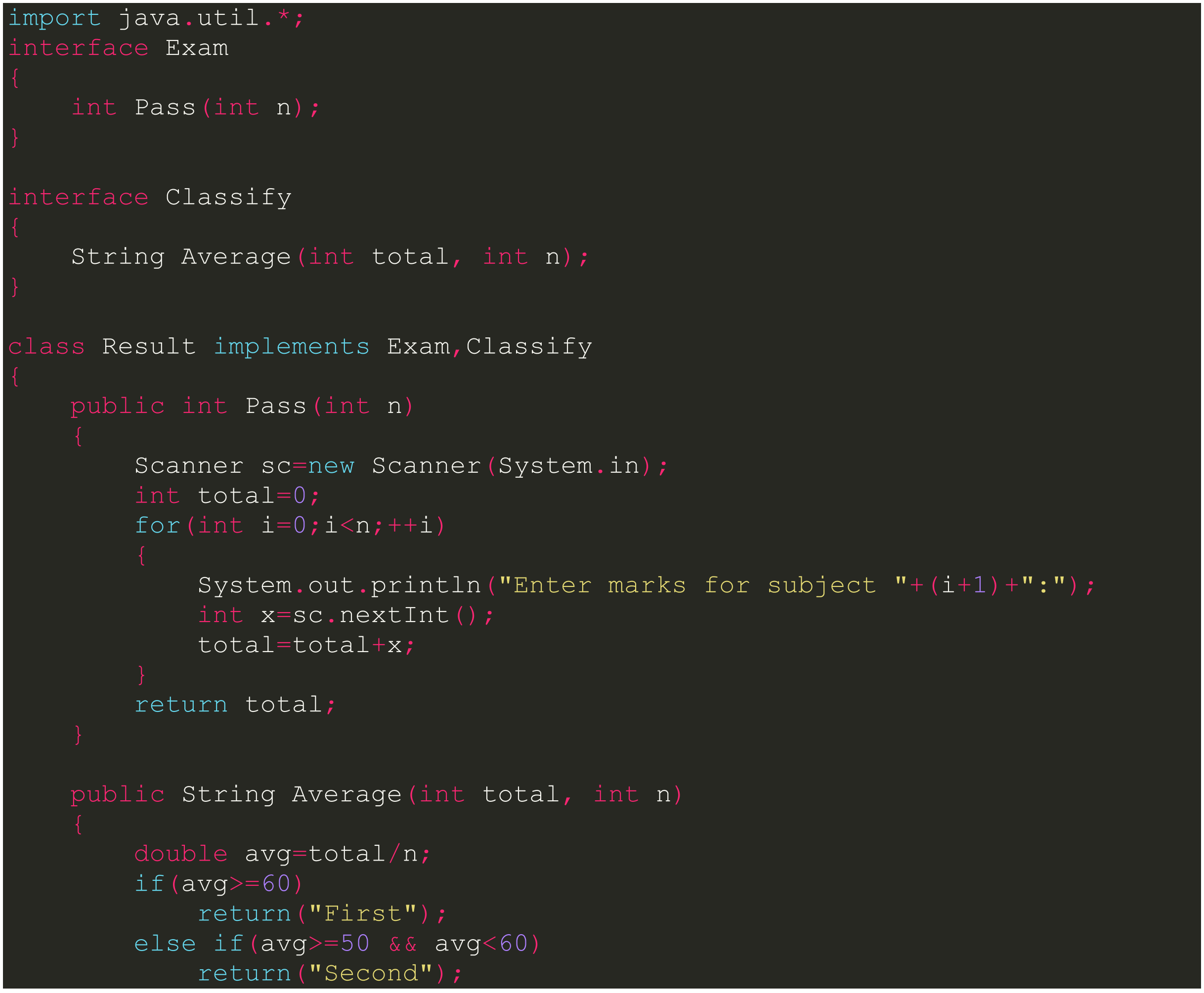
Assessment-3

Name: Sanjoi Sethi
Reg. No.: 18BCE2261
Course Name: Java Programming
Course Code: CSE1007
Slot: L9+L10

