

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.

PAGE No.	11
DATE	

Ques Write a 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.

CODE:

```
import java.io.*;
import java.util.*;
public class Quadratic
{
    public static void roots (double a, double b, double c)
    {
        double d, root1, root2;
        if (a == 0)
        
```

$$d = b^2 - 4ac;$$

if ( $d > 0$ )

{

$$\text{root1} = (-b + \sqrt{d}) / (2 * a);$$

```
root2 = (-b - Math.sqrt(d1)) / (2*a);
```

```
System.out.println ("Roots are equal");
```

```
System.out.println ("Root1:");
```

```
System.out.printf ("%f", root1);
```

```
System.out.println ("Root2:");
```

```
System.out.printf ("%f", root2);
```

```
}
```

```
if (d == 0)
```

```
{
```

System.out.println("roots are real and equal");

?  
else if(d<0)

{  
System.out.println("roots are complex and imaginary  
there are no real solutions");

}

?  
}

public static void main(String[] args) {

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

System.out.printf("Input the value of a");

double a = scanner.nextDouble();

System.out.printf("Input the value of b");

double b = scanner.nextDouble();

System.out.printf("Input the value of c");

double c = scanner.nextDouble();

roots(a,b,c);

}

```
C:\Users\admin\Documents>java QuadEq
```

```
Input the value of a 3
```

```
Input the value of b 2
```

```
Input the value of c 1
```

```
roots are complex and imaginary there are no real solutions.
```

**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.**

Develop a Java Program to create a class Student with member usnname, an array credits and an array marks. Include methods to accept and display details and method to calculate SGPA of the student.

```
CODES import java.util.Scanner;  
class Student  
{  
    private String usn;  
    private String name;  
    private int[] credits = new int[20];  
    private int[] marks = new int[20];  
    private int n;  
  
    void getDetails()  
    {  
        Scanner sc = new Scanner(System.in);  
        System.out.println("Enter the no. of subjects:");  
        n = sc.nextInt();  
        System.out.print("Enter the usn");  
        usn = sc.next();  
        System.out.print("Enter the student name");  
        name = sc.next();  
        for (int i=0; i<n; i++)  
        {  
            System.out.print("Enter credits followed by  
marks for subject " + (i+1) + ":");  
            credits[i] = sc.nextInt();  
            marks[i] = sc.nextInt();  
        }  
    }  
}
```

void printDetails()

{

System.out.println("Student Details are:");

System.out.println("Name: " + name + " TuSN: " + tuSN);

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

{

System.out.print("Subject " + (i + 1) + " Marks is: " + marks[i] + " It Credit is: " + credits[i]);

}

void sgpacalc

{

double sgpa;

int gpcr[] = new int[n];

int credSum = 0; gp, sgpcr = 0;

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

{

credSum += credits[i];

if (marks[i] >= 90)

{

gp = 10;

}

else if (marks[i] >= 80)

{

gp = 9;

}

else if (marks[i] >= 70)

{

gp = 8;

}

else if (marks[i] >= 60)

{

gp = 7;

}

else if (marks[i] >= 50)

{

gp = 6;

}

else if (marks[i] >= 45)

{

gp = 5;

}

else if (marks[i] >= 40)

{

gp = 4;

}

else

{

gp = 0;

}

gpcr[i] = gp \* credit[i];

sgpcr += gpcr[i];

}

sgpa = sgpcr / (reditsum + 0.0);

System.out.println("Student SGPA is :" + sgpa);

??

class Main

{

public static void main (String[] args)

{

Student s1 = new Student();

s1.getDetails();

s1.printDetails();

s1.sgraCalc();

C:\Windows\system32\cmd.exe

```
C:\Users\admin\Documents>java Main
Enter no of subjects:
5
Enter Student usn
1BM19CS074
Enter Student name
Sharat
Enter credits followed by marks for subject 1:
4
90
Enter credits followed by marks for subject 2:
4
85
Enter credits followed by marks for subject 3:
3
78
Enter credits followed by marks for subject 4:
5
95
Enter credits followed by marks for subject 5:
5
64
Student details are as follows:
Name: Sharat      usn: 1BM19CS074
Sub1 Marks is: 90      Credit is: 4
Sub2 Marks is: 85      Credit is: 4
Sub3 Marks is: 78      Credit is: 3
Sub4 Marks is: 95      Credit is: 5
Sub5 Marks is: 64      Credit is: 5
Student's sgpa is: 8.80952380952381
```

**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.**

## LAB PROGRAM - 3

Ques 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. Include a toString() method that would display the complete detail of the Book. Develop a Java to create n book objects.

