DIGITAL ASSIGNMENT3

JAVA CSE1007

SEPTEMBER 27, 2021

ANISH SHRESTHA
20BCE2893

CSE1007- Java Programming

Inheritance

1. A training centre conducts a total of 7 tests for its students. Students are allowed to skip few tests. Let there be 25 students in the batch. So in the main class for every student, read the number of tests taken and the marks scored in each test. A class 'TestDetails' should be defined with a 2D array of float type to store the marks of all the students. Define a method 'storeMarks()' that will receive the following details for every student from the main class and create in the 2D array, those many columns equal to the number of tests, so as to store the marks. There is no need to store the number of tests. Define another method 'displayMarks()' to print the details.

   Also the training centre wishes to keep those students in notice period who have taken < 3 tests and those who have not scored ≥ 50 in at least 3 tests. Derive another class 'NoticePeriod' from 'TestDetails' that includes a method to count and print the number of students in bench. Also it should print the ID of those students assuming the row index of the array to be their ID. While checking do not proceed to check the marks in all tests, if the student has already scored more than 50 in 3 tests. Instantiate this class from the main class and do the required processing.

2. Create an inheritance hierarchy in java using following information given below that a bank might use to represent customers' bank accounts.

Base class **Account** should include one data member of type double to represent account balance. The class should provide constructor that receives an initial balance and uses it to initialize the data member.The constructor should validate the initial balance to ensure that it is greater than or equal to 0.If not the balance is set to 0.0 and the constructor should display an error message,indicating that the initial balance was invalid. The class also provides three member functions **credit, debit**(debit amount should not exceed the account balance) and **enquiry**.Derived class **SavingsAccount** should inherit the functionality of an Account, but also include data member of type double indicating the interest rate assigned to the Account. SavingsAccount constructor should receive the initial balance, as well as an initial value for SavingsAccount's interest rate. SavingsAccount should provide public member function **calculateInterest** that returns double indicating the amount of interest earned by an account.The method **calculateInterest** should determine this amount by multiplying the interest rate by the account balance.SavingsAccount function should inherit member functions credit,debit and enquiry without redefining them. Derived class **CheckingAccount** should inherit the functionality of an Account, but also include data member of type double that represents the fee charged per transaction. CheckingAccount constructor should receive the initial balance, as well as parameter indicating fee amount. class CheckingAccount should redefine credit and debit function so that they subtract the fee from account balance whenever either transaction is performed. CheckingAccount's debit function should charge a fee only if the money is actually withdrawn (debit amount should not exceed the account balance).After defining the class hierarchy, write program that creates object of each class and tests their member functions. Add interest to SavingAccount object by first invoking its calculateInterest function, then passing the returned interest amount to object's credit function.

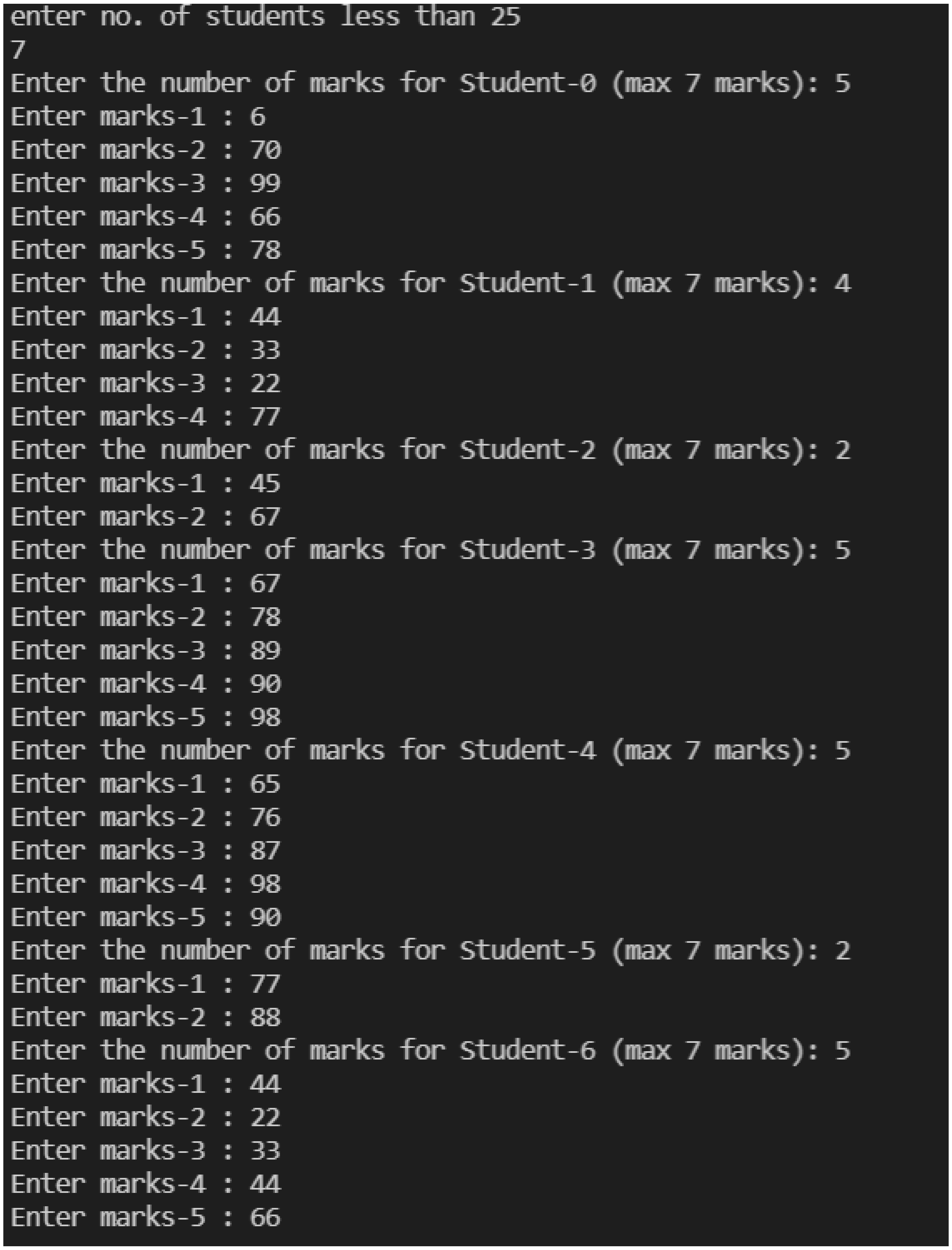
Question1:

Code:

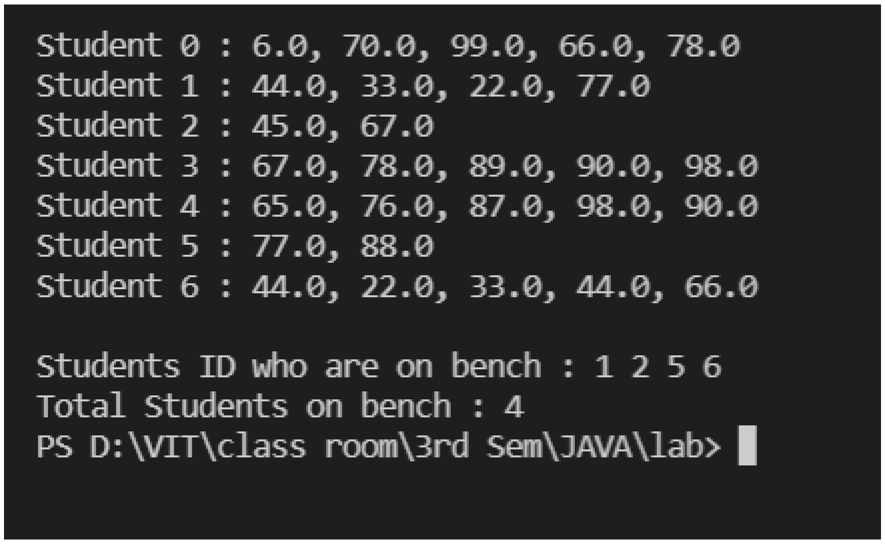
```
import java.util.*;

class TestDetails {
    protected float marks[][];

    TestDetails() {
        marks = null;
    }

    public void storeMarks(float inMarks[][]) {
        marks = new float[inMarks.length][];
        for (int i = 0; i < inMarks.length; i++) {
            marks[i] = new float[inMarks[i].length];
            for (int j = 0; j < inMarks[i].length; j++)
                marks[i][j] = inMarks[i][j];
        }
    }

    public void displayMarks() {
        for (int i = 0; i < marks.length; i++) {
            System.out.print("\nStudent " + i + " : ");
            for (int j = 0; j < marks[i].length - 1; j++)
                System.out.print(marks[i][j] + ", ");
            System.out.print(marks[i][marks[i].length - 1]);
        }
        System.out.println();
    }
}

class NoticePeriod extends TestDetails {
    public void printBench() {
        System.out.print("Students ID who are on bench : ");
        int benchStudents = 0;
        int countAbove50;
        for (int i = 0; i < marks.length; i++) {
            if (marks[i].length < 3) {
                benchStudents++;
                System.out.print(i + " ");
            } else {
                countAbove50 = 0;
                for (int j = 0; j < marks[i].length; j++) {
                    if (marks[i][j] >= 50)
                        countAbove50++;
```

```java
                    if (countAbove50 >= 3)
                        break;
                }
                if (countAbove50 < 3) {
                    System.out.print(i + " ");
                    benchStudents++;
                }
            }
        }
        System.out.println("\nTotal Students on bench : " + benchStudents);
    }
}

public class Training {

    public static void main(String[] args)

    {

        NoticePeriod notice = new NoticePeriod();
        Scanner scan = new Scanner(System.in);
        System.out.println("enter no. of students less than 25");
        float marks[][] = new float[scan.nextInt()][];

        for (int i = 0; i < marks.length; i++) {
            System.out.print("Enter the number of marks for Student-
" + i + " (max 7 marks): ");
            int numMarks = scan.nextInt();
            marks[i] = new float[numMarks];
            for (int j = 0; j < marks[i].length; j++) {
                System.out.print("Enter marks-" + (j + 1) + " : ");
                marks[i][j] = scan.nextFloat();

            }
        }
        notice.storeMarks(marks);
        notice.displayMarks();
        System.out.println();
        notice.printBench();
    }

}
```

