National University of Modern Languages

Student Name	Muhammad Zohaib
Class	BSSE-31-5-(A)
Roll no.	SP21295
Subject	SCD
Due	@April 1, 2024

Lab Report 7:

Q.1: Implement a car rental system using Naming conventions and follow Hungarian Notation as standard. Include fields for car details, rental duration, and customer information and methods for editing and printing these details.

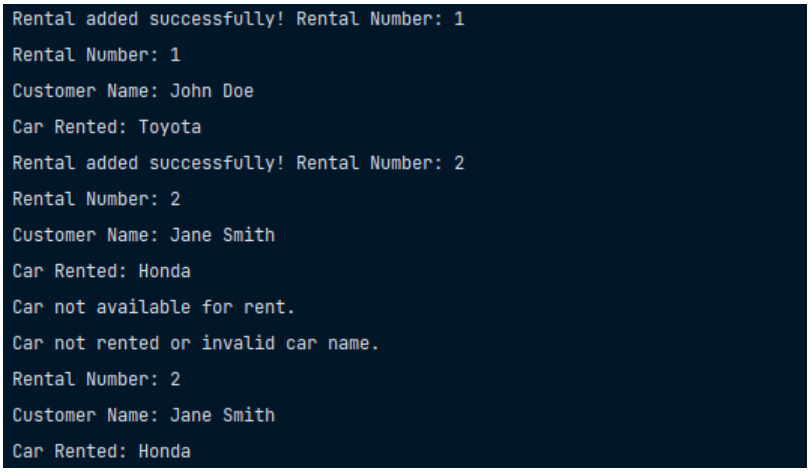
Code:

```
package SCD_labs;

class CarRentalSystem {
    String CustomerName;
    int RentalNo = 0;
    String CarRented;
    String CarsAvailable[] = {"Toyota", "Honda", "Nissan"};

    public void AddRental(String customerName, String carRented) {
        if (CarAvailable(carRented)) {
            this.CustomerName = customerName;
            this.RentalNo++;
            this.CarRented = carRented;
            CarsAvailable[FindCarIndex(carRented)] = null;
            System.out.println("Rental added successfully! Rental Number: " + Re
ntalNo);
        } else {
            System.out.println("Car not available for rent.");
        }
    }

    public void RemoveRental(String carRented) {
        if (this.CarRented != null && this.CarRented.equals(carRented)) {
            CarsAvailable[FindCarIndex(carRented)] = carRented;
            this.CustomerName = null;
            this.CarRented = null;
            System.out.println("Rental removed successfully for car: " + carRent
ed);
        } else {
```

```
            System.out.println("Car not rented or invalid car name.");
        }
    }

    public void ShowRentals() {
        if (this.CustomerName != null) {
            System.out.println("Rental Number: " + RentalNo);
            System.out.println("Customer Name: " + CustomerName);
            System.out.println("Car Rented: " + CarRented);
        } else {
            System.out.println("No active rentals found.");
        }
    }

    private boolean CarAvailable(String car) {
        for (String availableCar : CarsAvailable) {
            if (availableCar != null && availableCar.equals(car)) {
                return true;
            }
        }
        return false;
    }

    private int FindCarIndex(String car) {
        for (int i = 0; i < CarsAvailable.length; i++) {
            if (CarsAvailable[i] != null && CarsAvailable[i].equals(car)) {
                return i;
            }
        }
        return -1;
    }
}

class Main {
    public static void main(String[] args) {
        CarRentalSystem rentalSystem = new CarRentalSystem();

        rentalSystem.AddRental("John Doe", "Toyota");
        rentalSystem.ShowRentals();

        rentalSystem.AddRental("Jane Smith", "Honda");
        rentalSystem.ShowRentals();

        rentalSystem.AddRental("Mike Jones", "Ford");

        rentalSystem.RemoveRental("Toyota");
        rentalSystem.ShowRentals();

        rentalSystem.RemoveRental("Honda");
        rentalSystem.ShowRentals();
```

```
        rentalSystem.RemoveRental("Nissan");
    }
}
```

Output:



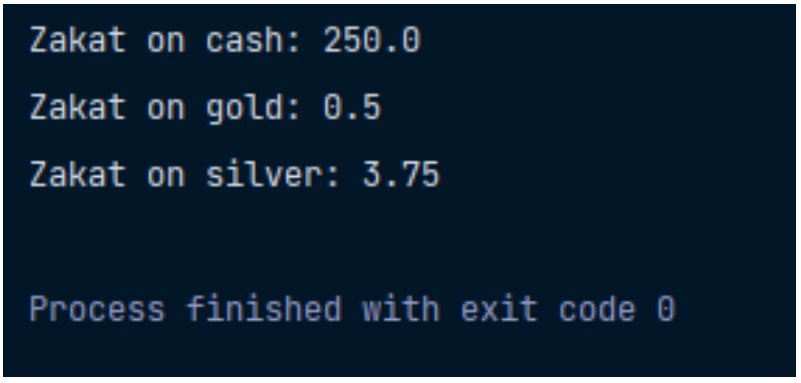
Q.2: Implement Zakat Calculator using Naming conventions follow snake case Notation as standard.

Code:

```
package SCD_labs;

public class Lab7Q2 {
    public void zakat_cash(int amount) {
        float percent = 2.5F;
        float zakat_amount = amount * percent / 100;
        System.out.println("Zakat on cash: " + zakat_amount);
    }

    public void zakat_gold(float tola) {
        float nisab = 7.5F;
        if (tola >= nisab) {
            float zakat_amount = tola * 2.5F / 40;
            System.out.println("Zakat on gold: " + zakat_amount);
        } else {
            System.out.println("Gold is less than Nisab. No Zakat payable.");
        }
    }

    public void zakat_silver(float tola) {
        float nisab = 52.5F;
        if (tola >= nisab) {
            float zakat_amount = tola * 2.5F / 40;
            System.out.println("Zakat on silver: " + zakat_amount);
        } else {
            System.out.println("Silver is less than Nisab. No Zakat payable.");
```

```
        }
    }

    public static void main(String[] args) {
        Lab7Q2 zakat = new Lab7Q2();

        zakat.zakat_cash(10000);
        zakat.zakat_gold(8);
        zakat.zakat_silver(60);
    }
}
```

Output:



Q.3: Give coding examples of TOdo comments and Comments.

Todo Comments:

```
// TODO: Test case for cash with amount below Zakat threshold
zakat.zakat_cash(10000);

// TODO: Test case for gold with weight exceeding Nisab but below Zakat exemptio
n
zakat.zakat_gold(8);

// TODO: Test case for silver with weight exceeding Nisab but below Zakat exempt
ion
zakat.zakat_silver(60);
```

Redundant Comments:

```
// String CustomerName; // Redundant comment - variable name is clear
int RentalNo = 0; // Redundant comment - variable name and initialization are cl
ear
// String CarRented; // Redundant comment - variable name is clear
String CarsAvailable[] = {"Toyota", "Honda", "Nissan"};
```