
    String name;
    double balance;
    Scanner s = new Scanner (System.in);
    System.out.println ("Enter accno:");
    accno = s.nextDouble ();
    System.out.println ("Enter name:");
    name = s.next ();
    System.out.println ("Enter balance:");
    balance = s.nextDouble ();
    System.out.println ("Enter balance after :");
    double time;
    System.out.println ("Enter the time of loan:");
    int time = s.nextInt ();
    System.out.println ("Enter balance rate:");
    double rate = s.nextDouble ();
    System.out.println ("Enter min balance");
    double minBalance = s.nextDouble ();
    System.out.println ("Do you want current account (1) or saving account (2): ");
    int ch = s.nextInt ();
```

```
if (ch == 1) {
    Savings s1 = new Savings (accno, name, balance,
                           time, rate);
    System.out.println ("Depositing 1000Rs.");
    s1.deposit(1000);
    System.out.println ("Withdrawing 500Rs.");
    s1.withdraw(500);
    s1.calculateInterest();
    System.out.println ("Displaying balance");
    s1.display();
} else if (ch == 2) {
    Current c = new Current (accno, name,
                             balance, time, rate, minBalance);
    System.out.println ("Depositing 1000Rs.");
    c.deposit(1000);
    System.out.println ("Withdrawing 500Rs.");
    c.withdraw(500);
    c.applyServiceCharge();
    System.out.println ("Displaying balance");
    c.display();
}
System.out.println ("Closing account");
s.close();
}
```

## Package

Package CIE;

Public class Student {

Public String USN;

Public String name;

Public int dem;

Public Student (String U, String n, int S)

}

this.USN = U;

this.name = n;

this.dem = S;

}

Public class Internals extends CIE.Student

{ Public double marks[];

Public Internals (String U, String n,  
double m[]) { Super (U, n, S) }

this.marks = m;

}

}

Package SEE;

import CIE.Student;

Public class Externals extends CIE.Student

{

Public double smarks[];

Public Externals (String U, String n,  
int S, double m[]) {

Super (U, n, S);

this.smarks = m;

}

}

package result;

```
import CIE.Student;
import CIE.internal;
import SEE.external;
public static void main (String args[])
{
    double internal[] = { 43, 43, 43, 44 };
    double external[] = { 90, 83, 65, 98, 43 };
    Student s1 = new Student ("IBM2214", "Jose",
    3);
```

```
internal i1 = new internal ("IBM2215", 3,
    internal);
```

```
external e1 = new external ("IB2216", 150, 3,
    external);
```

```
System.out.println ("USN: " + s1.USN + "name",
    + s1.name + "Sem: " + s1.sem);
```

```
System.out.println ("internal marks");
for (i = 0; i <; i++)
    {
```

~~```
System.out.println ("internal marks" +
    (i+1) + i1.imarks[i]);
```~~~~```
System.out.println ("external marks");
for (int i = 0; i < 5; i++)
    {
```~~~~```
System.out.println ("external marks")
    + (i+1) + e1.smarks[i]);
```~~

{

{

{

## ⑤ Exception Handling

import java.util.Scanner;

class Wrongage extends Exception {

    public String toString() {

        return "Age cannot be less than 0";

}

}

class Son extends Exception {

    public String toString() {

        return "Son cannot be older than  
            father";

}

}

class Father {

    int age;

    Father(int age) throws Wrongage {

        this.age = age;

        if (age < 0) {

            throw new Wrongage();

}

}

}

Class Son extends Father {

    int fatherage, sonage;

    Son (int fatherage, int sonage) throws

super (sonage);  
this. fatherage = fatherage;  
this. sonage = sonage;

if (sonage == fatherage) {

    System.out.println("Son cannot be father");

} else {

    System.out.println("fatherage" +  
        " " + "in sonage" + sonage);

}

{

{

class main {

    public static void main (String[] args) {

        int fatherage, sonage;

        Scanner s = new Scanner (System.in);

    try {

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

        fatherage = s.nextInt();

        father f = new father (fatherage);

