

LAB 2

Java program to find roots of quadratic equation

```

import java.util.Scanner;
import static java.lang.Math.*;
class Quadratic
{
    public static void main (String args[])
    {
        Scanner sc = new Scanner (System.in);
        System.out.print ("Enter the value of a : ");
        float a = sc.nextFloat();
        System.out.print ("Enter the value of b : ");
        float b = sc.nextFloat();
        System.out.print ("Enter the value of c : ");
        float c = sc.nextFloat();
        if (a == 0)
        {
            System.out.print ("Invalid! a cannot be zero");
        }
        else if ()
        {
            float d = b*b - 4*a*c;
            float sqrt_val = (float) Math.Sqrt (abs(d));
            float root1 = (-b + sqrt_val) / (2*a);
            float root2 = (-b - sqrt_val) / (2*a);
            if (d == 0)
                System.out.print ("Roots are real and equal : " + root1);
            else if (d > 0)
                System.out.print (" Roots are real and unique : ");
                System.out.print (root1 + "\n" + root2);
            }
            else
            {
                System.out.print ("Roots are 'imaginary'");
                System.out.print (b / (2*a) + "+i" + sqrt_val);
                System.out.print (-b / (2*a) + "-i" + sqrt_val);
            }
        }
    }
}

```

Algorithm

- 1) Get the input for values of a, b, c in expression $ax^2 + bx + c$;
- 2) Calculate the value of discriminant
$$d = \sqrt{b^2 - 4ac}$$
- 3) Calculate the value of the roots
$$\text{root 1} = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad \text{root 2} = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$
- 4) If value of a is zero print invalid statement and exit.
- 5) If d is equal to zero print that the roots are real and equal and also print the roots
- 6) If d is greater than zero print the roots with a message saying the roots are real and unequal/distinct
- 7) If d is less than zero print the roots with a message saying that the roots are imaginary

Output:

Enter value of a : 1

Enter value of b : 2

Enter value of c : 1

The roots are real and equal : -1.6

Enter value of a : 6

Enter value of b : 2

Enter value of c : 5

Invalid Input.

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.

Algorithm :-

- 1) Define student class and declare variables name, usn, array of marks and credits, sgpa.
- 2) Define function/method to read details from the user
- 3) Define function/method display details to display the user details and sgpa.
- 4) Define function/method to calculate grade points for each subject and thus calculate the sgpa using following formula

$$\text{sgpa} = (\sum \text{grade points} * \text{credits}) / (\sum \text{credits});$$

```
import java.util.Scanner;
class Student {
    String usn, name;
    int credits[] = new int[6];
    int marks[] = new int[6];
    int i;
    float sgpa;
    Scanner in = new Scanner(System.in);
    void read_details()
    {
        System.out.print("Enter your Name : ");
        name = in.nextLine();
        System.out.print("Enter your USN : ");
        usn = in.nextLine();
        System.out.println("Enter credits for each Sub:");
        for(i=0; i<6; i++)
        {
            System.out.print("Subject"+(i+1)+": ");
            credits[i] = in.nextInt();
        }
        for(i=0; i<6; i++)
        {
            System.out.print("Subject Marks: ");
            marks[i] = in.nextInt();
        }
    }
    void display_details()
    {
        System.out.println("Name : "+name);
        System.out.println("USN : "+usn);
        System.out.println("Credits ");
        for(i=0; i<6; i++)
        {
            System.out.println("Subject"+(i+1)+" : "
                + credits[i]);
        }
        System.out.println("Marks ");
    }
}
```

```
for(i=0; i<6; i++)
{ System.out.println("Subject "+(i+1)+  
"marks : "+marks[i]);}
```

{

```
calc_sgpa();
```

```
System.out.println("Sgpa : "+sgpa);
```

}

```
void calc_sgpa()
```

```
{ int sumgp=0, sumcred=0, gp;
```

```
for(i=0; i<6; 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 gp = 5;
```

```
sumgp = sumgp + (gp * credits[i]);
```

```
sumcred = sumcred + credits[i];
```

}

```
sgpa = (sumgp / sumcred);
```

}

```
class Student-main{
```

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

```
{ student s = new student();
```

```
s.read_details();
```

```
s.display_details();
```

