

# Rajalakshmi Engineering College

Name: Joe Benedict A  
Email: 241901042@rajalakshmi.edu.in  
Roll no:  
Phone: 6381868628  
Branch: REC  
Department: CSE (CS) - Section 2  
Batch: 2028  
Degree: B.E - CSE (CS)

Scan to verify results



## 2024\_28\_III\_OOPS Using Java Lab

### 2028\_REC\_OOPS using Java\_Week 5\_Q5

Attempt : 1  
Total Mark : 10  
Marks Obtained : 10

#### Section 1 : Coding

##### 1. Problem Statement

Ram is working as a developer for BrightEdu Coaching Center, which wants to build a student fee management system.

Each student's enrollment has:

An Enrollment ID (integer) A Student Name (string) The Number of Subjects (integer)

The fee calculation rules are:

Registration Fee = 1000 units (flat for every student). Per Subject Fee = 800 units. If the student enrolls in more than 5 subjects, a 20% scholarship (discount) is applied on the total fee.

Ram has been asked to implement this system using:

A class with attributes for student details. A constructor to initialize student details. Setter methods to update details if needed. Getter methods to retrieve details. Objects of the class to represent student enrollments.

Finally, display each student's details and final fee.

### ***Input Format***

The first line of input contains an integer N, representing the number of students.

For each student:

- The next line contains the Enrollment ID (integer).
- The following line contains the student's name (string).
- The next line contains the Number of subjects (integer).

### ***Output Format***

For each student, print the details in the following format:

- Enrollment ID: <enrollment\_id>
- Student Name: <student\_name>
- Final Fee: <final\_fee> (rounded to one decimal place)

Refer to the sample output for formatting specifications.

### ***Sample Test Case***

Input: 1

1234

Ravi Kumar

3

Output: Enrollment ID: 1234

Student Name: Ravi Kumar

Final Fee: 3400.0

### ***Answer***

```
import java.util.Scanner;
```

```
class StudentEnrollment {  
    private int enrollmentId;
```

```
private String studentName;
private int numberOfSubjects;

// Constructor
public StudentEnrollment(int enrollmentId, String studentName, int
numberOfSubjects) {
    this.enrollmentId = enrollmentId;
    this.studentName = studentName;
    this.numberOfSubjects = numberOfSubjects;
}

// Setter methods
public void setEnrollmentId(int enrollmentId) {
    this.enrollmentId = enrollmentId;
}

public void setStudentName(String studentName) {
    this.studentName = studentName;
}

public void setNumberOfSubjects(int numberOfSubjects) {
    this.numberOfSubjects = numberOfSubjects;
}

// Getter methods
public int getEnrollmentId() {
    return enrollmentId;
}

public String getStudentName() {
    return studentName;
}

public int getNumberOfSubjects() {
    return numberOfSubjects;
}

// Method to calculate final fee
public double calculateFee() {
    double registrationFee = 1000;
    double subjectFee = numberOfSubjects * 800;
    double totalFee = registrationFee + subjectFee;
```

```

        if (numberOfSubjects > 5) {
            totalFee *= 0.80; // Apply 20% scholarship
        }

        return totalFee;
    }

    // Method to display student details
    public void displayDetails() {
        System.out.println("Enrollment ID: " + enrollmentId);
        System.out.println("Student Name: " + studentName);
        System.out.printf("Final Fee: %.1f\n", calculateFee());
    }
}

class FeeManagementSystem {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int N = Integer.parseInt(sc.nextLine());

        StudentEnrollment[] students = new StudentEnrollment[N];

        for (int i = 0; i < N; i++) {
            int id = Integer.parseInt(sc.nextLine());
            String name = sc.nextLine();
            int subjects = Integer.parseInt(sc.nextLine());

            students[i] = new StudentEnrollment(id, name, subjects);
        }

        for (StudentEnrollment student : students) {
            student.displayDetails();
        }

        sc.close();
    }
}

```

**Status :** Correct

**Marks :** 10/10