**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 discriminate b 2 -4ac**

**is negative, display a message stating that there are no real solutions.**

import java.util.Scanner;

class Quad {

int a, b, c;

double root1, root2, d;

Scanner s = new Scanner(System.in);

void input() {

System.out.println("Quadratic equation is in the form: ax^2 + bx + c");

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

a = s.nextInt();

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

b = s.nextInt();

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

c = s.nextInt();

if (a == 0) {

System.out.println("Invalid Inputs");

System.exit(0);

}

}

void discriminant() {

d = (b \* b) - (4 \* a \* c);

}

void calculateRoots() {

if (d > 0) {

System.out.println("Roots are real and unequal:");

root1 = (-b + Math.sqrt(d)) / (2 \* a);

root2 = (-b - Math.sqrt(d)) / (2 \* a);

System.out.println("First root is: " + root1);

System.out.println("Second root is: " + root2);

} else if (d == 0) {

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

root1 = -b / (2 \* a);

System.out.println("Root: " + root1);

} else {

System.out.println("No real solutions. Roots are imaginary.");

double real = -b / (2 \* a);

double imaginary = Math.sqrt(-d) / (2 \* a);

System.out.println("The equation has two complex roots: " + real + " + " + imaginary + "i and " + real + " - " + imaginary + "i");

}

}

}

public class Main {

public static void main(String[] args) {

Quad q = new Quad();

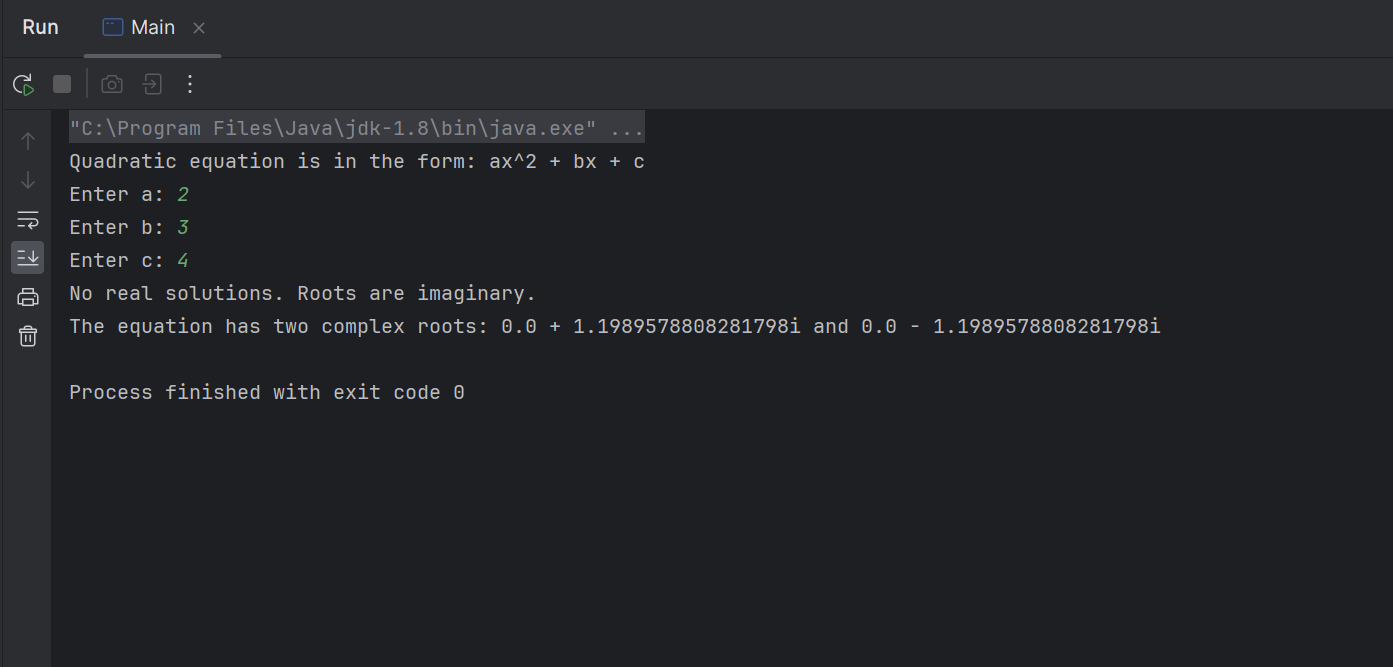
q.input();

q.discriminant();

q.calculateRoots();