Output:



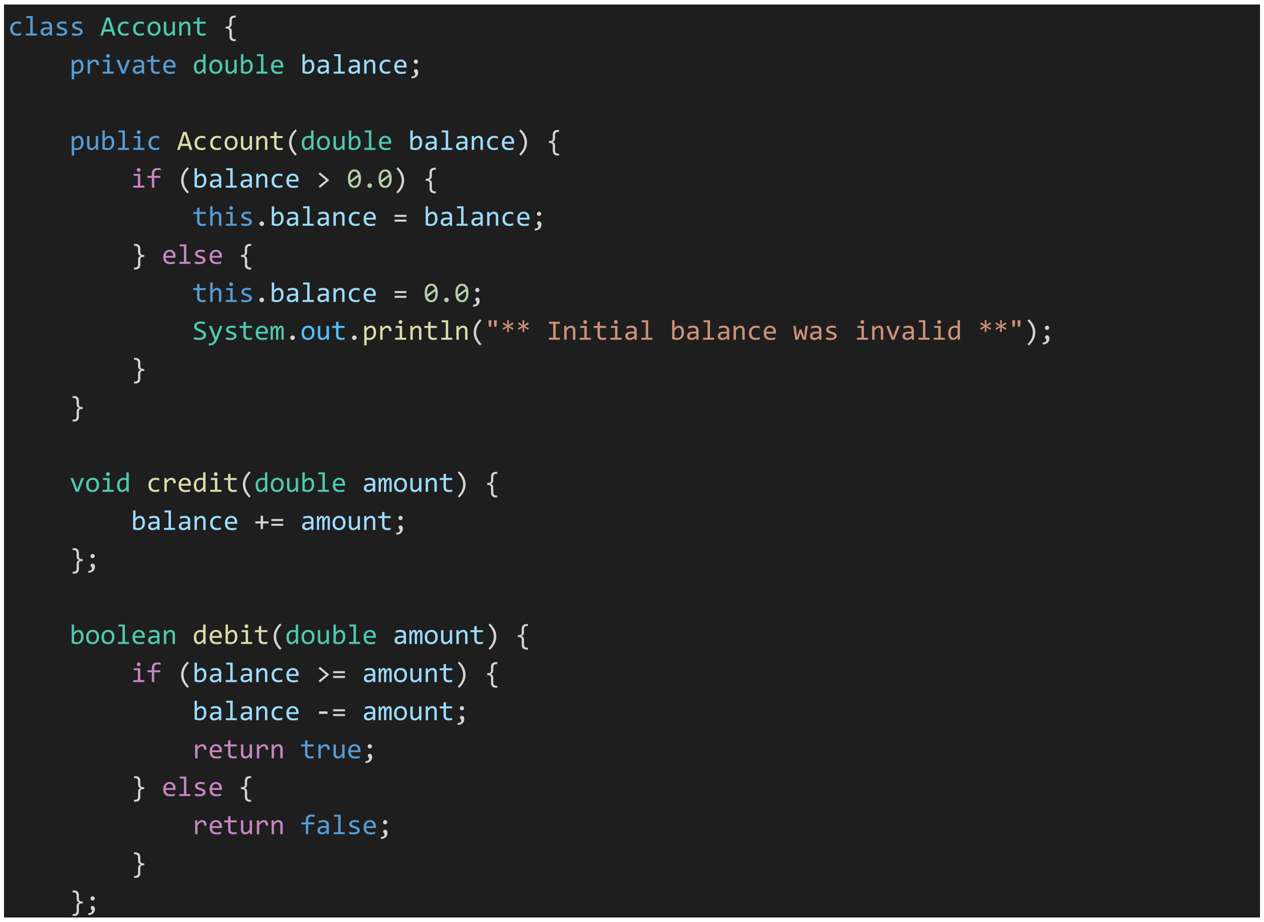
```
enter no. of students less than 25
7
Enter the number of marks for Student-0 (max 7 marks): 5
Enter marks-1 : 6
Enter marks-2 : 70
Enter marks-3 : 99
Enter marks-4 : 66
Enter marks-5 : 78
Enter the number of marks for Student-1 (max 7 marks): 4
Enter marks-1 : 44
Enter marks-2 : 33
Enter marks-3 : 22
Enter marks-4 : 77
Enter the number of marks for Student-2 (max 7 marks): 2
Enter marks-1 : 45
Enter marks-2 : 67
Enter the number of marks for Student-3 (max 7 marks): 5
Enter marks-1 : 67
Enter marks-2 : 78
Enter marks-3 : 89
Enter marks-4 : 90
Enter marks-5 : 98
Enter the number of marks for Student-4 (max 7 marks): 5
Enter marks-1 : 65
Enter marks-2 : 76
Enter marks-3 : 87
Enter marks-4 : 98
Enter marks-5 : 90
Enter the number of marks for Student-5 (max 7 marks): 2
Enter marks-1 : 77
Enter marks-2 : 88
Enter the number of marks for Student-6 (max 7 marks): 5
Enter marks-1 : 44
Enter marks-2 : 22
Enter marks-3 : 33
Enter marks-4 : 44
Enter marks-5 : 66
```