}

}

Expected Output:

Enter your name : QWERTY

Enter your USN : 123

Credits for each subject:

Subject 1 : 2

Subject 2 : 4

Subject 3 : 4

Subject 4 : 4

Subject 5 : 4

Subject 6 : 5

Marks for each subject:

Subject 1 : 45

Subject 2 : 56

Subject 3 : 67

Subject 4 : 78

Subject 5 : 89

Subject 6 : 90

Name : QWERTY

USN : 123

~~Subject~~

Credits:

Subject 1 : 2

Subject 2 : 4

Subject 3 : 4

Subject 4 : 4

Subject 5 : 4

Subject 6 : 5

Marks: Subject 1 : 45

Subject 2 : 56

Subject 3 : 67

Subject 4 : 78

Subject 5 : 89

Subject 6 : 90

SgPA : 7.0

Lab - 3

Name: S Skanda
USN: IBM119CS137 Batch 2

Create a class book which contains four members:
name, author, price and num pages.
Include a constructor to set values for the members.
Include a `toString()` method that could display
the complete details of the book.

```

import java.util.Scanner;
class book{
    String title, author;
    int no_of_pages;
    float price;
    Scanner in = new Scanner(System.in);
    book()
    {
        System.out.print("Title:");
        title = in.nextLine();
        System.out.print("Author:");
        author = in.nextLine();
        System.out.print("Number of pages:");
        no_of_pages = in.nextInt();
        System.out.print("Price:");
        price = in.nextFloat();
    }
    public String toString()
    {
        return ("In Book title:" + title + "Author" +
                author + "Number of pages:" + no_of_pages +
                "Price:" + price);
    }
}

```

```
class books {
    public static void main(String[] args)
    {
        book[] b = new book[3];
        int i;
        for(i=0; i<3; i++)
        {
            System.out.println("Enter book " + (i+1) + " details:");
            b[i] = new book();
        }
        for(i=0; i<3; i++)
        {
            System.out.println("The book " + (i+1) + " details:");
            System.out.println(b[i]);
        }
    }
}
```

Output

Enter book 1 details

Title: 9

Author: a

Number of pages: 45

Price: 56

Enter book 2 details

Title: 6

Author: s

Number of pages: 23

Price: 12

The book 1 details:

Title: 9 Author: a Number of pages: 45 Price: 56

The book 2 details:

Title: 6 Author: s Number of pages: 23 Price: 12

Lab - 9

Name S Skande
USN : IBM19CS137

Batch - 2

To create abstract class shape with two integer members, and create three classes namely circle, rectangle and triangle to demonstrate ~~function~~ function overriding

abstract class shape {

int dim1;

int dim2;

shape (int a, int b) {

dim1 = a;

dim2 = b;

}

class Rectangle extends shape {

Rectangle (int a, int b) {

super (a, b);

}

void area () {

double area = dim1 * dim2;

System.out.println ("Area of Rectangle is " + area);

}

}

class Triangle extends shape {

Triangle (int a, int b) {

super (a, b);

}

void area () {

double area = dim1 * dim2 / 2;

System.out.println ("Area of Triangle is " + area);

}

}

```

class Circle extends Shape {
    Circle (int a, int b) {
        super (a, b);
    }
    void area() {
        double area = 3.14 * dim1 * dim1;
        System.out.println ("Area of circle is " + area);
    }
}

class Shapes {
    public static void main (String args[]) {
        Rectangle r = new Rectangle (7, 3);
        Triangle t = new Triangle (6, 8);
        Circle c = new Circle (5, 5);
        Shape s;
        s = r;
        s.area();
        s = t;
        s.area();
        s = c;
        s.area();
    }
}

```

Output

Area of rectangle is 21
 Area of triangle is 24
 Area of circle is 78.5

Name : S. Skanda
USN : 18M19CS137

Batch 2

To create Account class and two subclass savings account and current account

class account {

String name, acc-no, acc-type;
double balance;

account (String name, String acc-no, String acc-type)

this.name = name;

this.acc-no = acc-no;

this.acc-type = acc-type;

balance = 0;

}

void deposit (double amt) {

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

balance + = amount;

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

}

void withdraw (double amt) {

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

balance - = amount;

