

LAB PROGRAM 1

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.

Program :

Week-3 (LAB-1)

1) Java program that prints all real solutions to quadratic equation $ax^2 + bx + c = 0$. If discriminant $b^2 - 4ac$ is negative, display a message saying there are no real solutions.

```
→ import java.util.Scanner;
public class RootsOfQE {
    public static void main(String args[]) {
        double Root1 = 0, Root2 = 0;
        Scanner sc = new Scanner(System.in);
        System.out.println("Consider a quadratic equation");
        System.out.println("Enter the value of a:");
        double a = sc.nextDouble();
        System.out.println("Enter the value of c:");
        double c = sc.nextDouble();
        double determinant = (b*b) - (4*a*c);
        double Sqrt = Math.Sqrt(determinant);

        if (determinant >= 0) {
            Root1 = (-b + Sqrt) / (2*a);
            Root2 = (-b - Sqrt) / (2*a);
            System.out.println("Roots are real and distinct");
            System.out.println("Roots are " + Root1 +
                               " and " + Root2);
        }
        if else (determinant == 0) {
            Root1 = (-b) / (2*a);
            Root2 = Root1;
            System.out.println("Roots are equal and real");
        }
    }
}
```

`System.out.println("Roots are" + Root1 + "and" + Root2);
 }
 else {
 System.out.println("There are no real solutions");
 }
}`

CS Scanned with CamScanner

```

Enter the value of a :
1
Enter the value of b :
-2
Enter the value of c :
-8
Roots are real and distinct
Roots are 4.0 and -2.0

```

```

Enter the value of a :
1
Enter the value of b :
2
Enter the value of c :
1
Roots are real and equal
Roots are :: -1.0 and -1.0

```

```

Enter the value of a :
5
Enter the value of b :
6
Enter the value of c :
7
There are no real roots

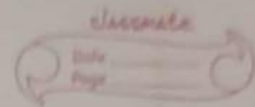
```

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.

Program:

Week-4 (LAB-2)



```
1) import java.util. Scanner;
class student {
    private int Cred [], tc = 0;
    private double mark [], Cal [], Sum = 0, Sgpa;
    private String name, VSN;

    void accept()
    {
        System.out.println("Enter Student details");
        Scanner s1 = new Scanner(System.in);
        System.out.println("Enter name:");
        name = s1.nextLine();
        System.out.println("Enter V.SN:");
        VSN = s1.nextLine();
        mark = new double[5];
        Cred = new int[5];
        Cal = new double[5];
        for (int i = 0; i < 5; i++) {
            System.out.println("Enter mark [" + i + "]:");
            mark[i] = s1.nextDouble();
        }
        for (int i = 0; i < 5; i++) {
            System.out.println("Enter Cred [" + i + "]:");
            Cred[i] = s1.nextInt();
        }
    }

    void display()
    {
        System.out.println("name:" + name);
        System.out.println("VSN:" + VSN);
        System.out.println("Marks of Student are:");
    }
}
```

```

    for(int i=0; i<5; i++){
        System.out.println(mark[i]);
    }
}

Void calculate()
{
    num = new double[5]
    for(int i=0; i<5; i++){
        if(mark[i] > 100){
            System.out.println("Not-Valid");
        }
        else if(mark[i] >= 90){
            num[i] = 10;
        }
        else if(mark[i] >= 80 && mark[i] < 90)
        {
            num[i] = 9;
        }
        else if(mark[i] >= 70 && mark[i] < 80){
            num[i] = 8;
        }
        else if(mark[i] >= 60 && mark[i] < 70)
        {
            num[i] = 7;
        }
        else if(mark[i] >= 50 && mark[i] < 60){
            num[i] = 5;
        }
        else if(mark[i] >= 40 && mark[i] < 50)
        {
            num[i] = 4;
        }
    }
}

```



```

else {
    num[i] = 0;
}
Cal[i] = num[i] * Cred[i];
Sum = Sum + Cal[i];
tc = tc + Cred[i];
}
Sgpa = Sum / tc;
System.out.println("Sgpa :- " + Sgpa);
}
}

public class Main
{
    public static void main (String args[])
    {
        Student obj = new Student();
        obj.accept();
        obj.display();
        obj.calculate();
    }
}