        System.out.print ("Enter sonage:");

        sonage = s.nextInt();

        son son object = new son (fatherage, sonage);

    } catch (Exception e) {

        System.out.println("Exception:  
                       " + e);

} .

g  
g  
g

✓

## generic

class generic<A, B, C, D, E> {

- A a;
- B b;
- C c;
- D d;
- E e;

generic(A a1, B b1, C c1, D d1, E e1) {

    a = a1;  
    b = b1;  
    c = c1;  
    d = d1;  
    e = e1;

}

void showType() {

    System.out.println(a.getClass().getName());  
    System.out.println(b.getClass().getName());  
    System.out.println(c.getClass().getName());  
    System.out.println(d.getClass().getName());  
    System.out.println(e.getClass().getName());

{

    A returnType1() {  
        return a;

    } B returnType2() {  
        return b;

}

    C returnType3() {  
        return c;

```
D return type3() {  
    return d;  
}  
E return type4() {  
    return e;  
}
```

```
class generic demo {
```

```
    public static void main (String [] args) {  
        generic < Integer, Float, String, Double,  
        Boolean > g1;
```

```
        g1 = new generic < Integer, Float, String,  
        Double, Boolean > (d1: 1, (float) 1.2322,  
        c1: "BMS", d1: 2.11332, e1: true);
```

~~g1. showType();~~~~int x = g1. returnType1();~~~~System.out.println(x);~~~~Float x1 = g1. returnType2();~~~~System.out.println(x1);~~~~String x2 = g1. returnType3();~~~~System.out.println(x2);~~~~Double x3 = g1. returnType4();~~~~System.out.println(x3);~~~~Boolean x4 = g1. returnType5();~~~~System.out.println(x4);~~

```
}
```

```
}
```

import java.awt.\*;  
import java.awt.event.\*;  
  
public class DivisionMain1 extends Frame implements  
ActionListener

{

Text Field num1, num2  
Button dResult;  
Label outResult;  
String out = "";  
double resultNum = 0;  
int flag = 0;

public DivisionMain1()

{

setLayout(new FlowLayout());  
dResult = new Button("Result");  
Label number1 = new Label("Number1:",  
"Label.RIGHT");  
Label number2 = new Label("Number2:",  
"Label.RIGHT");  
num1 = new TextField(5);  
num2 = new TextField(5);  
outResult = new Label("Result:", "Label.  
RIGHT");  
add(number1);  
add(num1);  
add(number2);  
add(num2);  
add(dResult);  
add(outResult);

num1.addActionListener(this);  
num2.addActionListener(this);  
dResult.addActionListener(this);  
addWindowListener(new WindowAdapter()  
{})

{

```
Public void windowClosing(WindowEvent we)
{
    System.exit(0);
}

}

Public void actionPerformed(ActionEvent ae)
{
    int n1, n2;
    try
    {
        if (ae.getSource() == dResult)
        {
            n1 = Integer.parseInt(num1.getText());
            n2 = Integer.parseInt(num2.getText());
            /*if (n2 == 0)
                throw new ArithmeticException("Division by zero");
            out = n1 + " + " + n2 + " = " + resultNum;
            resultNum = n1 / n2;
            out += String.valueOf(resultNum);
            repaint();
        }
    }
    catch (NumberFormatException e1)
    {
        flag = 1;
        out = "Number Format Exception! " + e1;
        repaint();
    }

    catch (ArithmeticException e2)
    {
        flag = 1;
        out = "Divide by 0 Exception! " + e2;
        repaint();
    }
}
```

public void paint (Graphics g)

{ if (flag == 0)

g.drawString (out, outResult.getX () + outResult.getWidth (),  
outResult.getY () + outResult.getHeight () - 8);

else  
g.drawString (out, 100, 200);

flag = 0;

} P6

public static void main (String [] args)

{ Division Main1 dm = new Division main1 ();  
dm.setFize (new Dimension (800, 400));  
dm.setTitle ("Division of Integers");  
dm.setVisible (true);

✓  
23/7/24