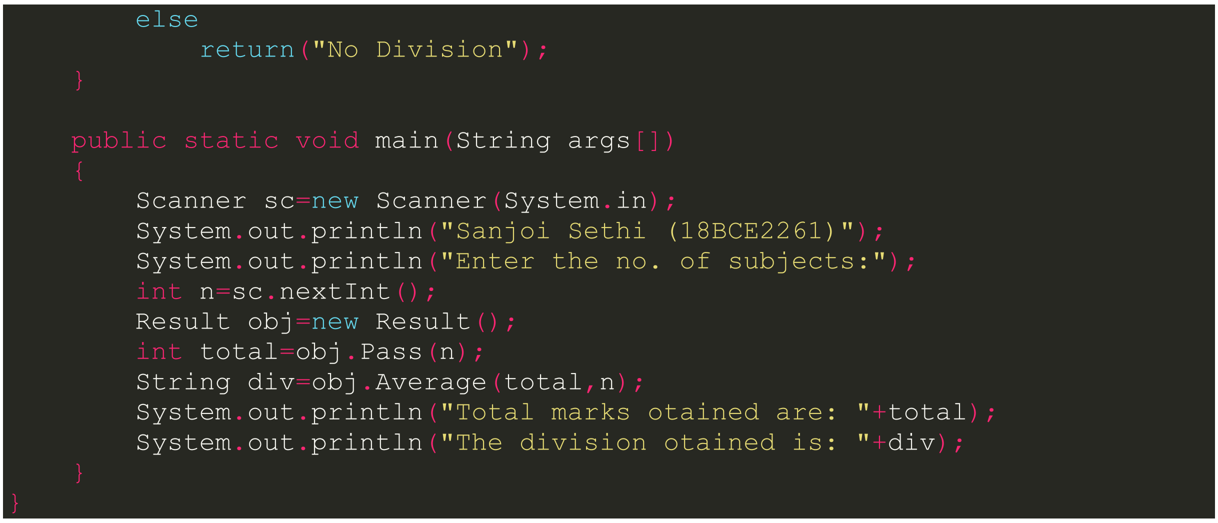
21-9-2021:
Program Based on Interface and abstract class:

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.

1) **Program:**

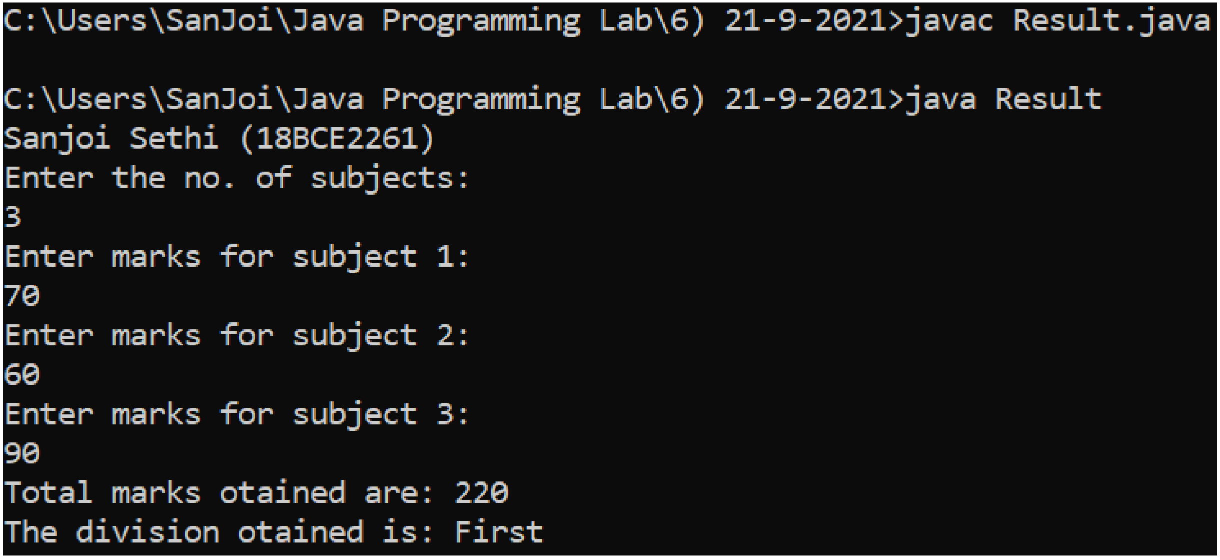


```java
import java.util.*;
interface Exam
{
    int Pass(int n);
}

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

class Result implements Exam,Classify
{
    public int Pass(int n)
    {
        Scanner sc=new Scanner(System.in);
        int total=0;
        for(int i=0;i<n;++i)
        {
            System.out.println("Enter marks for subject "+(i+1)+":");
            int x=sc.nextInt();
            total=total+x;
        }
        return total;
    }

    public String Average(int total, int n)
    {
        double avg=total/n;
        if(avg>=60)
            return("First");
        else if(avg>=50 && avg<60)
            return("Second");
```



```
        else
            return("No Division");
    }

    public static void main(String args[])
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Sanjoi Sethi (18BCE2261)");
        System.out.println("Enter the no. of subjects:");
        int n=sc.nextInt();
        Result obj=new Result();
        int total=obj.Pass(n);
        String div=obj.Average(total,n);
        System.out.println("Total marks otained are: "+total);
        System.out.println("The division otained is: "+div);
    }
}
```