Info about toString() method:

```
class Test {
    int a = 10;
    public String toString() {
        return ("a=" + a);
    }
}
```

class TestMain {
 public static void main (String args) {
 Test t = new Test();
 System.out.println(t);
 }
}

Code:

```
import java.io.*;
import java.util.*;
class Book {
    private String name;
    private String author;
    private double p;
    private int np;
    Book() {
        name = null;
        author = null;
    }
}
```

```

p=0.0;
np=0;
}
void getdetails()
{
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter book title:");
    name = sc.next();
    System.out.print("Enter book author:");
    author = sc.next();
    System.out.println("Enter the book price:");
    p = sc.nextDouble();
    System.out.print("Enter the book pages:");
    np = sc.nextInt();
}

public String toString()
{
    return ("BOOK TITLE:" + name + " BOOK AUTHOR:" +
            author + " BOOK PRICE :" + p + " NUMBER OF PAGES:" +
            np);
}

class Main
{
    public static void main(String args[])
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of books");
        int n = sc.nextInt();
        Book b[] = new Book[n];
        for (int i = 0; i < n; i++)
        {
    }
}

```

System.out.print("Enter the book details of  
book no : " + (i+1));

BCI7 = new Book7();

BCI7.getdetails();

}

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

{

System.out.println("The book details of book " + (i+1));

System.out.println(BCI7);

}

}

```
C:\Users\admin\Documents>java Main
ENTER THE NUMBER OF BOOKS
2
Enter the Book details of book no:1
Enter book title:
Physics
Enter book author:
xyz
Enter the book price:
2000
Enter the book pages:
230
Enter the Book details of book no:2
Enter book title:
Chemistry
Enter book author:
sharat
Enter the book price:
3000
Enter the book pages:
34
The book details of book1
BOOK TITLE:Physics  BOOK AUTHOR:xyz  BOOK PRICE: 2000.0  NUMBER OF PAGES:230
The book details of book2
BOOK TITLE:Chemistry  BOOK AUTHOR:sharat  BOOK PRICE: 3000.0  NUMBER OF PAGES:34
```

**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.**

## LAB PROGRAM - 9

PAGE No.	
DATE	2/1/2020

- (i) 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 one of the classes extends the class shape. Each of the classes contains only method printArea() that prints the area of given shape.

```
Ans: import java.io.*;
import java.util.*;
abstract class Shape
{
    double x, y;
    Shape(double a, double b)
    {
        x=a;
        y=b;
    }
    abstract void printArea();
}
class Rectangle extends Shape
{
    Rectangle(double a, double b)
    {
        super(a,b);
    }
    void printArea()
    {
    }
```

```
System.out.println("Area is: "+(x*y))
```

```
}
```

```
}
```

```
class Triangle extends Shape
```

```
{
```

```
Triangle(double a, double b)
```

```
{
```

```
super(a,b)
```

```
}
```

```
void printArea()
```

```
{
```

```
System.out.println("Area is: "+(0.5*x*y))
```

```
}
```

```
class Circle extends Shape
```

```
{
```

```
Circle(double a, double b)
```

```
{
```

```
super(a,b)
```

```
}
```

```
void printArea()
```

```
{
```

```
System.out.println("Area is: "+(3.14*x*y))
```

```
}
```

```
class AbstractTest
```

```
{
```

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

```
Rectangle r1 = new Rectangle(10, 20);
```

```
Triangle t1 = new Triangle(3, 2);
```

```
Circle c1 = new Circle(12, 2);
```

```
rl.printArea();  
rl.printArea();  
cl.printArea();  
}  
}
```

C:\Users\admin\Documents>java Abstract\_test

Area is: 200.0

Area is: 2.0

Area is: 28.259999999999998

**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.

## 14B-Program 5

Q 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 but no cheque book facility. The current account provides cheque book facility but no interest. Current account holder should also maintain minimum balance and if 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 Current and Savings to make them more specific to the requirements. Include required methods to do the following tasks:- Display the balance, compute and deposit interest, permit withdrawal and update balance, check for minimum balance, impose penalty if needed, and update balance.

