

Rajalakshmi Engineering College

Name: Gnaana Kumaar

Email: 241001062@rajalakshmi.edu.in

Roll no: 241001062

Phone: 9790249960

Branch: REC

Department: IT - Section 4

Batch: 2028

Degree: B.E - IT

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 5_MCQ

Attempt : 1

Total Mark : 15

Marks Obtained : 14

Section 1 : MCQ

1. What will be the output of the following code?

```
class Person {  
    int age = 18;  
}  
  
public class Main {  
    public static void main(String[] args) {  
        Person p = new Person();  
        p.age += 2;  
        System.out.println("Age: " + p.age);  
    }  
}
```

Answer

Age: 20

Status : Correct

Marks : 1/1

2. What will be the output of the following code?

```
class A {  
    int x = 50;  
}  
  
public class Main {  
    public static void main(String[] args) {  
        A obj1 = new A();  
        A obj2 = obj1;  
        obj2.x = 100;  
        System.out.println(obj1.x);  
    }  
}
```

Answer

100

Status : Correct

Marks : 1/1

3. What will be the output of the following code?

```
class A {  
    int val = 20;  
}  
  
public class Main {  
    public static void main(String[] args) {  
        A obj1 = new A();  
        A obj2 = obj1;  
        obj2.val += 5;  
        System.out.println(obj1.val);  
    }  
}
```

Answer

25

Status : Correct

Marks : 1/1

4. What will be the output of the following code?

```
class Box {  
    int length = 5;  
    int width = 4;  
  
    int area() {  
        return length * width;  
    }  
  
    public static void main(String[] args) {  
        Box b = new Box();  
        System.out.println("Area = " + b.area());  
    }  
}
```

Answer

Area = 20

Status : Correct

Marks : 1/1

5. What will be the output of the following code?

```
class Demo {  
    void printMessage() {  
        System.out.println("Hello from Demo");  
    }  
}
```

```
public class Main {  
    public static void main(String[] args) {  
        Demo d = new Demo();  
        d.printMessage();  
    }  
}
```

Answer

Hello from Demo

Status : Correct

Marks : 1/1

6. What will be the output of the following code?

```
class Alpha {  
    void greet(String name) {  
        System.out.println("Hello " + name);  
    }  
}  
  
public class Main {  
    public static void main(String[] args) {  
        Alpha obj = new Alpha();  
        obj.greet("Anu");  
    }  
}
```

Answer

Hello Anu

Status : Correct

Marks : 1/1

7. What will be the output of the following code?

```
class Ball {  
    int size = 11;  
}  
  
class Game {  
    public static void main(String[] args) {  
        Ball b1 = new Ball();  
        Ball b2 = new Ball();  
        b2.size = 10;  
        System.out.println(b1.size);  
    }  
}
```

```
}
```

Answer

11

Status : Correct

Marks : 1/1

8. What will be the output of the following code?

```
class Person {  
    String name;  
    void setName(String n) {  
        name = n;  
    }  
    void printName() {  
        System.out.println(name);  
    }  
}
```

```
class Test {  
    public static void main(String[] args) {  
        Person p = new Person();  
        p.printName();  
    }  
}
```

Answer

null

Status : Correct

Marks : 1/1

9. What will be the output of the following code?

```
class A {  
    int y = 30;  
}
```

```
public class Main {  
    public static void main(String[] args) {
```

```
A a1 = new A();
A a2 = new A();
a1.y = 50;
System.out.println(a2.y);
}
}
```

Answer

30

Status : Correct

Marks : 1/1

10. What will be the output of the following code?

```
class Box {
    int volume(int l, int b, int h) {
        return l * b * h;
    }
}

public class Main {
    public static void main(String[] args) {
        Box b = new Box();
        System.out.println(b.volume(2, 3, 4));
    }
}
```

Answer

24

Status : Correct

Marks : 1/1

11. What will be the output of the following code?

```
class Test {
    private int value;
    Test(int value) {
        this.value = value;
    }
}
```

```
    public int getValue() {  
        return value;  
    }  
}  
  
public class Main {  
    public static void main(String[] args) {  
        Test obj = new Test(10);  
        System.out.println(obj.value);  
    }  
}
```

Answer

Runtime error

Status : Wrong

Marks : 0/1

12. What will be the output of the following code?

```
class MathUtils {  
    int add(int x) {  
        return x + x;  
    }  
}  
  
public class Main {  
    public static void main(String[] args) {  
        MathUtils m = new MathUtils();  
        System.out.println(m.add(5));  
    }  
}
```

Answer

10

Status : Correct

Marks : 1/1

13. What will be the output of the following code?

```
class A {
```

```
int p = 5;
int q = 2;
}

class Main {
    public static void main(String[] args) {
        A obj = new A();
        System.out.println(obj.p + obj.q);
    }
}
```

Answer

7

Status : Correct

Marks : 1/1

14. What will be the output of the following code?

```
class Sample {
    int x = 10;

    void display() {
        System.out.println("x = " + x);
    }

    public static void main(String[] args) {
        Sample s = new Sample();
        s.display();
    }
}
```