}

}



**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.**

import java.util.Scanner;

class Student {

String USN;

String name;

double[] marks = new double[6];

public void acceptDetails() {

Scanner s = new Scanner(System.in);

System.out.println("Enter USN:");

USN = s.nextLine();

System.out.println("Enter name:");

name = s.nextLine();

System.out.println("Enter marks for 6 subjects:");

for (int i = 0; i < 6; i++) {

System.out.print("Subject " + (i + 1) + ": ");

marks[i] = s.nextDouble();

}

}

public void displayDetails() {

System.out.println("USN: " + USN);

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

System.out.println("SGPA: " + calculateSGPA());

}

public double calculatePercentage() {

double totalMarks = 0;

for (int i = 0; i < 6; i++) {

totalMarks += marks[i];

}

return (totalMarks / 6.0);

}

public double calculateSGPA() {

double percentage = calculatePercentage();

double sgpa;

if (percentage >= 90)

sgpa = 10.0;

else if (percentage >= 80)

sgpa = 9.0;

else if (percentage >= 70)

sgpa = 8.0;

else if (percentage >= 60)

sgpa = 7.0;

else if (percentage >= 50)

sgpa = 6.0;

else

sgpa = 0.0;

return sgpa;

}

}

public class Main {

public static void main(String[] args) {

Scanner s = new Scanner(System.in);

System.out.println("Enter the number of Students:");

int num = s.nextInt();

Student[] stu = new Student[num];

for (int i = 0; i < num; i++) {

System.out.println("Enter details for student " + (i + 1) + ":");

stu[i] = new Student();

stu[i].acceptDetails();

}

System.out.println("Student details:");

for (int i = 0; i < num; i++) {

System.out.println("Details for student " + (i + 1) + ":");

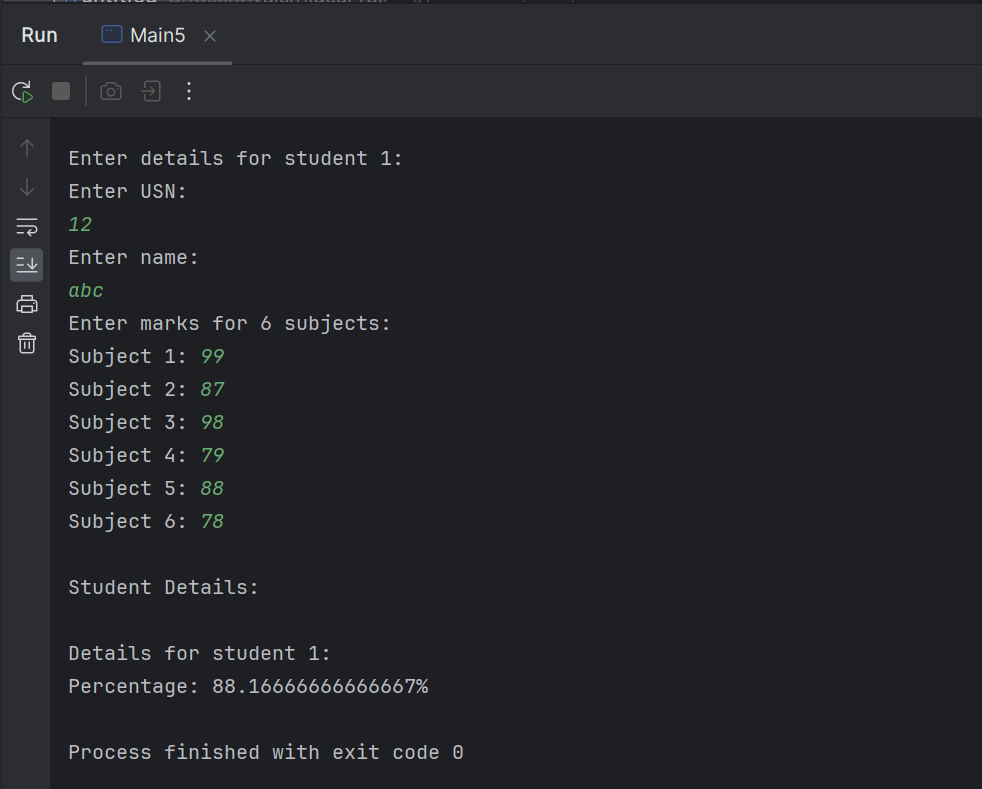
stu[i].displayDetails();