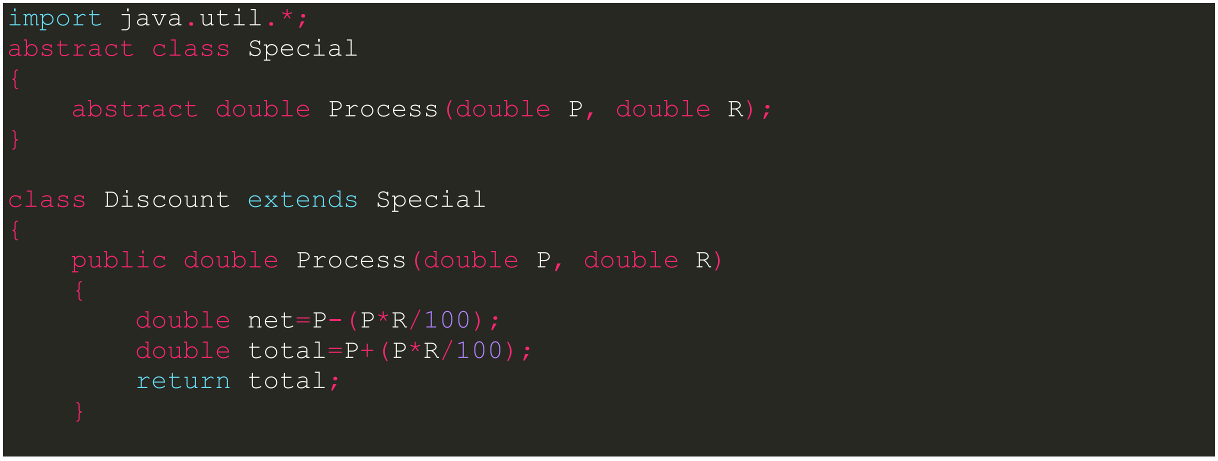
Output:



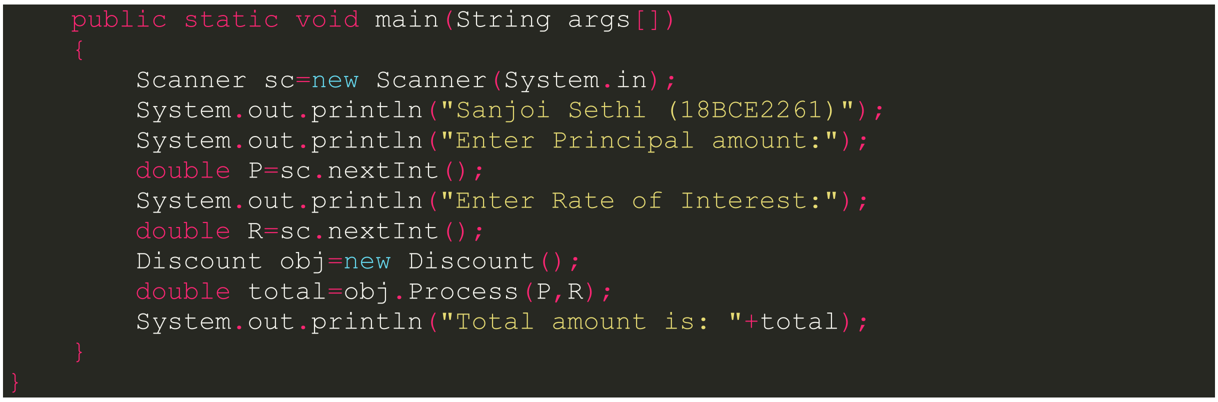
```
C:\Users\SanJoi\Java Programming Lab\6) 21-9-2021>javac Result.java

C:\Users\SanJoi\Java Programming Lab\6) 21-9-2021>java Result
Sanjoi Sethi (18BCE2261)
Enter the no. of subjects:
3
Enter marks for subject 1:
70
Enter marks for subject 2:
60
Enter marks for subject 3:
90
Total marks otained are: 220
The division otained is: First
```

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.