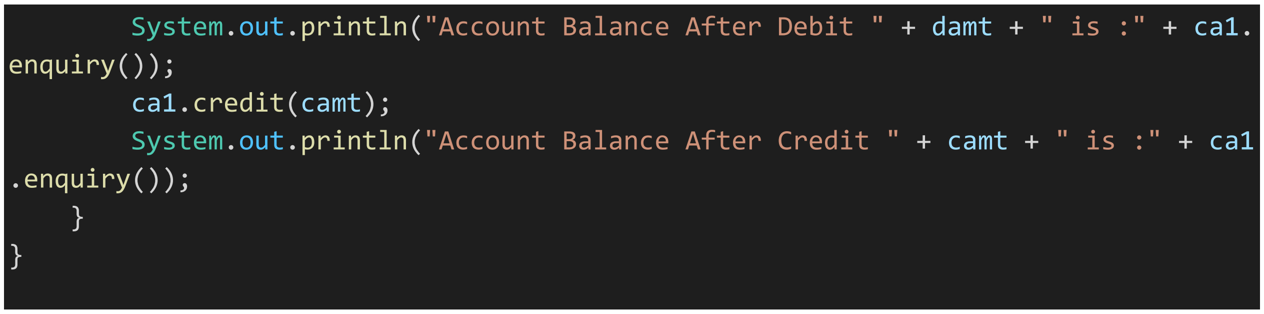
```
Student 0 : 6.0, 70.0, 99.0, 66.0, 78.0
Student 1 : 44.0, 33.0, 22.0, 77.0
Student 2 : 45.0, 67.0
Student 3 : 67.0, 78.0, 89.0, 90.0, 98.0
Student 4 : 65.0, 76.0, 87.0, 98.0, 90.0
Student 5 : 77.0, 88.0
Student 6 : 44.0, 22.0, 33.0, 44.0, 66.0

Students ID who are on bench : 1 2 5 6
Total Students on bench : 4
PS D:\VIT\class room\3rd Sem\JAVA\lab>
```

Question2:

Code:

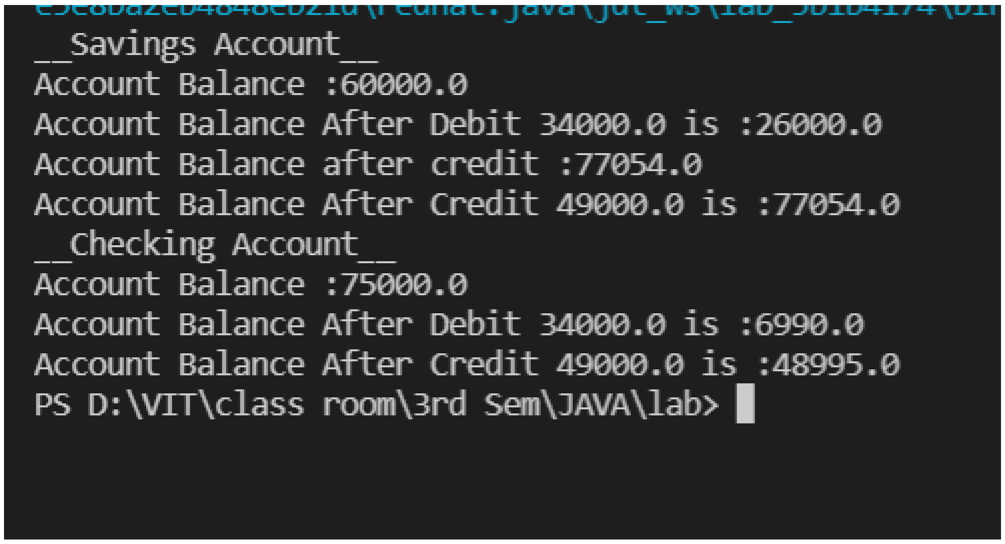


```java
class Account {
    private double balance;

    public Account(double balance) {
        if (balance > 0.0) {
            this.balance = balance;
        } else {
            this.balance = 0.0;
            System.out.println("** Initial balance was invalid **");
        }
    }

    void credit(double amount) {
        balance += amount;
    };

    boolean debit(double amount) {
        if (balance >= amount) {
            balance -= amount;
            return true;
        } else {
            return false;
        }
    };
    double enquiry() {
        return this.balance;
    };

    void setBalance(double amt) {
        this.balance = amt;
    };
};

class SavingsAccount extends Account {
    private double rate;

    public SavingsAccount(double balance, double rate) {
        super(balance);
        this.rate = rate;
    }
    double calculateInterest() {
        return enquiry() * (rate / 100);
    };

    void credit(double amount) {
        double interest = calculateInterest();
        setBalance(enquiry() + amount + interest);
        System.out.println("Account Balance after credit :" + enquiry());
    };

    boolean debit(double amount) {
        if (amount > enquiry())
        {
            System.out.println("Debit amount exceeded account balance.");
            return false;
        }
        else
        {
            setBalance(enquiry() - amount);
            return true;
        }
    };
}

// Creating a Checking Account class derivied from Account class

class CheckingAccount extends Account {
    private double feePerTx;
    public CheckingAccount(double balance, double feePerTx) {
        super(balance);
        this.feePerTx = feePerTx;
    }

    void credit(double amount) {
        setBalance(amount - feePerTx);
    };

    boolean debit(double amount) {
        double amt = (amount + feePerTx);
        if (!super.debit(amt))
        {
            System.out.println("Debit amount exceeded account balance.");
            return false;
        }
        else
        {
            super.debit(amt);
            return true;
        }
    };
};

public class Bank {

    public static void main(String[] args) {
        double damt, camt;
        SavingsAccount sa1 = new SavingsAccount(60000, 7.9);
        CheckingAccount ca1 = new CheckingAccount(75000, 5);
        damt = 34000;
        camt = 49000;

        System.out.println("__Savings Account__");
        System.out.println("Account Balance :" + sa1.enquiry());
        sa1.debit(damt);
        System.out.println("Account Balance After Debit " + damt + " is :" + sa1.
enquiry());
        sa1.credit(camt);
        System.out.println("Account Balance After Credit " + camt + " is :" + sa1
.enquiry());
        System.out.println("__Checking Account__");
        System.out.println("Account Balance :" + ca1.enquiry());
        ca1.debit(damt);
```



```
        System.out.println("Account Balance After Debit " + damt + " is :" + ca1.
enquiry());
        ca1.credit(camt);
        System.out.println("Account Balance After Credit " + camt + " is :" + ca1
.enquiry());
    }
}
```

Output:



```
__Savings Account__
Account Balance :60000.0
Account Balance After Debit 34000.0 is :26000.0
Account Balance after credit :77054.0
Account Balance After Credit 49000.0 is :77054.0
__Checking Account__
Account Balance :75000.0
Account Balance After Debit 34000.0 is :6990.0
Account Balance After Credit 49000.0 is :48995.0
PS D:\VIT\class room\3rd Sem\JAVA\lab>
```