System.out.println();

}

}

}



**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.**

import java.util.Scanner;

class Book {

String name, author;

int price, numPages;

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

this.name = name;

this.author = author;

this.price = price;

this.numPages = numPages;

}

public String toString() {

return "Book name: " + name + "\nAuthor: " + author + "\nPrice: " + price + "\nNumber of pages: " + numPages;

}

}

public class Main {

public static void main(String[] args) {

Scanner s = new Scanner(System.in);

System.out.println("Enter the number of books:");

int n = s.nextInt();

Book[] books = new Book[n];

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

System.out.println("Enter details for Book " + (i+1) + ": ");

System.out.print("Enter Book name: ");

String name = s.nextLine();

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

String author = s.nextLine();

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

int price = s.nextInt();

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

int numPages = s.nextInt();

books[i] = new Book(name, author, price, numPages);

}

System.out.println("Book Details:");

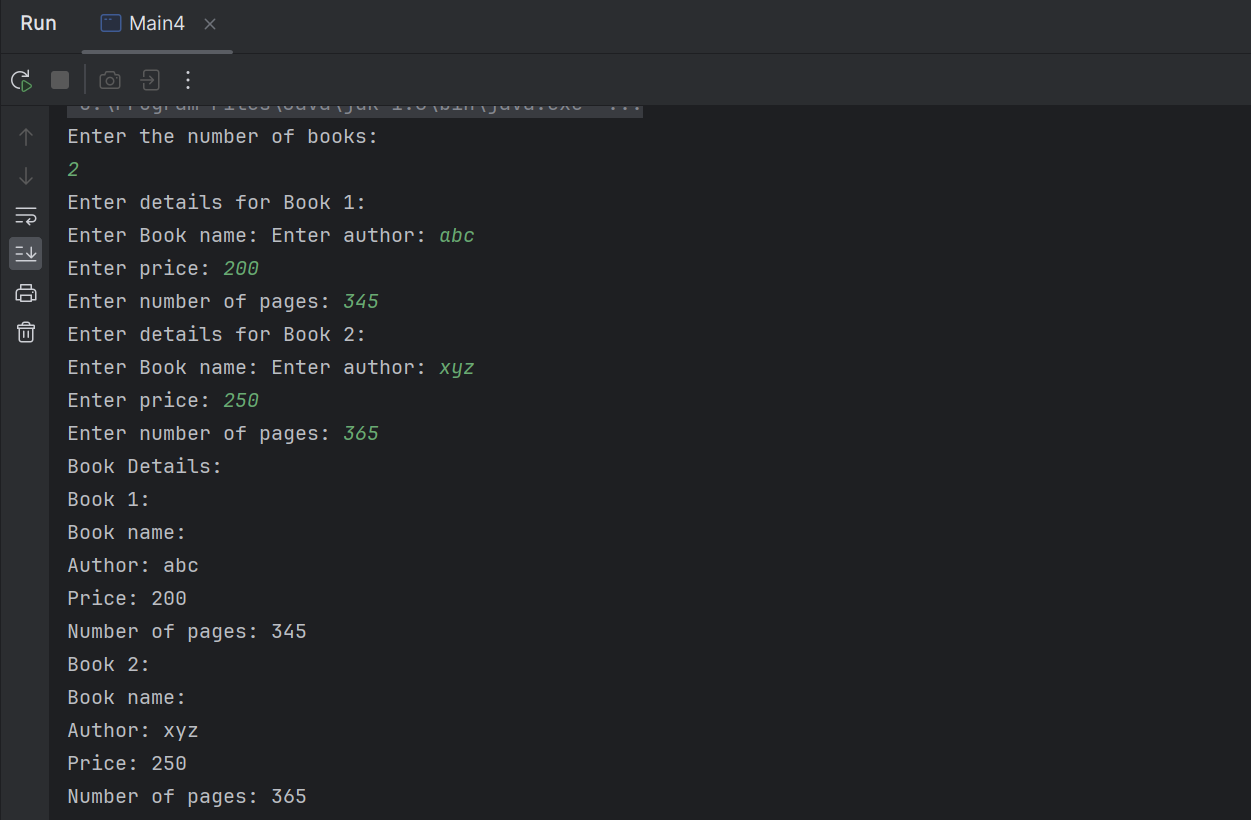
for(int i = 0; i < n; i++) {

System.out.println("Book " + (i+1) + ":");

System.out.println(books[i]);

}