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

}

}

class curr-acct extends account {

curr-acct (String name, String acc-no, String acc-type)

super (name, acc-no, acc-type);

}

double service = 100;

double min-bal = 3000;

int charged = 0;

```
void check() {
    if (balance < min-bal && charged == 0) {
        balance -= service;
        charged = 1;
        System.out.println("Service deducted
                           due to low balance");
    }
    if (charged == 1)
        System.out.println("Your balance is low to
                           avoid being fined again increase your balance");
}
```

8

```
void dispbal() {
    check();
    System.out.println("Your balance is " + balance);
}

class sav acct extends account {
    sav acct (String name, String acc-no, String acc-type) {
        super (name, acc-no, acc-type);
    }

    int given = 0;

    void interest() {
        if (balance > 10000 && given == 0) {
            balance += 0.007 * balance;
            System.out.println("0.7% interest credited");
            given += 1;
        }

        if (balance > 100000 && given == 1) {
            balance += 0.005 * balance; given += 1;
            System.out.println("0.5% interest credited");
        }

        if (balance > 1000000 && given == 2) {
            balance += 0.002 * balance;
            System.out.println("0.2% interest credited");
            given += 1;
        }
    }
}
```

```
void disp-bal()
```

```
{ interest();
```

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

```
}
```

```
3
```

```
class bank {
```

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

```
{ Sav-acct sav = new Sav-acct('A', '1b', "Savings");
```

```
System.out.println("Savings account function: ");
```

```
sav.deposit(11000);
```

```
sav.disp-bal();
```

```
sav.withdraw(5000);
```

```
curr-acct cur = new curr-acct('B', '2b', "Current");
```

```
System.out.println("Current account functions: ");
```

```
cur.deposit(5000);
```

```
cur.withdraw(2500);
```

```
cur.disp-bal();
```

```
}
```

```
3
```

Output:

Savings account functions:

balance: 0.0

updated balance: 11000.0

your account has been credited

0.7% interest credited

your balance is 11077.0

balance: 11077.0

updated balance: 6047.0

Current account functions:

balance: 0.0

updated balance: 5000.0

balance: 5000.0

updated balance: 2500

100.0 deducted due to low balance

your balance is too low to avoid being fined again
increase your balance

Package program

File: student.java

```

package cie;
import java.util.*;
public class student
{
    public String usn, name;
    public int sem;
    public void read()
    {
        Scanner in = new Scanner (System.in);
        System.out.print("Enter usn of the student");
        usn = in.nextLine();
        System.out.print("Enter name of the student");
        name = in.nextLine();
        System.out.print("Enter semester of the student : ");
        sem = in.nextInt();
    }
}

```

File: internals.java

```

import java.util.*;
public class internals extends student
{
    public int[] cie_m = new int[3];
    public void read()
    {
        super.read();
        Scanner in = new Scanner (System.in);
        System.out.println("Enter the CIE marks:");
        for(int i=0; i<3; i++)
        {
            System.out.print("Enter course"+(i+1)+" marks:");
            cie_m[i] = in.nextInt();
        }
    }
    public void display()
    {
        System.out.println("USN of the student is "+usn);
        System.out.println("Name of student is "+name);
        System.out.println("Semester of student is "+sem);
    }
}

```

File externals.java

Package see;

```
import java.util.*;  
public class external extends cieo.student  
{ public int [] se-m = new int [3];  
    public int [] mar;  
    public void read()  
{ Scanner in = new Scanner (System.in);  
    System.out.println ("Enter SE Marks :");  
    for (int i=0; i<3; i++)  
    { System.out.print ("Enter course " + (i+1) + " Marks ");  
        se-m[i] = in.nextInt();  
    }  
}
```

