

# Rajalakshmi Engineering College

Name: Sunetra B  
Email: 241801282@rajalakshmi.edu.in  
Roll no: 241801282  
Phone: 9150958405  
Branch: REC  
Department: AI & DS - Section 5  
Batch: 2028  
Degree: B.E - AI & DS

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 {  
    // attributes
```

```
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;
}

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

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

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

// getters
public int getEnrollmentId() {
    return enrollmentId;
}

public String getStudentName() {
    return studentName;
}

public int getNumberOfSubjects() {
    return numberOfSubjects;
}

// fee calculation
public double calculateFee() {
    double registrationFee = 1000;
    double perSubjectFee = 800;
```

```

double totalFee = registrationFee + (numberOfSubjects * perSubjectFee);

    // apply 20% scholarship if subjects > 5
    if (numberOfSubjects > 5) {
        totalFee = totalFee - (totalFee * 0.20);
    }

    return totalFee;
}

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

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        int N = sc.nextInt(); // number of students
        sc.nextLine(); // consume newline

        for (int i = 0; i < N; i++) {
            int id = sc.nextInt();
            sc.nextLine(); // consume newline
            String name = sc.nextLine();
            int subjects = sc.nextInt();

            // create student object
            StudentEnrollment student = new StudentEnrollment(id, name, subjects);

            // display student details
            student.displayDetails();
        }

        sc.close();
    }
}

```

Status : Correct

Marks : 10/10

# Rajalakshmi Engineering College

Name: Sunetra B  
Email: 241801282@rajalakshmi.edu.in  
Roll no: 241801282  
Phone: 9150958405  
Branch: REC  
Department: AI & DS - Section 5  
Batch: 2028  
Degree: B.E - AI & DS

Scan to verify results



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

### 2028\_REC\_OOPS using Java\_Week 5\_Q4

Attempt : 1  
Total Mark : 10  
Marks Obtained : 10

#### Section 1 : Coding

##### 1. Problem Statement

You are working as a developer for CityCab, a taxi service company that wants to build a ride fare management system.

Each customer booking has:

A Booking ID (integer) A Customer Name (string) A Distance Travelled in km (double)

The fare calculation rules are:

Base Fare = 50 units (flat charge for every ride). Per km charge = 10 units/km. If the distance is greater than 20 km, a 10% discount is applied on the total fare.

You are required to implement this system using:

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

Finally, display each booking's details and final fare.

### ***Input Format***

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

For each booking:

- The next line contains the booking ID (integer).
- The following line contains the customer's name (string).
- The next line contains the distance travelled (double).

### ***Output Format***

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

1. Booking ID: <booking\_id>
2. Customer Name: <customer\_name>
3. Final Fare: <final\_fare> (rounded to one decimal place)

Refer to the sample output for formatting specifications.

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

Input: 1  
1234  
Rahul Sharma  
15

Output: Booking ID: 1234  
Customer Name: Rahul Sharma  
Final Fare: 200.0

### ***Answer***

```
// You are using Java
import java.util.Scanner;
```

```
class Booking {
    private int bookingId;
    private String customerName;
    private double distance;
    private double fare;

    public Booking(int bookingId, String customerName, double distance) {
        this.bookingId = bookingId;
        this.customerName = customerName;
        this.distance = distance;
        calculateFare();
    }

    public void setBookingId(int bookingId) {
        this.bookingId = bookingId;
    }

    public void setCustomerName(String customerName) {
        this.customerName = customerName;
    }

    public void setDistance(double distance) {
        this.distance = distance;
        calculateFare();
    }

    public int getBookingId() {
        return bookingId;
    }

    public String getCustomerName() {
        return customerName;
    }

    public double getDistance() {
        return distance;
    }

    public double getFare() {
        return fare;
    }
}
```



```

private void calculateFare() {
    fare = 100 + (distance * 10-50);
    if (distance > 20) {
        fare = fare - (fare * 0.1);
    }
}
}

class CityCabApp {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = Integer.parseInt(sc.nextLine());
        for (int i = 0; i < n; i++) {
            int id = Integer.parseInt(sc.nextLine());
            String name = sc.nextLine();
            double distance = Double.parseDouble(sc.nextLine());
            Booking booking = new Booking(id, name, distance);
            System.out.println("Booking ID: " + booking.getBookingId());
            System.out.println("Customer Name: " + booking.getCustomerName());
            System.out.printf("Final Fare: %.1f\n", booking.getFare());
        }
        sc.close();
    }
}

```

**Status :** Correct

**Marks :** 10/10

# Rajalakshmi Engineering College

Name: Sunetra B  
Email: 241801282@rajalakshmi.edu.in  
Roll no: 241801282  
Phone: 9150958405  
Branch: REC  
Department: AI & DS - Section 5  
Batch: 2028  
Degree: B.E - AI & DS

Scan to verify results



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

### 2028\_REC\_OOPS using Java\_Week 5\_Q3

Attempt : 1  
Total Mark : 10  
Marks Obtained : 10

#### Section 1 : Coding

##### 1. Problem Statement

Neha is working as a developer for CityElectricity Board, which wants to build a household electricity billing system.

Each customer's electricity account has:

A Customer ID (integer) A Customer Name (string) Units Consumed (double)

The electricity bill is calculated based on these rules:

For the first 100 units 5 units charge per unit For the next 100 units (101–200) 7 units charge per unit For units above 200 10 units charge per unit If the total bill exceeds 2000 units, a 5% discount is applied on the final bill.

Neha has been asked to implement this system using:

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

Finally, display each customer's details and final bill amount.

### ***Input Format***

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

For each customer:

- The next line contains the Customer ID (integer).
- The following line contains the Customer Name (string).
- The next line contains the Units Consumed (double).

### ***Output Format***

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

Customer ID: <customer\_id>

Customer Name: <customer\_name>

Final Bill: <final\_bill> (rounded to one decimal place)

Refer to the sample output for formatting specifications.

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

Input: 1

1001

Ravi Kumar

80

Output: Customer ID: 1001

Customer Name: Ravi Kumar

Final Bill: 400.0

### ***Answer***

// You are using Java

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