Program Based on Interface and abstract class

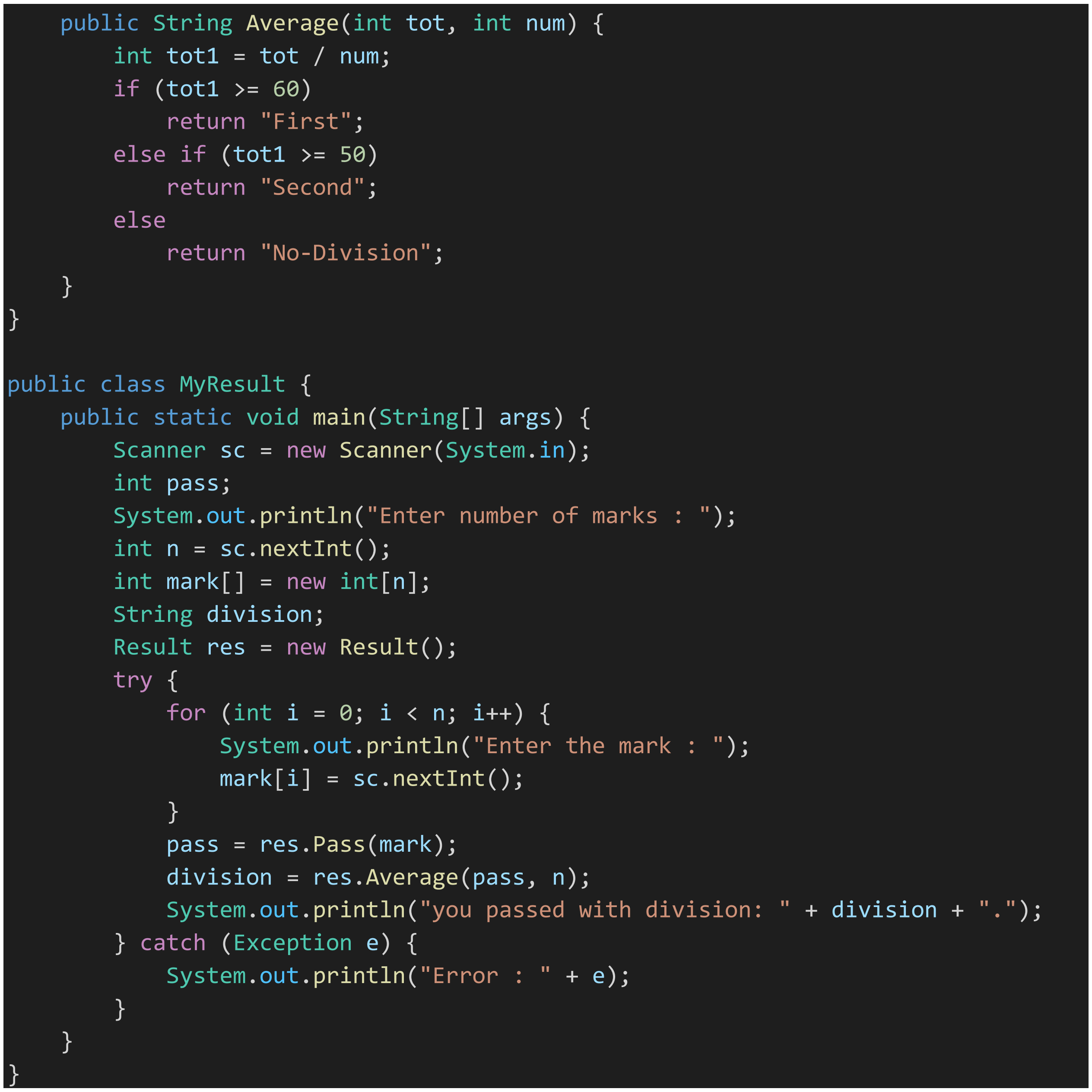
1. Write an interface called Exam with a method Pass ( ) that returns the total marks. Write another interface called Classify with a method Average (int total) which returns a string. Write a Class called Result which implements both Exam and Classify. The Pass method should get the marks from the user and finds the total marks and return. The Division method calculate the average marks and return "First" if the average is 60 or more, "SECOND" when average is 50 or more but below 60, "NO DIVISION" when average is less than 50

2. Write an abstract class special with an abstract method double Process (double P,double R). Create a subclass Discount and implement the Process() method with the following formula: net=P-P*R/100. Return the Process() method with the following formula: total=P+P*R/100. Return the total.

Question 1:

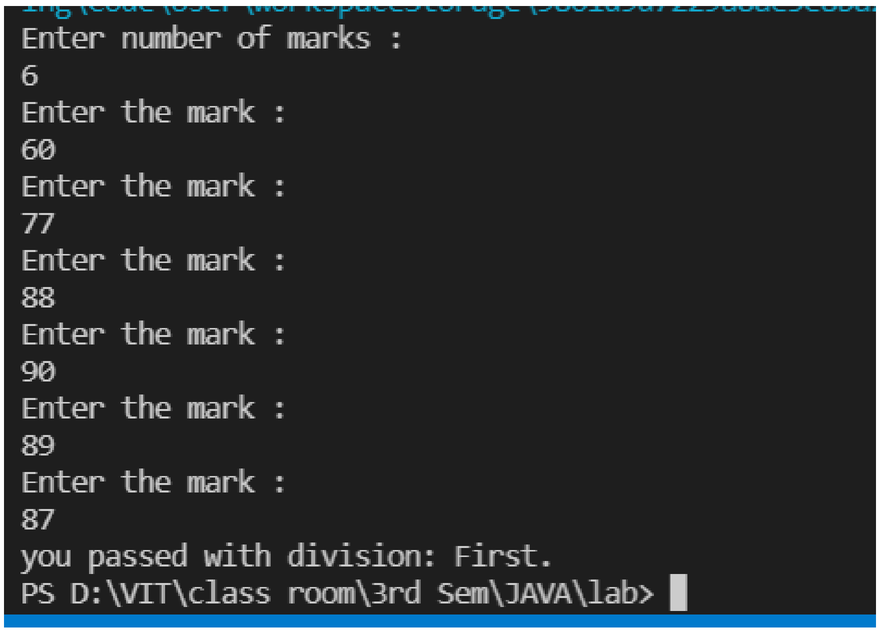
Code:

```java
import java.util.*;

interface Exam {
    int Pass(int mark[]);
}

interface Classify {
    String Average(int total, int num);
}

class Result implements Exam, Classify {
    public int Pass(int[] mark) {
        int total = 0;
        for (int i = 0; i < mark.length; i++) {
            total = total + mark[i];
        }
        return total;
    }
```



```java
    public String Average(int tot, int num) {
        int tot1 = tot / num;
        if (tot1 >= 60)
            return "First";
        else if (tot1 >= 50)
            return "Second";
        else
            return "No-Division";
    }
}

public class MyResult {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int pass;
        System.out.println("Enter number of marks : ");
        int n = sc.nextInt();
        int mark[] = new int[n];
        String division;
        Result res = new Result();
        try {
            for (int i = 0; i < n; i++) {
                System.out.println("Enter the mark : ");
                mark[i] = sc.nextInt();
            }
            pass = res.Pass(mark);
            division = res.Average(pass, n);
            System.out.println("you passed with division: " + division + ".");
        } catch (Exception e) {
            System.out.println("Error : " + e);
        }
    }
}
```

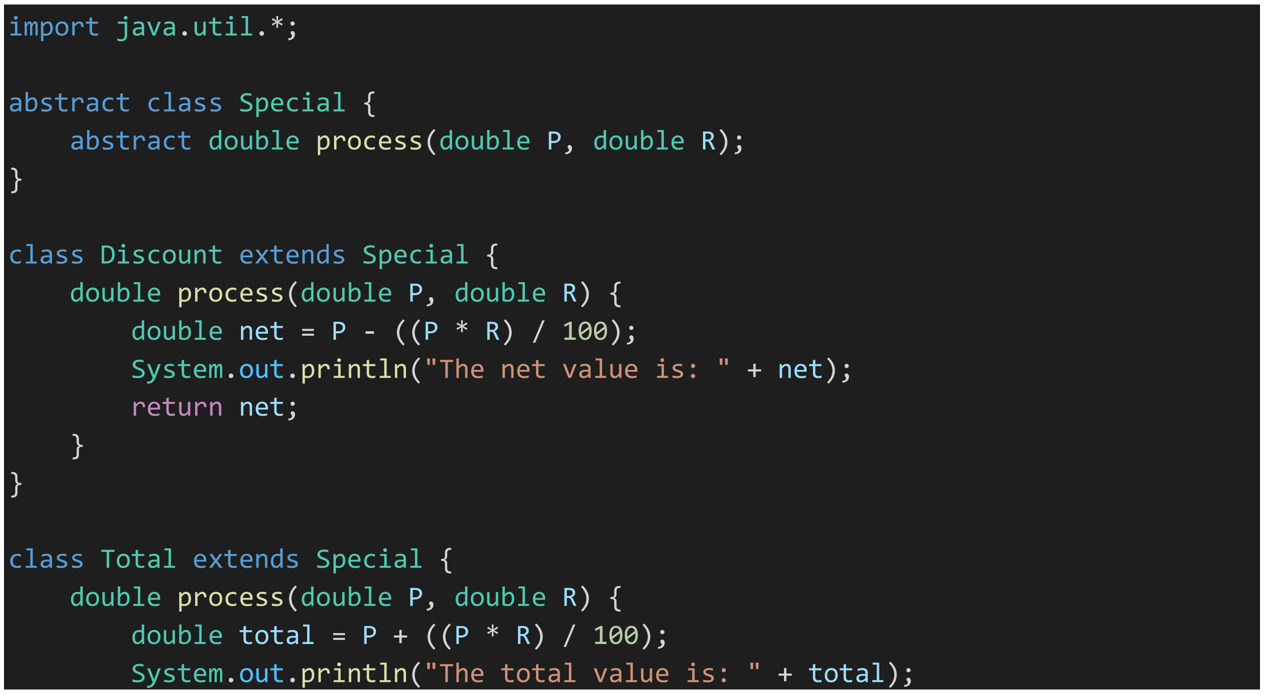
Output:



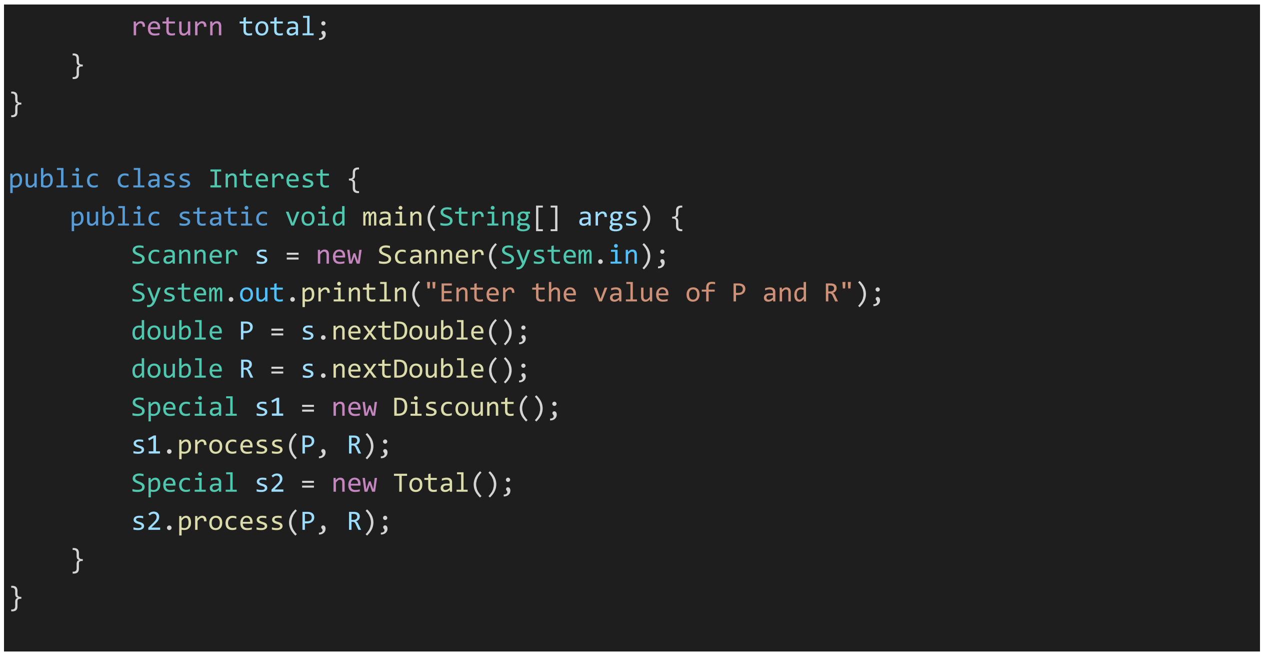
```
Enter number of marks :
6
Enter the mark :
60
Enter the mark :
77
Enter the mark :
88
Enter the mark :
90
Enter the mark :
89
Enter the mark :
87
you passed with division: First.
PS D:\VIT\class room\3rd Sem\JAVA\lab>
```

Question2:

Code:

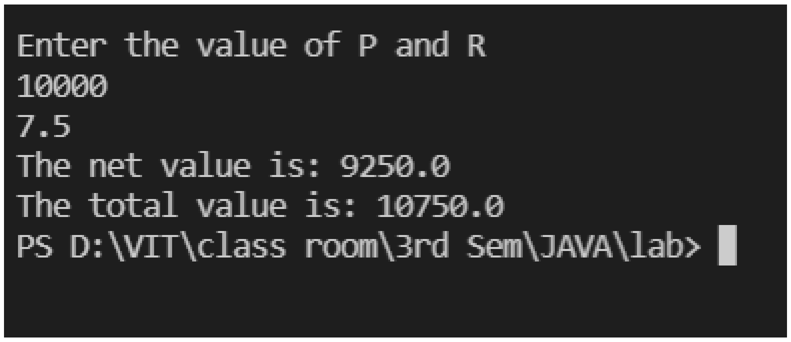


```java
import java.util.*;

abstract class Special {
    abstract double process(double P, double R);
}

class Discount extends Special {
    double process(double P, double R) {
        double net = P - ((P * R) / 100);
        System.out.println("The net value is: " + net);
        return net;
    }
}

class Total extends Special {
    double process(double P, double R) {
        double total = P + ((P * R) / 100);
        System.out.println("The total value is: " + total);
```



```
        return total;
    }
}

public class Interest {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter the value of P and R");
        double P = s.nextDouble();
        double R = s.nextDouble();
        Special s1 = new Discount();
        s1.process(P, R);
        Special s2 = new Total();
        s2.process(P, R);
    }
}
```

Output:



```
Enter the value of P and R
10000
7.5
The net value is: 9250.0
The total value is: 10750.0
PS D:\VIT\class room\3rd Sem\JAVA\lab>
```