3

main-stu.java

```
import java.util.*;  
import cieo.*;  
import see.*;  
public class stud-end  
{ public static void main (String [] args)  
{ int n;  
    Scanner s6 = new Scanner (System.in);  
    int final-mark;  
    System.out.print ("Enter the number of  
                      Students :");  
    n = s6.nextInt();  
    internal [] in = new internal [n];  
    external [] ex = new external [n];  
    internal ob1 = new internal ();  
    external ob2 = new external ();  
    ob2.mar = new int [n]
```

```

for (int i=0; i<n; i++)
{
    System.out.println ("Enter the details of student");
    in[i] = new internal();
    in[i].read();
    ex[i] = new external();
    ex[i].read();
}

System.out.print();
for (int i=0; i<n; i++)
{
    System.out.println ("Student "+(i+1)+" details");
    System.out.println ("USN : " + in[i].usn);
    System.out.println ("name : " + in[i].name);
    System.out.println ("Semester : " + in[i].sem);
}

for (int j=0; j<3; j++)
{
    finalmark = in[i].cie-m[j]+(ex[i].see-m[j]/2);
    System.out.println ("Final Mark of student "+(i+1)+" in course "+(j+1)+" is "+final-mark);
}

System.out.print();
}
}
}

```

WAP To demonstrate generics with multiple object parameters

```

import java.util.*;
class gen<T>
{
    T ob;
    gen(T o)
    {
        ob=o;
    }
    T getob()
    {
        return ob;
    }
    void showtype()
    {
        System.out.println("Type of T is "+ob.getClass().getName());
    }
}
class generic
{
    public static void main (String[] args)
    {
        String n;
        Scanner sc = new Scanner (System.in);
        System.out.println ("Enter Integer");
        n= sc.next();
        gen<Integer> ob1 = new gen<Integer>(Integer.parseInt(n));
        ob1.showtype();
        int val = ob1.getob();
        System.out.println ("Value is " + val);
        System.out.println();
        System.out.println ("Enter the string to be displayed");
        n= sc.next();
        gen<String> ob2 = new gen<String>(n);
        ob2.showtype();
    }
}

```

```
String x = ob2.getobj();
```

```
System.out.println ("Value : "+x);
```

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

```
System.out.print("Enter Double number ");
```

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

```
gen<Double> ob3 = new gen<Double>
```

```
(Double.parseDouble(n));
```

```
ob3.showtopel();
```

```
double ans = ob3.getobj();
```

```
System.out.println ("Value : "+ans);
```

```
}
```

```
}
```

```
b
```

Exception handling

```
import java.util.*;  
class WrongAge extends Exception{  
    int detail;  
    WrongAge (int a){  
        detail=a;  
    }  
    public String toString(){  
        return "enter correct age "+detail+" is invalid";  
    }  
}  
  
class Father{  
    public int age;  
    Scanner in = new Scanner (System.in);  
    father () throws WrongAge {  
        System.out.print ("Enter the father's age : ");  
        age = in.nextInt();  
        if (age < 0)  
            throw new WrongAge (age);  
    }  
}  
  
class Son extends Father {  
    Scanner in = new Scanner (System.in);  
    int fage;  
    son (Father f) throws WrongAge {  
        this.fage=f.age;  
        System.out.print ("Enter the son's age : ");  
        this.age = in.nextInt();  
        if (this.age < 0)  
            throw new WrongAge (age);  
        if (this.age > fage)  
            throw new WrongAge (age);  
    }  
}
```

```
3  
class ages{  
    public static void main (String [] args)  
    { try {  
        father f = new Father (2);  
        son s = new son (f);  
        3  
        catch (Exception e)  
        { System.out.println (e);  
        3  
    }  
}
```

Name: S Skanda
Usn: IBMACCS137
Multi-threaded program

```
class NewThread implements Runnable {  
    Thread t;  
    NewThread() {  
        t = new Thread(this, "DemoThread");  
        t.start();  
    }  
    public void run() {  
        try {  
            for (int i = 25; i > 0; i--) {  
                System.out.println("CSE");  
                Thread.sleep(2000);  
            }  
        } catch (InterruptedException e) {  
            System.out.println("Thread interrupted.");  
        }  
        System.out.println("Exiting thread.");  
    }  
}  
class Thread1 {  
    public static void main(String args[]) {  
        new NewThread();  
        try {  
            for (int i = 5; i > 0; i--) {  
                System.out.println("BMS College of Engineering");  
                Thread.sleep(10000);  
            }  
        } catch (InterruptedException e) {  
            System.out.println("Main thread interrupted.");  
        }  
        System.out.println("Main thread exiting.");  
    }  
}
```