```
class ElectricityAccount {  
    private int customerId;  
    private String customerName;  
    private double unitsConsumed;
```

```
    public ElectricityAccount(int customerId, String customerName, double  
unitsConsumed) {  
        this.customerId = customerId;  
        this.customerName = customerName;  
        this.unitsConsumed = unitsConsumed;  
    }
```

```
    public void setCustomerId(int customerId) {  
        this.customerId = customerId;  
    }
```

```
    public void setCustomerName(String customerName) {  
        this.customerName = customerName;  
    }
```

```
    public void setUnitsConsumed(double unitsConsumed) {  
        this.unitsConsumed = unitsConsumed;  
    }
```

```
    public int getCustomerId() {  
        return customerId;  
    }
```

```
    public String getCustomerName() {  
        return customerName;  
    }
```

```
    public double getUnitsConsumed() {  
        return unitsConsumed;  
    }
```

```
    public double calculateBill() {  
        double bill = 0;  
        double units = unitsConsumed;
```

```

        if (units <= 100) {
            bill = units * 5;
        } else if (units <= 200) {
            bill = 100 * 5 + (units - 100) * 7;
        } else {
            bill = 100 * 5 + 100 * 7 + (units - 200) * 10;
        }

        if (bill > 2000) {
            bill = bill - (bill * 0.05);
        }

        return bill;
    }

    public void displayDetails() {
        System.out.println("Customer ID: " + customerId);
        System.out.println("Customer Name: " + customerName);
        System.out.printf("Final Bill: %.1f\n", calculateBill());
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        int N = sc.nextInt();
        sc.nextLine();

        for (int i = 0; i < N; i++) {
            int id = sc.nextInt();
            sc.nextLine();
            String name = sc.nextLine();
            double units = sc.nextDouble();

            ElectricityAccount customer = new ElectricityAccount(id, name, units);

            customer.displayDetails();
        }
        sc.close();
    }
}

```

}

**Status :** Correct

**Marks : 10/10**

# Rajalakshmi Engineering College

Name: Sunetra B  
Email: 241801282@rajalakshmi.edu.in  
Roll no: 241801282  
Phone: 9150958405  
Branch: REC  
Department: AI & DS - Section 5  
Batch: 2028  
Degree: B.E - AI & DS

Scan to verify results



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

### 2028\_REC\_OOPS using Java\_Week 5\_Q2

Attempt : 1  
Total Mark : 10  
Marks Obtained : 10

#### Section 1 : Coding

##### 1. Problem Statement

You are working as a developer for CityBank, which wants to build a basic account management system.

Each customer at the bank has:

An Account Number (integer) A Customer Name (string) An Initial Balance (double)

The bank allows two types of transactions:

Deposit – increases the balance. Withdrawal – decreases the balance only if enough funds are available.

If the withdrawal amount is greater than the balance, the withdrawal should not happen, and the balance should remain the same.

You are required to implement this system using:

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

Finally, display each customer's account details after all transactions.

### ***Input Format***

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

For each customer:

- The next line contains the account number (integer).
- The following line contains the customer name (string).
- The next line contains the initial balance (double).
- The next line contains the deposit amount (double).
- The next line contains the withdrawal amount (double).

### ***Output Format***

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

1. Account Number: <account\_number>
2. Customer Name: <customer\_name>
3. Final Balance: <final\_balance> (rounded to one decimal place)

Refer to the sample output for formatting specifications.

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

Input: 1

1234

Rahul Sharma

5000

2000

3000

Output: Account Number: 1234

Customer Name: Rahul Sharma



Final Balance: 4000.0

**Answer**

```
import java.util.Scanner;

class BankAccount {
    // attributes
    private int accountNumber;
    private String customerName;
    private double balance;

    // constructor
    public BankAccount(int accountNumber, String customerName, double
balance) {
        this.accountNumber = accountNumber;
        this.customerName = customerName;
        this.balance = balance;
    }

    // setters
    public void setAccountNumber(int accountNumber) {
        this.accountNumber = accountNumber;
    }

    public void setCustomerName(String customerName) {
        this.customerName = customerName;
    }

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

    // getters
    public int getAccountNumber() {
        return accountNumber;
    }

    public String getCustomerName() {
        return customerName;
    }

    public double getBalance() {
```

```

        return balance;
    }

    // deposit method
    public void deposit(double amount) {
        if (amount >= 0) {
            balance += amount;
        }
    }

    // withdrawal method
    public void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
        }
        // if amount > balance do nothing
    }

    // display details
    public void displayDetails() {
        System.out.println("Account Number: " + accountNumber);
        System.out.println("Customer Name: " + customerName);
        System.out.printf("Final Balance: %.1f\n", balance);
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        int N = sc.nextInt(); // number of customers
        sc.nextLine(); // consume newline

        for (int i = 0; i < N; i++) {
            int accNo = sc.nextInt();
            sc.nextLine(); // consume newline
            String name = sc.nextLine();
            double initialBalance = sc.nextDouble();
            double depositAmount = sc.nextDouble();
            double withdrawAmount = sc.nextDouble();

            // create object

```

```
BankAccount customer = new BankAccount(accNo, name, initialBalance);

// perform transactions
customer.deposit(depositAmount);
customer.withdraw(withdrawAmount);

// display final details
customer.displayDetails();
}

sc.close();
}
```

**Status :** Correct

**Marks :** 10/10