Answer

x = 10

Status : Correct

Marks : 1/1

15. What is the output of the following code?

```
class Box {
```



```
int height;  
Box(int height) {  
    this.height = height;  
}  
void modifyHeight(Box b) {  
    b.height += 10;  
}  
}  
public class Main {  
    public static void main(String[] args) {  
        Box b1 = new Box(20);  
        b1.modifyHeight(b1);  
        System.out.println(b1.height);  
    }  
}
```

Answer

30

Status : Correct

Marks : 1/1

Rajalakshmi Engineering College

Name: Gnaana Kumaar
Email: 241001062@rajalakshmi.edu.in
Roll no: 241001062
Phone: 9790249960
Branch: REC
Department: IT - Section 4
Batch: 2028
Degree: B.E - IT

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

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

```
class Student {
```

```
private int enrollmentId;
private String studentName;
private int numberOfSubjects;

// Constructor
public Student(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;
}

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

```

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

        return totalFee;
    }
}

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

        for (int i = 0; i < N; i++) {
            int enrollmentId = Integer.parseInt(scanner.nextLine());
            String studentName = scanner.nextLine();
            int numberOfSubjects = Integer.parseInt(scanner.nextLine());

            Student student = new Student(enrollmentId, studentName,
            numberOfSubjects);
            double finalFee = student.calculateFinalFee();

            // Output in required format
            System.out.printf("Enrollment ID: %d\n", student.getEnrollmentId());
            System.out.printf("Student Name: %s\n", student.getStudentName());
            System.out.printf("Final Fee: %.1f\n", finalFee);
        }

        scanner.close();
    }
}

```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: Gnaana Kumaar

Email: 241001062@rajalakshmi.edu.in

Roll no: 241001062

Phone: 9790249960

Branch: REC

Department: IT - Section 4

Batch: 2028

Degree: B.E - IT

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 : 0

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 = 50 - distance * 10;
    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 : Wrong

Marks : 0/10

Rajalakshmi Engineering College

Name: Gnaana Kumaar

Email: 241001062@rajalakshmi.edu.in

Roll no: 241001062

Phone: 9790249960

Branch: REC

Department: IT - Section 4

Batch: 2028

Degree: B.E - IT

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 : 0

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 Customer {  
    private int CustomerId;  
    private String CustomerName;  
    private double UnitsConsumed;
```

```
    public Customer(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 class ElectricityBillingSystem {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        int N = Integer.parseInt(scanner.nextLine());
        for (int i = 0; i < N; i++) {
            int id = Integer.parseInt(scanner.nextLine());
            String name = scanner.nextLine();
            double units = Double.parseDouble(scanner.nextLine());

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

            double finalBill = customer.calculateBill();

            /
            System.out.printf("Customer ID: %d\n", customer.getCustomerId());
            System.out.printf("Customer Name: %s\n",
customer.getCustomerName());
            System.out.printf("Final Bill: %.1f\n", finalBill);
        }
    }
}

```

241001062

```
scanner.close();
```

```
}  
}
```

241001062

241001062

241001062

Status : Wrong

Marks : 0/10

241001062

241001062

241001062

241001062

241001062

241001062

241001062

241001062

241001062

241001062

241001062

241001062

Rajalakshmi Engineering College

Name: Gnaana Kumaar

Email: 241001062@rajalakshmi.edu.in

Roll no: 241001062

Phone: 9790249960

Branch: REC

Department: IT - Section 4

Batch: 2028

Degree: B.E - IT

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 {
```

```
    private int accountNumber;  
    private String customerName;  
    private double balance;
```

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

```
    public int getAccountNumber() {  
        return accountNumber;  
    }
```

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

```
    public double getBalance() {  
        return balance;  
    }
```

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

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

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

```
public void withdraw(double amount) {  
    if (amount <= balance) {  
        balance -= amount;  
    }  
}
```

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

```
        int N = Integer.parseInt(sc.nextLine());
```

```
        BankAccount[] accounts = new BankAccount[N];
```

```
        for (int i = 0; i < N; i++) {
```

```
            int accountNumber = Integer.parseInt(sc.nextLine());
```

```
            String customerName = sc.nextLine();
```

```
            double initialBalance = Double.parseDouble(sc.nextLine());
```

```
            BankAccount account = new BankAccount(accountNumber,  
            customerName, initialBalance);
```

```
            double depositAmount = Double.parseDouble(sc.nextLine());
```

```
account.deposit(depositAmount);

double withdrawalAmount = Double.parseDouble(sc.nextLine());
account.withdraw(withdrawalAmount);

accounts[i] = account;
}

for (int i = 0; i < N; i++) {
    BankAccount acc = accounts[i];
    System.out.println("Account Number: " + acc.getAccountNumber() +
        " Customer Name: " + acc.getCustomerName() +
        " Final Balance: " + String.format("%.1f", acc.getBalance()));
}

sc.close();
}
```

Status : Correct

Marks : 10/10