2) **Program:**

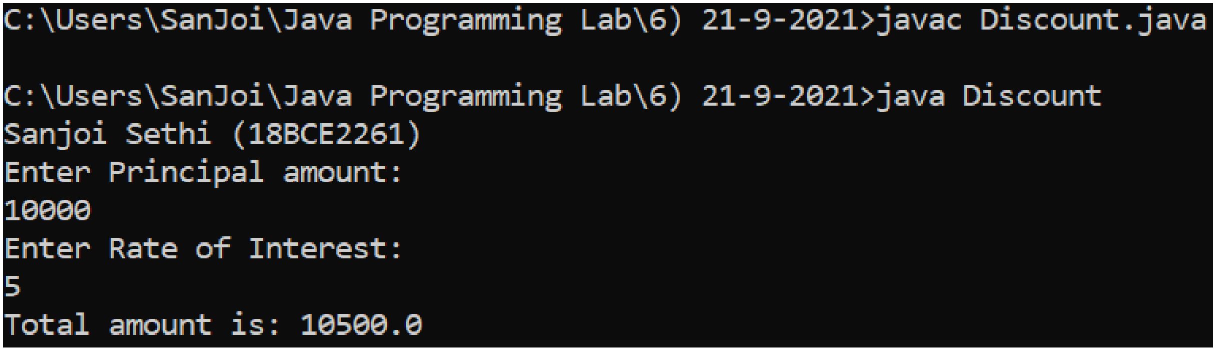


```
import java.util.*;
abstract class Special
{
    abstract double Process(double P, double R);
}

class Discount extends Special
{
    public double Process(double P, double R)
    {
        double net=P-(P*R/100);
        double total=P+(P*R/100);
        return total;
    }
```



```
    public static void main(String args[])
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Sanjoi Sethi (18BCE2261)");
        System.out.println("Enter Principal amount:");
        double P=sc.nextInt();
        System.out.println("Enter Rate of Interest:");
        double R=sc.nextInt();
        Discount obj=new Discount();
        double total=obj.Process(P,R);
        System.out.println("Total amount is: "+total);
    }
}
```

Output:

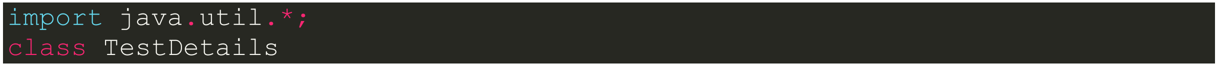


```
C:\Users\SanJoi\Java Programming Lab\6) 21-9-2021>javac Discount.java

C:\Users\SanJoi\Java Programming Lab\6) 21-9-2021>java Discount
Sanjoi Sethi (18BCE2261)
Enter Principal amount:
10000
Enter Rate of Interest:
5
Total amount is: 10500.0
```

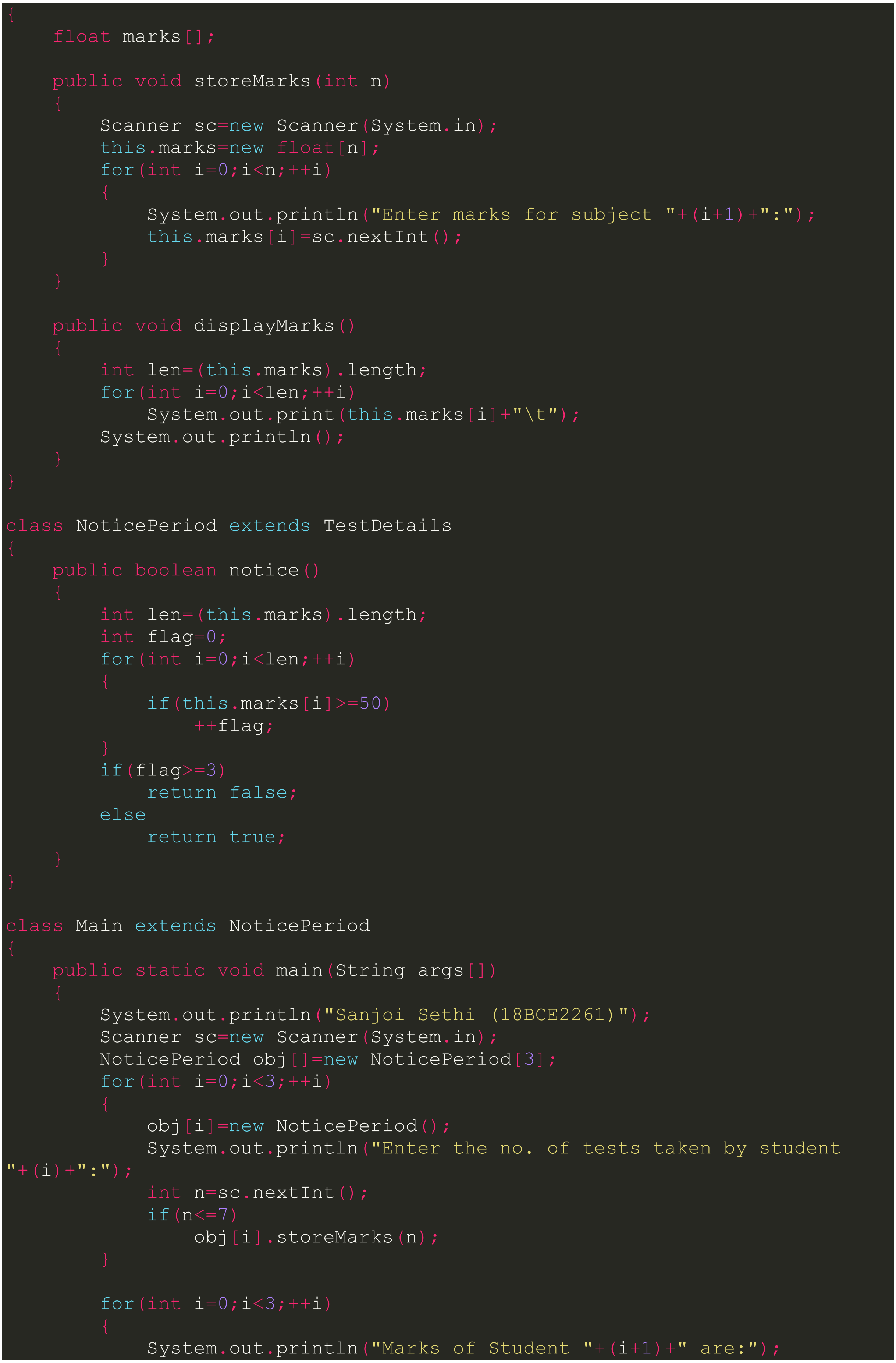
Inheritance Programs:

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.Deriveanother 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.

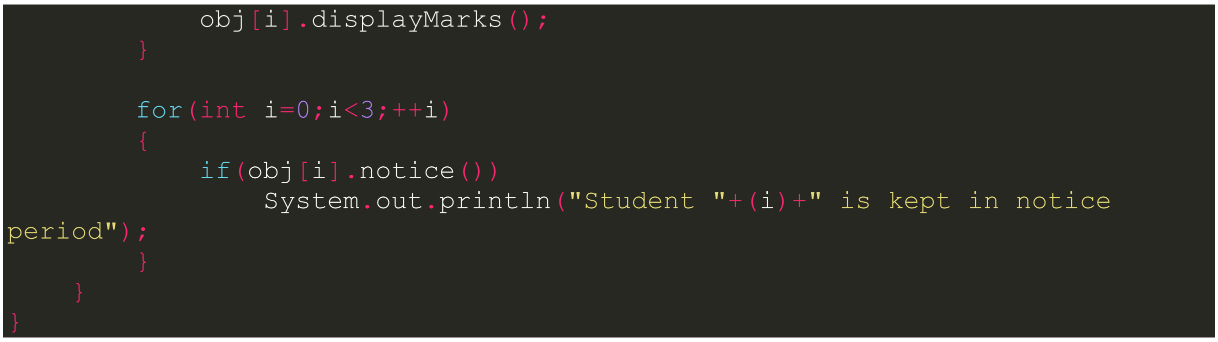
1) **Program:**



```
import java.util.*;
class TestDetails
```

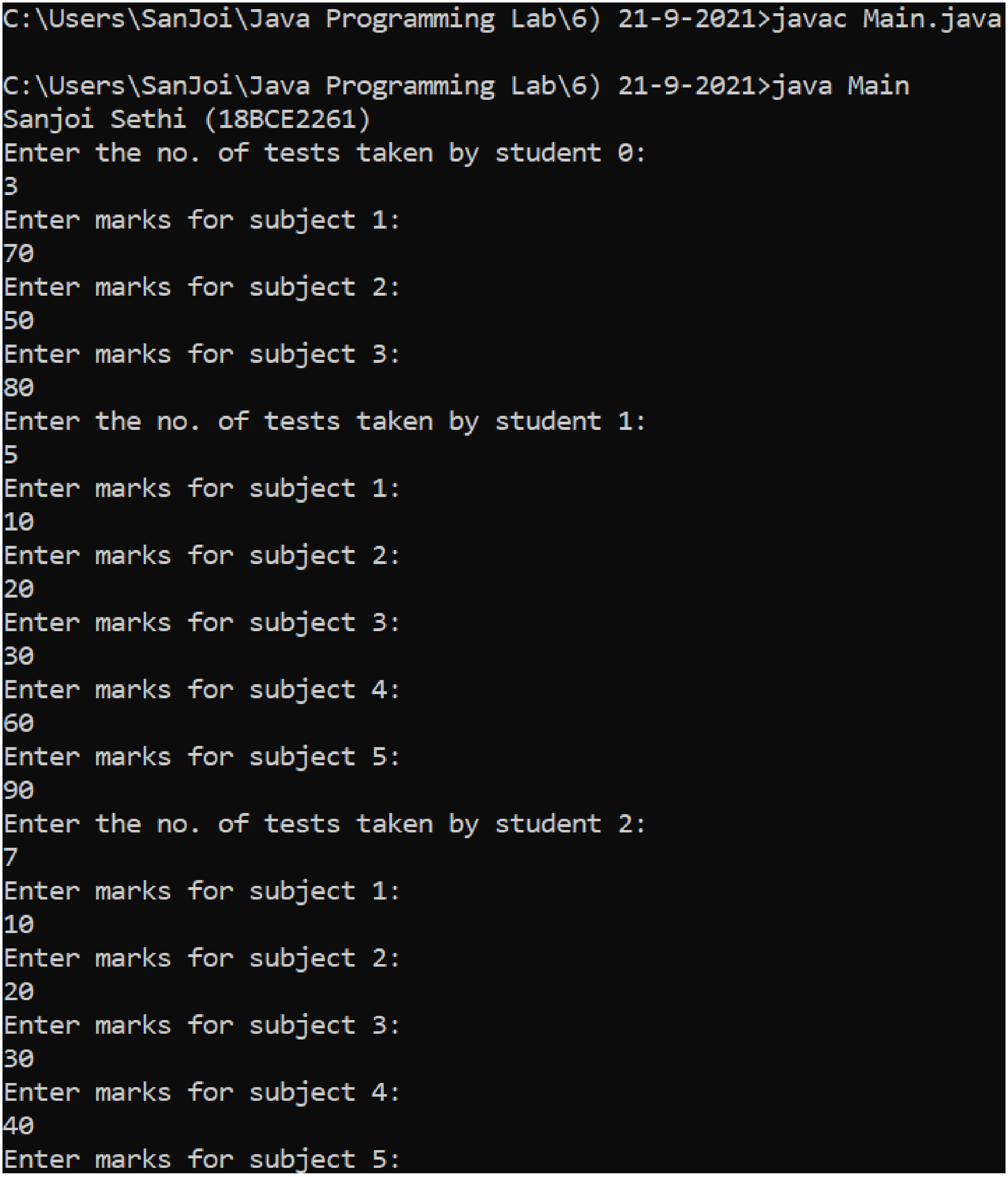


```
{
    float marks[];

    public void storeMarks(int n)
    {
        Scanner sc=new Scanner(System.in);
        this.marks=new float[n];
        for(int i=0;i<n;++i)
        {
            System.out.println("Enter marks for subject "+(i+1)+":");
            this.marks[i]=sc.nextInt();
        }
    }

    public void displayMarks()
    {
        int len=(this.marks).length;
        for(int i=0;i<len;++i)
            System.out.print(this.marks[i]+"\t");
        System.out.println();
    }
}

class NoticePeriod extends TestDetails
{
    public boolean notice()
    {
        int len=(this.marks).length;
        int flag=0;
        for(int i=0;i<len;++i)
        {
            if(this.marks[i]>=50)
                ++flag;
        }
        if(flag>=3)
            return false;
        else
            return true;
    }
}

class Main extends NoticePeriod
{
    public static void main(String args[])
    {
        System.out.println("Sanjoi Sethi (18BCE2261)");
        Scanner sc=new Scanner(System.in);
        NoticePeriod obj[]=new NoticePeriod[3];
        for(int i=0;i<3;++i)
        {
            obj[i]=new NoticePeriod();
            System.out.println("Enter the no. of tests taken by student 
"+(i)+":");
            int n=sc.nextInt();
            if(n<=7)
                obj[i].storeMarks(n);
        }

        for(int i=0;i<3;++i)
        {
            System.out.println("Marks of Student "+(i+1)+" are:");
```

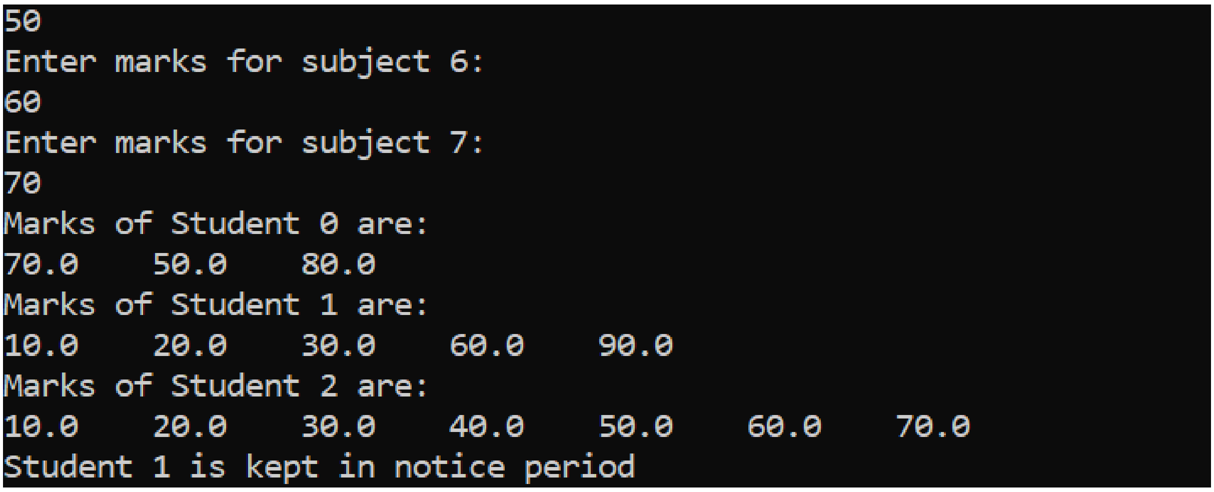


```
            obj[i].displayMarks();
        }

        for(int i=0;i<3;++i)
        {
            if(obj[i].notice())
                System.out.println("Student "+(i)+" is kept in notice
period");
        }
    }
}
```