}

} 

**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.**

import java.util.Scanner;

abstract class Shape {

double a, b;

abstract void printArea();

}

class Rectangle extends Shape {

void printArea(double length, double breadth) {

a = length;

b = breadth;

double area = a \* b;

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

}

}

class Triangle extends Shape {

void printArea(double base, double height) {

a = base;

b = height;

double area = 0.5 \* a \* b;

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

}

}

class Circle extends Shape {

void printArea(double radius, double dummy) {

a = radius;

double area = 3.14 \* a \* a;

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

}

}

public class Main {

public static void main(String[] args) {

Scanner s = new Scanner(System.in);

System.out.println("Enter length and breadth of rectangle:");

double length = s.nextDouble();

double breadth = s.nextDouble();

Rectangle rectangle = new Rectangle();

rectangle.printArea(length, breadth);

System.out.println("Enter base and height of triangle:");

double base = s.nextDouble();

double height = s.nextDouble();

Triangle triangle = new Triangle();

triangle.printArea(base, height);

System.out.println("Enter radius of circle:");

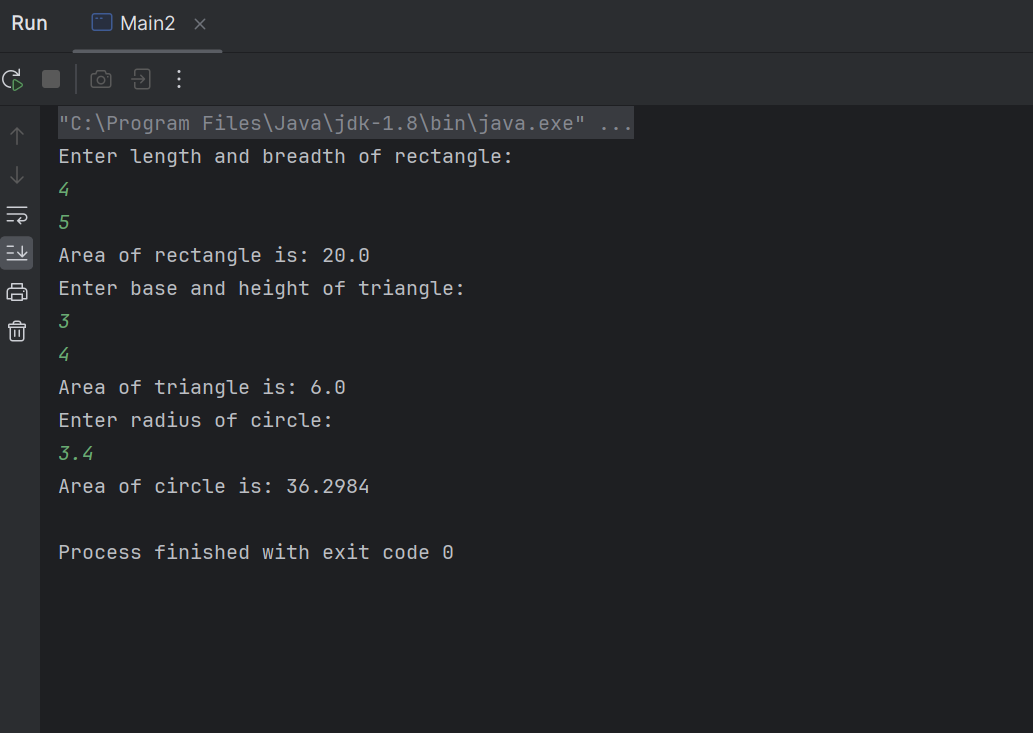
double radius = s.nextDouble();