CODE:

```

import java.util.Scanner;
abstract class Account {
    String cname, acctype;
    long AcNo;
    double bal;
    final double minBal = 1000.0;
}

```

```

Account (String cname, long acno, double bal,
String acctype)
    {
    }

```

```
this.accNo = accNo;
```

```
this.cName = cName;
```

```
this.bal = bal
```

```
this.acctType = acctType;
```

```
}
```

```
abstract void addBal(double amt);
```

```
abstract void dispBal();
```

```
abstract void withdraw(double amt);
```

```
}
```

```
class Current extends Account
```

```
{
```

```
Current(string cname, long accNo, double bal)
```

```
{
```

```
super(cName, accNo, bal, "Current");
```

```
System.out.println("Name: " + cname +
```

```
"\taccNo: " + accNo + "\tbal: " + bal +
```

```
"\tType: " + acctType);
```

```
}
```

```
void addBal(double amt)
```

```
{
```

```
this.bal += amt;
```

```
}
```

```
void dispBal()
```

```
{
```

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

```
}
```

```
void withdraw(double amt)
```

```
{
```

```
if(this.bal == 0 || amt > this.bal)
```

```
{
```

```
System.out.println("Withdraw not  
possible!");
```

```
}
```

else

{

this.bal = amt;

checkBal();

}

}

void checkBal()

{

if (this.bal &lt; minBal)

{

this.bal -= this.bal \* 0.02;

?

}

?

class Sav-Acc extends Account

{

Sav-Acc (String cname, long accNo, double bal);

{

super(cName, accNo, bal, "Savings");  
System.out.println("name: " + cname + " ac-  
no: " + accNo + " bal: " + bal + " type:  
+ accType);

?

void addIntr()

{

this.bal += this.bal \* 0.07;

?

void dispBal()

{

System.out.println("Your balance is: " + this.bal)

?

void withdraw (double amt)

{

if (this.bal == 0.0 || amt > this.bal)

{

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

}

else

{

this.bal -= amt;

}

?

}

{

public static void main (String [] args)

{

Scanner sc = new Scanner (System.in);

Double amt;

int flag = 0;

while (flag == 0)

{

current ac

System.out.println ("1 : Add account \n 2 : Savings &  
displayBal \n default : exit");

int ch = sc.nextInt();

String name;

long accno;

double balans;

switch (ch)

{

case 1:

System.out.println ("Enter name, acc no,

```
initial balance is order: "";  
name = sc.next();  
acno = sc.nextLong();  
balan = sc.nextDouble();  
curr_acct c = new curr_acct(name, acno,  
balan);  
System.out.println("In Current - acct (n");  
int flag1 = 0;  
while (flag1 == 0)  
{  
    System.out.println("1: Add amount\n2:  
display Balance\n3: withdraw\nDefault:  
exit");  
    int ch1 = sc.nextInt();  
    while (ch1 != 1 & ch1 != 2 & ch1 != 3 & ch1 != 0)  
    {  
        System.out.println("Enter choice again");  
        ch1 = sc.nextInt();  
    }  
    switch (ch1)  
    {  
        case 1:  
            System.out.println("Enter amt to be added");  
            amt = sc.nextDouble();  
            c.addBal(amt);  
            break;  
        case 2:  
            c.dispBal();  
            break;  
        case 3:  
            System.out.println("Enter amt to be withdrawn");  
            amt = sc.nextDouble();  
            c.withBal(amt);  
            break;  
    }  
}
```

default:

flag1 = 1;

3

3

breaker

case 2:

System.out.println (" \n Savings acct (n)");

System.out.print ("Enter name, accno,  
initial balance in order ");

name = sc.next();

accno = sc.nextInt();

balan = sc.nextDouble();

Sav\_acct s = new Sav\_acct (name, accno,  
balan);

int flag2 = 0;

while (flag2 == 0)

System.out.println ("1: AddBal\n2: dispBal  
3: withdraw \nDefault: exit");

int ch2 = sc.nextInt();

switch (ch2)

{

case 1:

System.out.print ("enter amt to be added: ");

amt = sc.nextDouble();

s.addBal(amt);

break;

case 2:

s.dispBal();

break;

case 3:

System.out.print ("enter amt to be withdraw ");

amt = sc.nextDouble();

s. withdrawl amt);

break;

default:

flag2 = 1;

}

{

break;

default:

flag = 1;

}

{

{

{

}

```
C:\Users\admin\Documents>java Account_test
1:Current acc.
2:Savings acc.
default:exit
2

Savings_acct

Enter name, acc no, initial balance in order:
Sharat 12
3000
name: Sharat    accno: 12        bal: 3000.0      type: Savings
1:AddBal
2:displayBal
3:withdraw
default:exit
)1
enter amt to be added:
4000
1:AddBal
2:displayBal
3:withdraw
default:exit
3
```