Output:



```
C:\Users\SanJoi\Java Programming Lab\6) 21-9-2021>javac Main.java

C:\Users\SanJoi\Java Programming Lab\6) 21-9-2021>java Main
Sanjoi Sethi (18BCE2261)
Enter the no. of tests taken by student 0:
3
Enter marks for subject 1:
70
Enter marks for subject 2:
50
Enter marks for subject 3:
80
Enter the no. of tests taken by student 1:
5
Enter marks for subject 1:
10
Enter marks for subject 2:
20
Enter marks for subject 3:
30
Enter marks for subject 4:
60
Enter marks for subject 5:
90
Enter the no. of tests taken by student 2:
7
Enter marks for subject 1:
10
Enter marks for subject 2:
20
Enter marks for subject 3:
30
Enter marks for subject 4:
40
Enter marks for subject 5:
```



```
50
Enter marks for subject 6:
60
Enter marks for subject 7:
70
Marks of Student 0 are:
70.0    50.0    80.0
Marks of Student 1 are:
10.0    20.0    30.0    60.0    90.0
Marks of Student 2 are:
10.0    20.0    30.0    40.0    50.0    60.0    70.0
Student 1 is kept in notice period
```

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

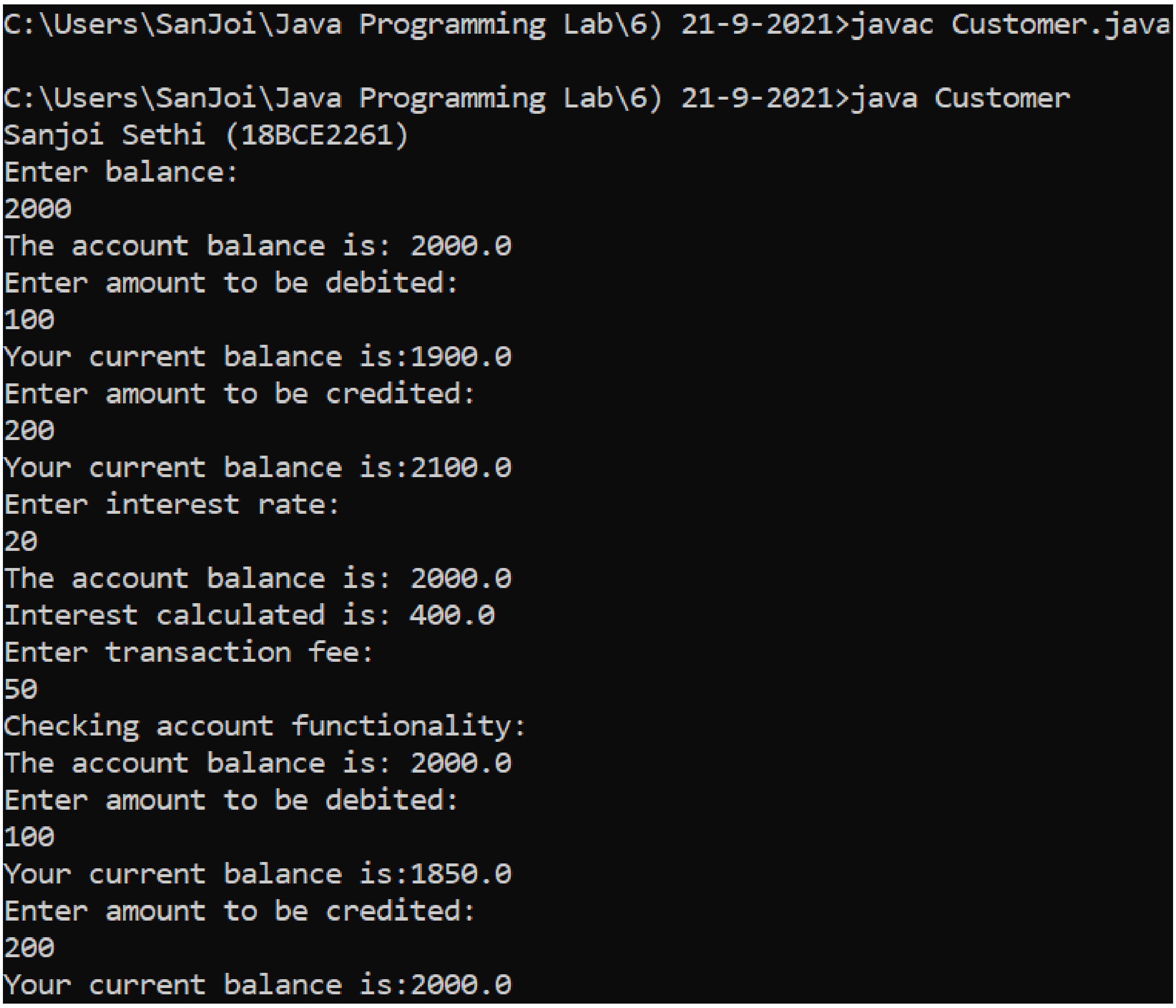
object by first invoking its calculateInterest function, then passing the returned interest amount to object’s credit function.