Circle circle = new Circle();

circle.printArea(radius, 0);

}

}

****

**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.**

import java.util.Scanner;

class Account {

String customerName;

int accountNumber;

String accountType;

double balance;

public Account(String customerName, int accountNumber, String accountType, double balance) {

this.customerName = customerName;

this.accountNumber = accountNumber;

this.accountType = accountType;

this.balance = balance;

}

public void deposit(double amount) {

balance += amount;

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

}

// Method to display balance

public void displayBalance() {

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

}

}

class CurAcct extends Account {

double minBalance;

double serviceCharge;

public CurAcct(String customerName, int accountNumber, double balance, double minBalance, double serviceCharge) {

super(customerName, accountNumber, "Current", balance);

this.minBalance = minBalance;

this.serviceCharge = serviceCharge;

}

public void checkMinBalance() {

if (balance < minBalance) {

balance -= serviceCharge;

System.out.println("Minimum balance not maintained. Service charge imposed. Updated balance: " + balance);

}

}

}

class SavAcct extends Account {

double interestRate;

public SavAcct(String customerName, int accountNumber, double balance, double interestRate) {

super(customerName, accountNumber, "Savings", balance);

this.interestRate = interestRate;

}

public void computeAndDepositInterest() {

double interest = balance \* interestRate / 100;

balance += interest;

System.out.println("Interest computed and deposited. Updated balance: " + balance);

}

public void withdraw(double amount) {

if (amount <= balance) {

balance -= amount;

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

} else {

System.out.println("Insufficient funds for withdrawal.");

}

}

}

public class Bank {

public static void main(String[] args) {

// Example usage

CurAcct currentAccount = new CurAcct("abc", 12345, 1000.0, 500.0, 10.0);

currentAccount.deposit(500.0);

currentAccount.checkMinBalance();

currentAccount.displayBalance();

SavAcct savingsAccount = new SavAcct("xyz", 67890, 2000.0, 5.0);

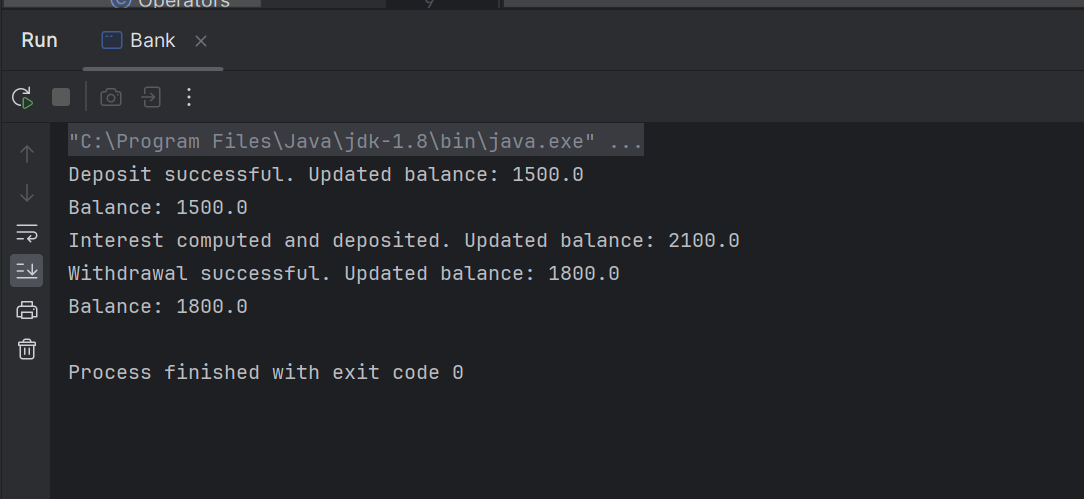
savingsAccount.computeAndDepositInterest();

