

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.

```
/* 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.*;
import java.math.*;

class Quadratic
{
    int a,b,c;
    double r1,r2,d;

    void coeff()
    {
        Scanner s=new Scanner(System.in);
        System.out.println("Enter coefficients a,b,c");
        a = s.nextInt();
        b = s.nextInt();
        c = s.nextInt();
        d = (b*b) - (4*a*c);
    }

    void evalu()
    {
        while(a==0)
        {
            System.out.println("Not a QE.");
            System.out.println("Enter non zero coefficient");
            Scanner s = new Scanner(System.in);
            a = s.nextInt();
        }

        if(d==0)
        {
            System.out.println("Roots are real and equal.");
            r1 = (-b)/(2*a);
            System.out.println("Root1=Root2="+r1);
        }
    }
}
```

```

else if(d<0)
{
    System.out.println("Roots are imaginary");
    r1 = (-b)/(2*a);
    r2 = Math.sqrt(-d)/(2*a);
    System.out.println("Root1="+r1+"i"+r2);
    System.out.println("Root2="+r1+"-i"+r2);
}
else
{
    System.out.println("Roots are real and distinct");
    r1 = (-b+(Math.sqrt(d)))/(2*a);
    r2 = (-b-(Math.sqrt(d)))/(2*a);
    System.out.println("root1= "+r1+"root2= "+r2);
}
}

class QuadraticEq
{
    public static void main(String sx[])
    {
        Quadratic q = new Quadratic();
        q.coeff();
        q.eval();
    }
}

```

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

import java.util.Scanner;

class Student {
    String usn,name;
    int[] credits,marks;

    // Method to accept details of a student
    void acceptDetails() {
        Scanner hello = new Scanner(System.in);

        System.out.print("Enter USN: ");
        this.usn = hello.next();

        System.out.print("Enter Name: ");
        this.name = hello.next();

        credits = new int[4];
        marks = new int[4];

        System.out.println("Enter details of credits and marks in order for 4 subjects:");

        for (int i = 0; i < 4; i++) {
            System.out.print("Enter credits for Subject " + (i + 1) + ": ");
            credits[i] = hello.nextInt();

            System.out.print("Enter marks for Subject " + (i + 1) + ": ");
            marks[i] = hello.nextInt();
        }
    }

    // Method to display details of a student
    void display() {
        System.out.println("Student Details.");
        System.out.println("USN: " + usn);
        System.out.println("Name: " + name);
    }
}
```

```

        for (int i = 0; i < 4; i++) {
            System.out.println("Subject " + (i + 1) + " - Credits: " + credits[i] + ", Marks: "
+ marks[i]);
        }
    }

    // Method to calculate SGPA of a student
    double calc() {
        int totalCredits = 0;
        double totalGradePoints = 0;

        for (int i = 0; i < 4; i++) {
            totalCredits += credits[i];
            totalGradePoints += gradePoints(marks[i]) * credits[i];
        }

        return totalGradePoints / totalCredits;
    }

    // Method to calculate grade points based on marks
    int gradePoints(int marks) {
        if (marks >= 90) return 10;
        else if (marks >= 80) return 9;
        else if (marks >= 70) return 8;
        else if (marks >= 60) return 7;
        else if (marks >= 50) return 6;
        else if (marks >= 40) return 5;
        else return 0;
    }
}

public class SGPA {
    public static void main(String[] args) {
        // Example usage of the Student class
        Student student = new Student();
        student.acceptDetails();
        System.out.println("\nStudent Details:");
        student.display();
        System.out.println("\nSGPA: " + student.calc());
    }
}

```

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.

```
/* 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;
```

```
    String author;
```

```
    double price;
```

```
    int numPages;
```

```
    // Constructor to set values for the members
```

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

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

```
        this.author = author;
```

```
        this.price = price;
```

```
        this.numPages = numPages;
```

```
    }
```

```
    // Method to set details of the book
```

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

```
        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();
```

```

    }

    // Method to get details of the book
    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);
    }

    // toString method to display complete details of the book
    public String toString() {
        return "Book Details:\n" +
            "Name: " + name + "\n" +
            "Author: " + author + "\n" +
            "Price: $" + price + "\n" +
            "Number of Pages: " + numPages;
    }
}

public class Books {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the number of books: ");
        int n = scanner.nextInt();

        // Creating an array to store n book objects
        Book[] books = new Book[n];

        // Creating n book objects and setting their details
        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();
        }

        // Displaying details of all the books
        System.out.println("\nDetails of all books:");
        for (int i = 0; i < n; i++) {
            System.out.println("\nBook " + (i + 1) + ":");
            books[i].getDetails();
        }
    }
}

```

```
// Using the toString method to display complete details of the books
System.out.println("\nComplete details of all books:");
for (int i = 0; i < n; i++) {
    System.out.println("\nBook " + (i + 1) + ":\n" + books[i].toString());
}
}
```

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.

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 Cur-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:

- a) Accept deposit from customer and update the balance.
- b) Display the balance.
- c) Compute and deposit interest
- d) Permit withdrawal and update the balance

Check for the minimum balance, impose penalty if necessary and update the balance.

6. Create a package CIE which has two classes- Student and Internals. The class Student has members like usn, name, sem. The class internals has an array that stores the internal marks scored in five courses of the current semester of the student. Create another package SEE which has the class External which is a derived class of Student. This class has an array that stores the SEE marks scored in five courses of the current semester of the student. Import the two packages in a file that declares the final marks of n students in all five courses.