2) **Program:**

```
import java.util.*;
class Account
{
    double accountBalance;
    public Account(double accountBalance)
    {
        if(accountBalance>0)
        {
            this.accountBalance=accountBalance;
            System.out.println("The account balance is: "+accountBalance);
        }
        else
        {
            accountBalance=0.0;
            System.out.println("Error! Initial Balance is Invalid!");
        }
    }

    public void credit(double creditAmount)
    {
        this.accountBalance=this.accountBalance+creditAmount;
    }

    public void debit(double debitAmount)
    {
        if(debitAmount>accountBalance)
            System.out.println("The balance in your account is 
insufficient!");
        else
            this.accountBalance=this.accountBalance-debitAmount;
    }

    public void enquiry()
    {
        System.out.println("Your current balance is:"+this.accountBalance);
    }
}

class SavingsAccount extends Account
{
    double interestRate;
    public SavingsAccount(double accountBalance, double interestRate)
    {
        super(accountBalance);
        this.interestRate=interestRate*0.01;
    }

    public double calculateInterest()
    {
        double interest=this.accountBalance*this.interestRate;
        return interest;
    }
}

class CheckingAccount extends Account
{
    double fee;
```

```java
    public CheckingAccount(double accountBalance, double fee)
    {
        super(accountBalance);
        this.fee=fee;
    }

    public void credit(double creditAmount)
    {
        this.accountBalance=this.accountBalance+creditAmount-fee;
    }

    public void debit(double debitAmount)
    {
        if(debitAmount>(accountBalance+fee))
            System.out.println("The balance in your account is 
insufficient!");
        else
            this.accountBalance=this.accountBalance-debitAmount-fee;
    }
}

public class Customer
{
    public static void main(String args[])
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Sanjoi Sethi (18BCE2261)");
        System.out.println("Enter balance:");
        double accountBalance=sc.nextDouble();
        Account obj=new Account(accountBalance);
        System.out.println("Enter amount to be debited:");
        double debitAmount=sc.nextDouble();
        obj.debit(debitAmount);
        obj.enquiry();
        System.out.println("Enter amount to be credited:");
        double creditAmount=sc.nextDouble();
        obj.credit(creditAmount);
        obj.enquiry();
        System.out.println("Enter interest rate:");
        double interestRate=sc.nextDouble();
        SavingsAccount obj1 = new 
SavingsAccount(accountBalance,interestRate);
        double x=obj1.calculateInterest();
        System.out.println("Interest calculated is: "+x);
        System.out.println("Enter transaction fee:");
        double fee=sc.nextDouble();
        System.out.println("Checking account functionality:");
        CheckingAccount obj2 = new CheckingAccount(accountBalance,fee);
        System.out.println("Enter amount to be debited:");
        debitAmount=sc.nextDouble();
        obj2.debit(debitAmount);
        obj2.enquiry();
        System.out.println("Enter amount to be credited:");
        creditAmount=sc.nextDouble();
        obj2.credit(creditAmount);
        obj2.enquiry();
    }
}
```

Output:



```
C:\Users\SanJoi\Java Programming Lab\6) 21-9-2021>javac Customer.java

C:\Users\SanJoi\Java Programming Lab\6) 21-9-2021>java Customer
Sanjoi Sethi (18BCE2261)
Enter balance:
2000
The account balance is: 2000.0
Enter amount to be debited:
100
Your current balance is:1900.0
Enter amount to be credited:
200
Your current balance is:2100.0
Enter interest rate:
20
The account balance is: 2000.0
Interest calculated is: 400.0
Enter transaction fee:
50
Checking account functionality:
The account balance is: 2000.0
Enter amount to be debited:
100
Your current balance is:1850.0
Enter amount to be credited:
200
Your current balance is:2000.0
```