savingsAccount.withdraw(300.0);

savingsAccount.displayBalance();

}

}



**6.Create a package CIE which has two classes- Student and Internals. The class Personal**

**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.**

package CIE;

public class Student

{

public String name,usn;

public int sem;

public Student(String usn,String name,int sem)

{

this.usn=usn;

this.name=name;

this.sem=sem;

}

}

package CIE;

public class Internals

{

public int[] marks = new int[5];

public Internals(int[] marks)

{

this.marks=marks;

}

}

package SEE;

import CIE.Student;

public class Externals extends Student

{

public int[] seemarks = new int[5];

public Externals(String usn,String name,int sem,int[] seemarks)

{

super(usn,name,sem);

this.seemarks=seemarks;

}

}import java.util.Scanner;

import CIE.Student;

import CIE.Internals;

import SEE.Externals;

public class Final

{

public static void main(String[] args)

{

int n,sem,sum=0;

String name,usn;

int[] marks = new int[5];

int[] seemarks = new int[5];

Scanner sc = new Scanner(System.in);

System.out.println("Enter no. of students:");

n=sc.nextInt();

Student[] s = new Student[n];

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

{

System.out.println("Enter usn for student " + (i+1));

usn=sc.nextLine();

sc.next();

System.out.println("Enter name of student " + (i+1));

name=sc.nextLine();

sc.next();

System.out.println("Enter sem for student " + (i+1));

sem=sc.nextInt();

System.out.println("Enter Internal marks of student " + (i+1));

for(int j=0;j<5;j++)

{

marks[j]=sc.nextInt();

}

System.out.println("Enter External marks of student " + (i+1));

for(int j=0;j<5;j++)

{

seemarks[j]=sc.nextInt();

}

s[i] = new Student(usn,name,sem);