```
//file Student.java in Aish/CIE

package CIE;
import java.util.Scanner;

public class Student{
    public String usn;
    public String name;
    public int sem;

    public void accept(){
        Scanner s = new Scanner(System.in);
        System.out.println("Enter Name:");
        this.name = s.nextLine();
        System.out.println("Enter usn:");
        this.usn = s.nextLine();
        System.out.println("Enter sem");
        this.sem = s.nextInt();
    }

    public void display(){
        System.out.println("Name: " + this.name + "\nUSN: " + this.usn + "\nSem: " + this.sem);
    }
}
```



```
//file Internal.java in AIsh/CIE

package CIE;

import java.util.Scanner;

public class Internal extends CIE.Student{
    public int m[] = new int[5];
    CIE.Student student = new CIE.Student();
    public void accept(){
        student.accept();

        Scanner s1 = new Scanner(System.in);
        System.out.println("Enter Internal Marks:");
        for(int i=0;i<5;i++){
            m[i] = s1.nextInt();
        }
    }

    public void display(){

        student.display();
        for(int i=0;i<5;i++){
            System.out.println("Marks of sub" + (i+1) + " = " + m[i]);
        }
    }
}

```

```
//file External.java in AIsh/SEE

package SEE;

import java.util.Scanner;
import CIE.Internal;
import CIE.Student;

public class External extends CIE.Student{
    public int x[] = new int[5];
    public void accept(){
        Scanner s2 = new Scanner(System.in);
        System.out.println("Enter External Marks:");
        for(int i=0;i<5;i++){
            x[i] = s2.nextInt();
        }
    }

    public void display(){
        super.display();

        for(int i=0;i<5;i++){
            System.out.println("Marks of sub" + (i+1) + " = " + x[i]);
        }
    }
}

```

```
//file Final.java in Aish

import java.util.Scanner;

import CIE.Student;
import CIE.Internal;
import SEE.External;

public class Final{
    public static void main(String[] args) {
        Scanner n = new Scanner(System.in);

        System.out.println("Enter n:");
        int y = n.nextInt();

        CIE.Internal[] c1 = new CIE.Internal[y];
        SEE.External[] c2 = new SEE.External[y];

        for(int i=0;i<y;i++){
            c1[i] = new CIE.Internal();
            c2[i] = new SEE.External();

            c1[i].accept();
            c2[i].accept();

            // c1[i].accept();c2[i].accept();
            c1[i].display();c2[i].display();

            for(int j=0;j<5;j++){
                double calc = c1[i].m[j]+((c2[i].x[j])/2);
                System.out.println("Final marks of sub["+(i+1)+"]= "+calc);
            }
        }
    }
}
```

7. Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called "Father" and derived class called "Son" which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge( ) when the input age<0. In Son class, implement a constructor that cases both father and son's age and throws an exception if son's age is >=father's age.

```
/* Write a program that demonstrates handling of exceptions in inheritance tree. Create
a base class called "Father" and derived class called "Son" which extends the base class.
In Father class, implement a constructor which takes the age and throws the exception
WrongAge( ) when the input age<0. In Son class, implement a constructor that cases
both father and son's age and throws an exception if son's age is >=father's age. */
```

```
import java.util.Scanner;

class WrongAgeException extends Exception {
    public WrongAgeException(String message) {
        super(message);
    }
}

class Father {
    private int fatherAge;

    public Father(int age) throws WrongAgeException {
        if (age < 0) {
            throw new WrongAgeException("Age cannot be negative");
        }
        this.fatherAge = age;
    }
}

class Son extends Father {
    private int sonAge;
    public Son(int fatherAge, int sonAge) throws WrongAgeException {
        super(fatherAge);
        if (sonAge >= fatherAge) {
            throw new WrongAgeException("Son's age should be less than Father's age");
        }
        this.sonAge = sonAge;
        System.out.println("Father's Age: " + fatherAge);
        System.out.println("Son's Age: " + sonAge);
    }
}
```

```
public class ExceptionInheritanceDemo {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
  
        try {  
            System.out.print("Enter Father's Age: ");  
            int fatherAge = scanner.nextInt();  
            Father father = new Father(fatherAge);  
  
            System.out.print("Enter Son's Age: ");  
            int sonAge = scanner.nextInt();  
            Son son = new Son(fatherAge, sonAge);  
  
        } catch (WrongAgeException e) {  
            System.out.println("Exception: " + e.getMessage());  
        }  
    }  
}
```