```

```
enter student details
```

```
enter name:
```

```
jayanti
```

```
enter USN:
```

```
1BM19CS067
```

```
Enter mark[0]:
```

```
90
```

```
Enter mark[1]:
```

```
87
```

```
Enter mark[2]:
```

```
92
```

```
Enter mark[3]:
```

```
80
```

```
Enter mark[4]:
```

```
87
```

```
Enter cred[0]:
```

```
5
```

```
Enter cred[1]:
```

```
4
```

```
Enter cred[2]:
```

```
4
```

```
Enter cred[3]:
```

```
3
```

```
Enter cred[4]:
```

```
3
```

```
name:jayanti
```

```
usn:1BM19CS067
```

```
Marks of student are:
```

```
90.0
```

```
87.0
```

```
92.0
```

```
80.0
```

```
87.0
```

```
sgpa:9.473684210526315
```

```
...Program finished with exit code 0
```

```
Press ENTER to exit console.
```

LABPROGRAM-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.

Program :

LAB-3

```
1) import java.util.Scanner;
   class Book {
       String name;
       String author;
       Double price;
       int num-pages;
       Book()
       {}
       Book(String name, String author, Double price,
            int num-pages)
       {
           this.name = name;
           this.author = author;
           this.price = price;
           this.num-pages = num-pages;
       }
       void accept()
       {
           Scanner S = new Scanner(System.in);
           System.out.println("Enter Book name");
           name = S.nextLine();
           System.out.println("Enter author");
           author = S.nextLine();
           System.out.println("Enter price");
           price = S.nextDouble();
           System.out.println("Enter the number of pages
                               of the book");
           num-pages = S.nextInt();
       }
   }
```

```

public String toString(){
    return("Name:" + name + "\n" + "Author:" + author + "\n"
           + "Price:" + price + "\n" + "No. of pages:" + num_pages);
}

public class Main{
    public static void main(String ss[])
    {
        Scanner a = new Scanner(System.in);
        Book b1 = new Book("Wizard", "Mary", 299, 345);
        System.out.println("Sample book: \n" + b1);
        System.out.println("Enter the number of books");
        int n = a.nextInt();
        Book b[] = new Book[n];
        for(int i=0; i<n; i++)
        {
            b[i] = new Book();
            System.out.println("Enter details" + (i+1) + "book");
            b[i].accept();
        }
        for(int i=0; i<n; i++)
        {
            System.out.println("Details of book" + (i+1));
            System.out.println(b[i]);
        }
    }
}

```

```
Sample book:
Name: wizard
Author: mary
Price: 299.0
Number of pages: 345
Enter the number of books
2
Enter the details of 1 book
Enter the name of the book
harry porter
Enter the author of the book
jk.rowling
Enter the price of the book
500
Enter the number of pages of the book
455
Enter the details of 2 book
Enter the name of the book
oceans
Enter the author of the book
seaan
Enter the price of the book
1011
Enter the number of pages of the book
900
```

```
Details of book 1
Name: harry porter
Author: jk.rowling
Price: 500.0
Number of pages: 455
Details of book 2
Name: oceans
Author: seaan
Price: 1011.0
Number of pages: 900
```

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

LABPROGRAM-4

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.

Program :

LAB-4

```
abstract class shape {
    int x, y;
    abstract void printArea();
}

class Rectangle extends shape
{
    Rectangle(int a, int b) {
        x = a;
        y = b;
    }
    void printArea() {
        System.out.println("Area of Rectangle is " + (x * y));
    }
}

class Triangle extends shape
{
    Triangle(int a, int b) {
        x = a;
        y = b;
    }
    void printArea() {
        System.out.println("Area of Triangle is " + (0.5 * x * y));
    }
}

class Circle extends shape
{
    Circle(int a) {
        x = a;
    }
}
```

classmate
Date _____
Page _____

```

Void printArea() {
    System.out.println("Area of circle is" + (3.142 *
                                                                n * n));
}
}
}
public class Main
{
    public static void main(String args[]) {
        rectangle rect = new rectangle(5, 5);
        triangle tri = new triangle(6, 3);
        circle cir = new circle(5);
        rect.printArea();
        tri.printArea();
        cir.printArea();
    }
}

```

CS Scanned with CamScanner

```

Area of rectangle is 25
Area of triangle is 9.0
Area of circle is 78.55

```

```

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

```


LABPROGRAM-5

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

Program :

LAB-5

```
import java.util.Scanner;
abstract class account {
    String accname;
    long accnum;
    double balance;
    final int minbal = 1000;
    account(String name, long num, double bal, String type) {
        accname = name;
        accnum = num;
        balance = bal;
        acctype = type;
    }
    abstract void addBal(double amt);
    abstract void dispBal();
    abstract void withdraw(double amt);
}

class curr-act extends account {
    curr-act(String name, long num, double bal, String type) {
        super(name, num, bal, type);
        System.out.println("name:" + accname + "\naccnum:" + accnum + "\nbalance:" + balance + "\naccname: Current");
    }
    void addBal(double amount) {
        balance = balance + amount;
    }
    void dispBal() {
        System.out.println("Your balance is" + balance);
    }
}
```

```

    void withdraw(double amount){
        if(balance < amount){
            System.out.println("You don't have enough balance.");
            System.out.println("Balance is " + balance);
            return;
        }
        balance = balance - amount;
        System.out.println("balance = " + balance);
        if(balance < minbal){
            System.out.println("Penalty of Rs." + (balance * 0.01)
                               + " as balance is less than minimum");
            balance = balance - balance * 0.01;
            System.out.println("Current balance = " + balance);
        }
    }
}

class sav_acct extends account{
    sav_acct(String name, long num, double bal){
        super(name, num, bal, "Savings");
        System.out.println("name: " + accname + " \t accno: " +
                           accnum + " \t bal: " + bal + " \t type: " + acctype);
    }
    void addBal(double amount){
        balance = balance + amount;
        interest();
    }
    void interest(){
        int t=2;
        balance = balance * Math.Pow(1 + (a2), t);
    }
}

```

classmate
Date _____
Page _____

```

void dispBal() {
    System.out.println("Your balance is:" + balance);
}

void withdraw(double amount) {
    balance = balance - amount;
    System.out.println("Balance = " + balance);
}
}

```

```

public class Main
{
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        Curr-act c = new Curr-act("jay", 123456,
                                   3000.0, "Current");
        double amount;
        int flag = 0;
        while (flag == 0) {
            System.out.println("1. Add Bal \n 2. Display Bal \n 3. withdraw \n 4. Check Book \n 5. quit");
            int ch = sc.nextInt();
            switch (ch) {
                case 1:
                    System.out.println("Enter amount to be withdrawn");
                    amount = sc.nextDouble();
                    c.withdraw(amount);
                    break;
                case 2:
                    c.dispBal();
                    break;
            }
        }
    }
}

```


Case 3:

```
System.out.println("Enter amount to be withdrawn");
amount = Sc.nextDouble();
C.withdraw(amount);
```

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

```
}
}
```

Sav acct S = new Sav acct("jennie", 500.676, 7000)

```
flag = 0;
```

```
while(flag == 0){
```

```
System.out.println("1. Add bal\n 2. display Bal\n 3. withdraw\n 4. quit");
```

```
int ch = Sc.nextInt();
```

```
Switch (ch){
```

Case 1:

```
System.out.println("Enter amt to be added:");
```

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

```
S.addBal(amount);
```

```
break;
```

Case 2:

```
S.dispBal();
```

```
break;
```

Case 3:

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

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

```
S.withdraw(amount);
```

```
break;
```

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

```
}
}
```

```
name: jay
accnum: 123456
balance: 3000.0
acctype: Current
1:AddBal
2:displayBal
3:withdraw
4:checkbook
5:quit
1
enter amount to be added:
2000
1:AddBal
2:displayBal
3:withdraw
4:checkbook
5:quit
3
enter amount to be withdrawn:
1000
balance = 4000.0
1:AddBal
2:displayBal
3:withdraw
4:checkbook
5:quit
3
enter amount to be withdrawn:
3500
balance = 500.0
penalty of RS.5.0 as balance is less than the minumun needed
current balance = 495.0
1:AddBal
2:displayBal
3:withdraw
4:checkbook
5:quit
4
enter details
enter name of the reciever:
jay
enter the amount to be sent:
200
Enter password
12345
    reciever : jay
amount sent is 200.0
balance = 295.0
```



```
1:AddBal
2:displayBal
3:withdraw
4:checkbook
5:quit
5
name: jennie      accno: 500676    bal: 7000.0      type: Savings
1:AddBal
2:displayBal
3:withdraw
4:quit
1
enter amt to be added:
400
1:AddBal
2:displayBal
3:withdraw
4:quit
2
Your balance is: 10656.0
1:AddBal
2:displayBal
3:withdraw
4:quit
3
enter amt to be withdrawn:
1000
balance = 9656.0
1:AddBal
2:displayBal
3:withdraw
4:quit
4

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