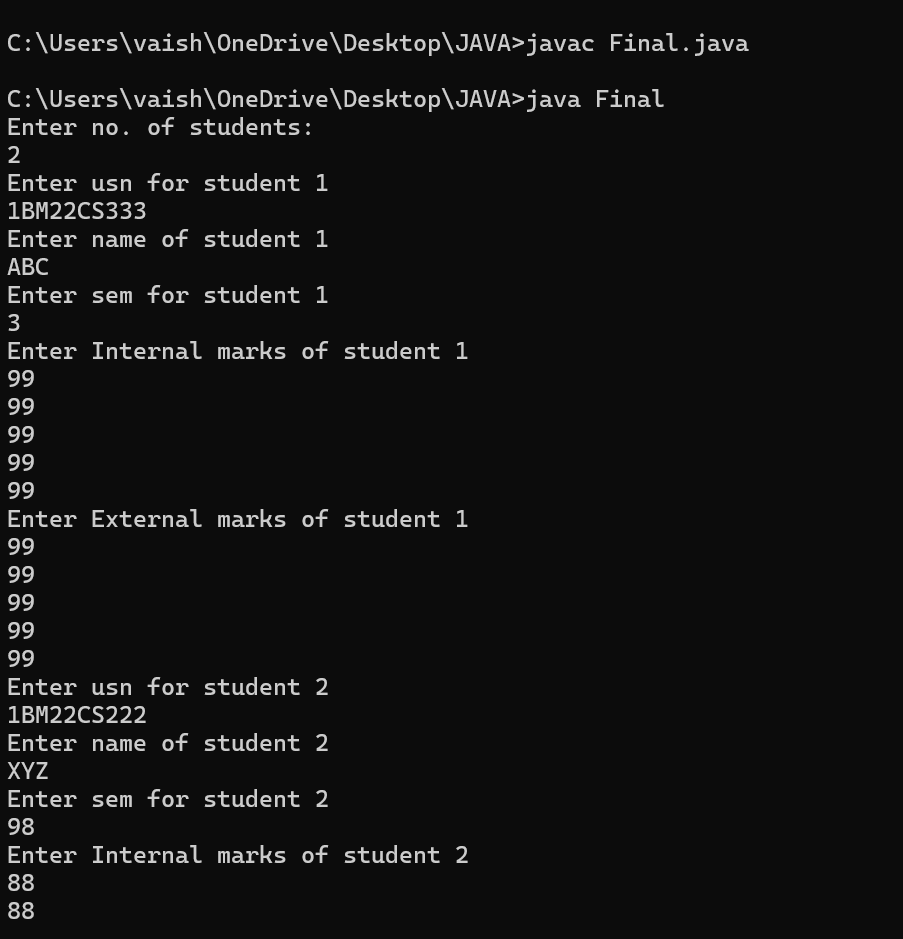
Internals it=new Internals(marks);

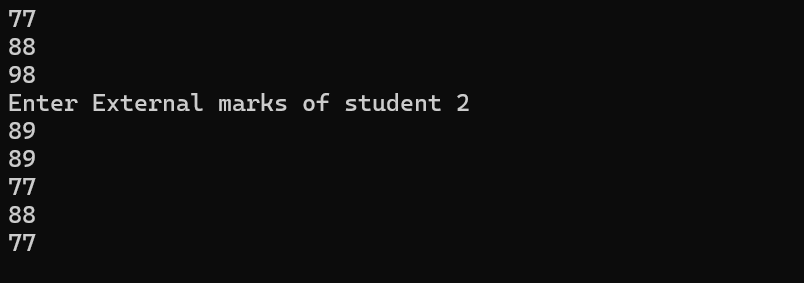
Externals e = new Externals(usn,name,sem,seemarks);

}

}

}

****

****

**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&lt;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.**

class WrongAge extends Exception {

public WrongAge(String message) {

super(message);

}

}

class Father {

int fAge;

public Father(int fAge) throws WrongAge {

if (fAge < 0)

throw new WrongAge("Invalid age input");

this.fAge = fAge;

}

}

class Son extends Father {

int sonAge;

public Son(int fAge, int sonAge) throws WrongAge {

super(fAge);

if (sonAge >= fAge)

throw new WrongAge("Son's age cannot be greater than or equal to father's age");

this.sonAge = sonAge;

}

}

public class Main {

public static void main(String[] args) {

try {

Father father = new Father(50);

Son son = new Son(50, 20);

System.out.println("Father's age: " + father.fAge);

System.out.println("Son's age: " + son.sonAge);

} catch (WrongAge e) {

System.out.println(e.getMessage());

}

}

}



**8.Write a program which creates two threads, one thread displaying “BMS College of**

**Engineering” once every ten seconds and another displaying “CSE” once every two**

**seconds.**

class One extends Thread {

public void run() {

for (int i = 0; i < 5; i++) {

System.out.println("BMS College of Engineering");

try {

Thread.sleep(10000);

} catch (InterruptedException e) {

System.out.println(e);

}

}

}

}

class Two extends Thread {

public void run() {

for (int i = 0; i < 5; i++) {

System.out.println("CSE");

try {

Thread.sleep(2000);

} catch (InterruptedException e) {

System.out.println(e);

}

}

}

}

public class Main {

public static void main(String[] args) {

One one = new One();

Two two = new Two();

one.start();

two.start();

}

}

