

digital assignment3

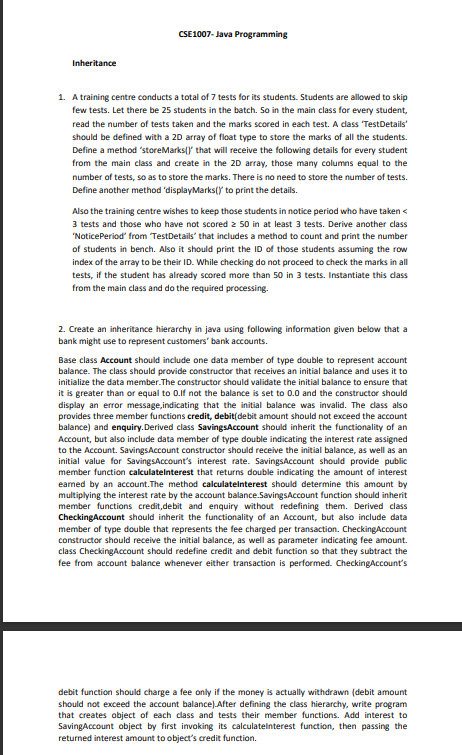
JAVA CSE1007



September 27, 2021

Anish shrestha

20BCE2893



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++;

                    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();

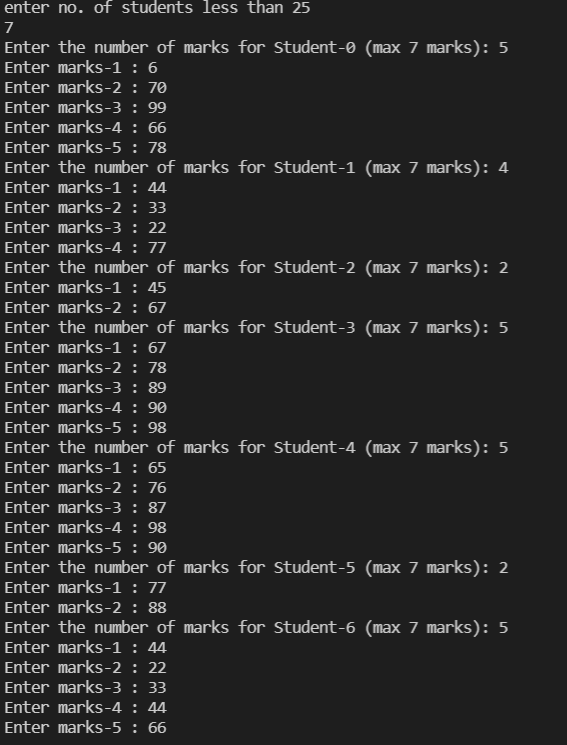
        System.out.println();

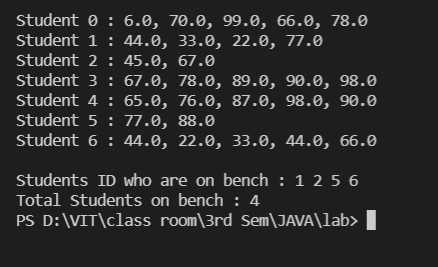
        notice.printBench();

    }

}

# Output:





Question2:

Code:

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());

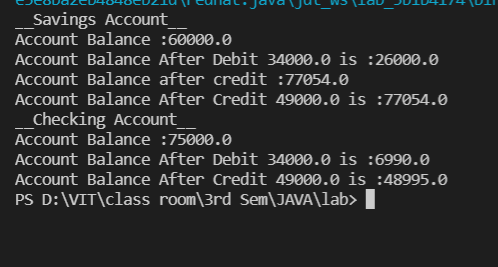
        ca1.credit(camt);

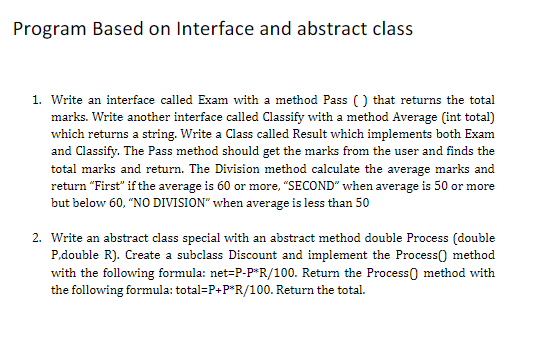
        System.out.println("Account Balance After Credit " + camt + " is :" + ca1.enquiry());

    }

}

Output:





Question 1:

Code:

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;

    }

    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) {

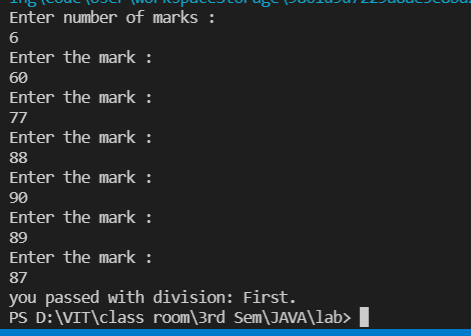
            System.out.println("Error : " + e);

        }

    }

}

Output:



Question2:

Code:

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:

