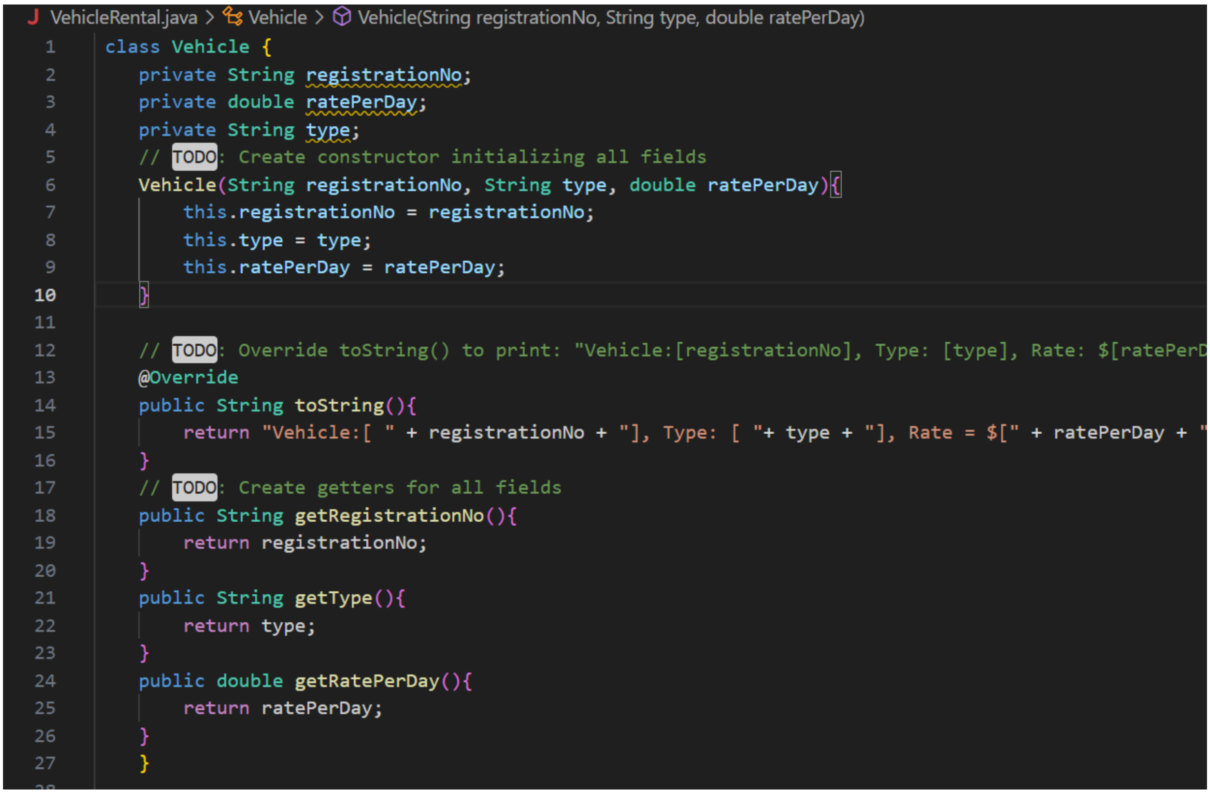
Week 9 - S9 - Advanced OOP - Object Class Methods, Inner Classes - Practice Problem

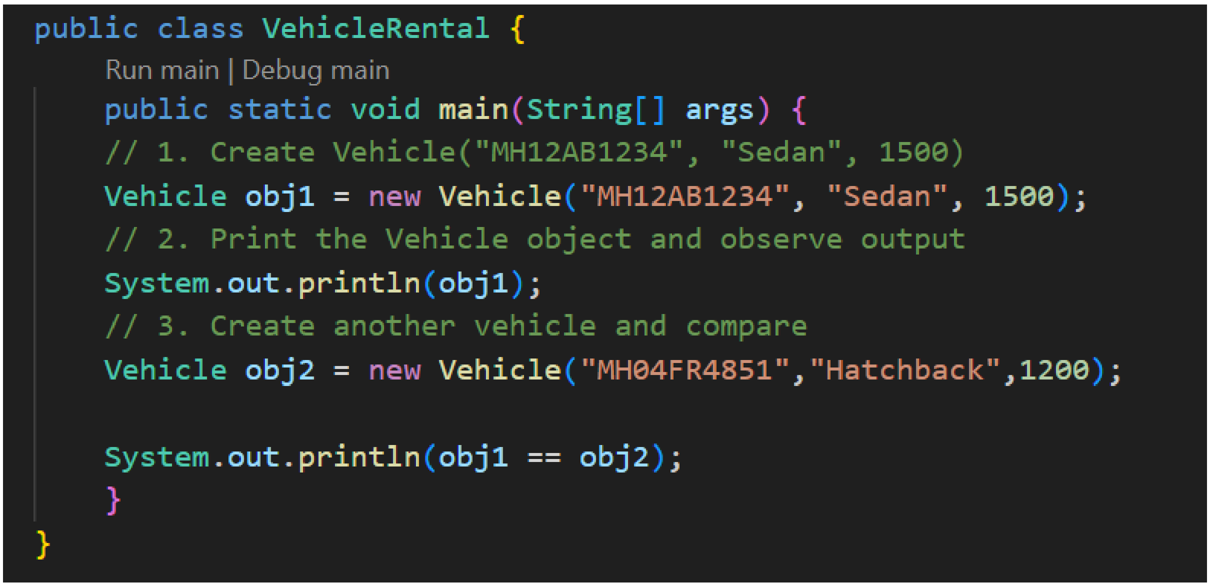
Name: Ramesh Harisabapathi Chettiar

Date of Submission:09/10/25

QNO1→

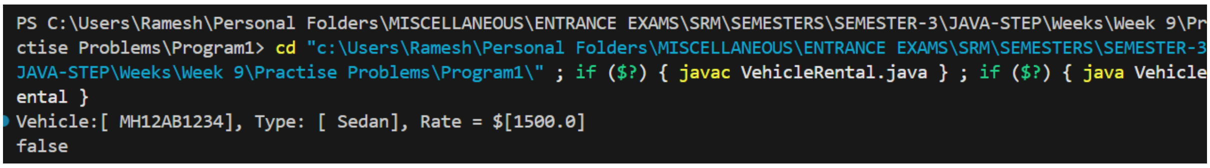

VehicleRental.java > Vehicle > Vehicle(String registrationNo, String type, double ratePerDay)

```java
class Vehicle {
    private String registrationNo;
    private double ratePerDay;
    private String type;
    // TODO: Create constructor initializing all fields
    Vehicle(String registrationNo, String type, double ratePerDay){
        this.registrationNo = registrationNo;
        this.type = type;
        this.ratePerDay = ratePerDay;
    }

    // TODO: Override toString() to print: "Vehicle:[registrationNo], Type: [type], Rate: $[ratePerD
    @Override
    public String toString(){
        return "Vehicle:[ " + registrationNo + "], Type: [ "+ type + "], Rate = $[" + ratePerDay + "
    }
    // TODO: Create getters for all fields
    public String getRegistrationNo(){
        return registrationNo;
    }
    public String getType(){
        return type;
    }
    public double getRatePerDay(){
        return ratePerDay;
    }
    }
```



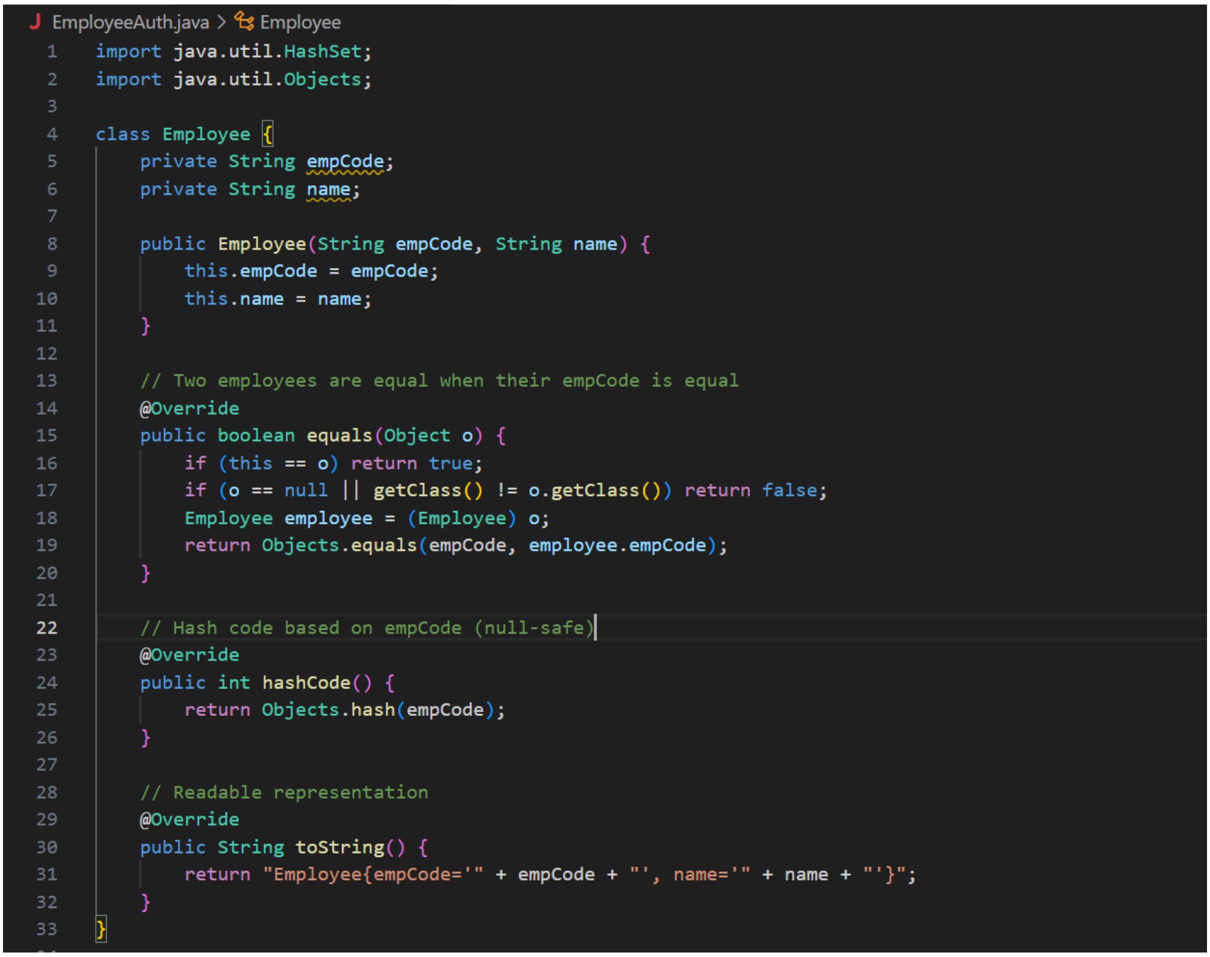
```java
public class VehicleRental {
    public static void main(String[] args) {
    // 1. Create Vehicle("MH12AB1234", "Sedan", 1500)
    Vehicle obj1 = new Vehicle("MH12AB1234", "Sedan", 1500);
    // 2. Print the Vehicle object and observe output
    System.out.println(obj1);
    // 3. Create another vehicle and compare
    Vehicle obj2 = new Vehicle("MH04FR4851","Hatchback",1200);

    System.out.println(obj1 == obj2);
    }
}
```

OUTPUT→

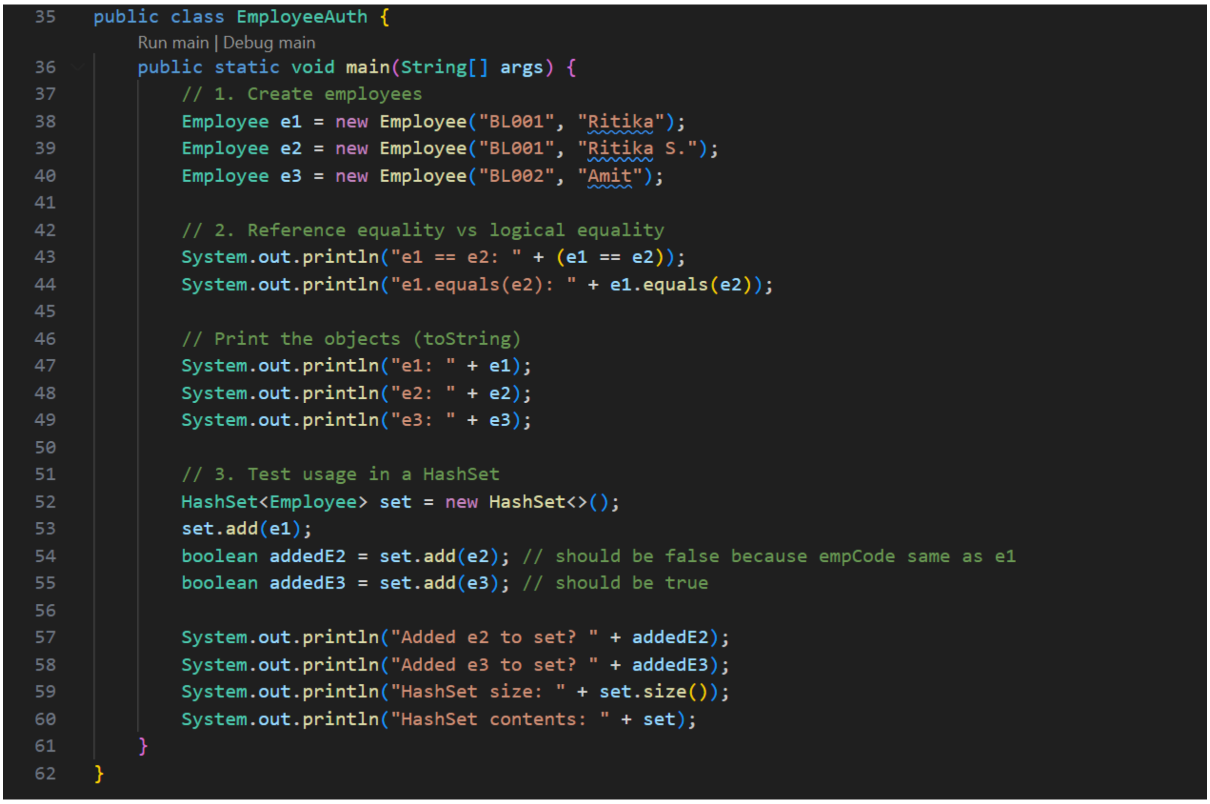


```
PS C:\Users\Ramesh\Personal Folders\MISCELLANEOUS\ENTRANCE EXAMS\SRM\SEMESTERS\SEMESTER-3\JAVA-STEP\Weeks\Week 9\Pr
ctise Problems\Program1> cd "c:\Users\Ramesh\Personal Folders\MISCELLANEOUS\ENTRANCE EXAMS\SRM\SEMESTERS\SEMESTER-3
JAVA-STEP\Weeks\Week 9\Practise Problems\Program1\" ; if ($?) { javac VehicleRental.java } ; if ($?) { java Vehicle
ental }
Vehicle:[ MH12AB1234], Type: [ Sedan], Rate = $[1500.0]
false
```

QNO2→

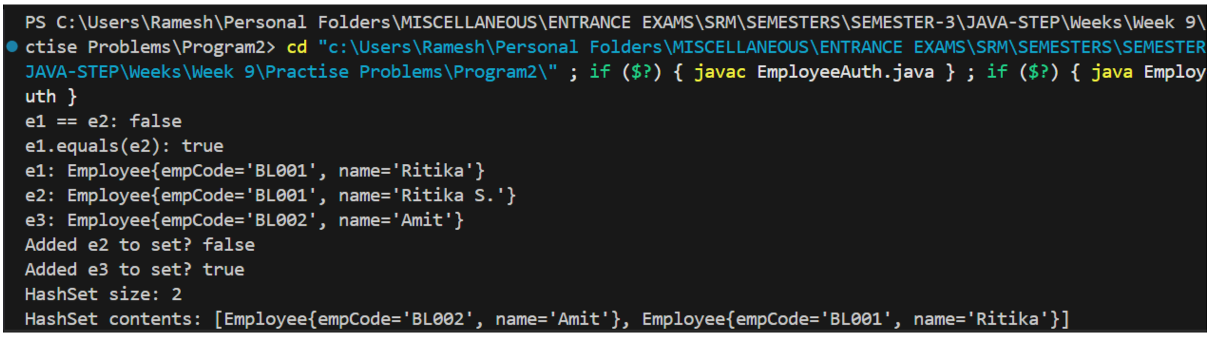


```java
import java.util.HashSet;
import java.util.Objects;

class Employee {
    private String empCode;
    private String name;

    public Employee(String empCode, String name) {
        this.empCode = empCode;
        this.name = name;
    }

    // Two employees are equal when their empCode is equal
    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null || getClass() != o.getClass()) return false;
        Employee employee = (Employee) o;
        return Objects.equals(empCode, employee.empCode);
    }

    // Hash code based on empCode (null-safe)
    @Override
    public int hashCode() {
        return Objects.hash(empCode);
    }

    // Readable representation
    @Override
    public String toString() {
        return "Employee{empCode='" + empCode + "', name='" + name + "'}";
    }
}
```



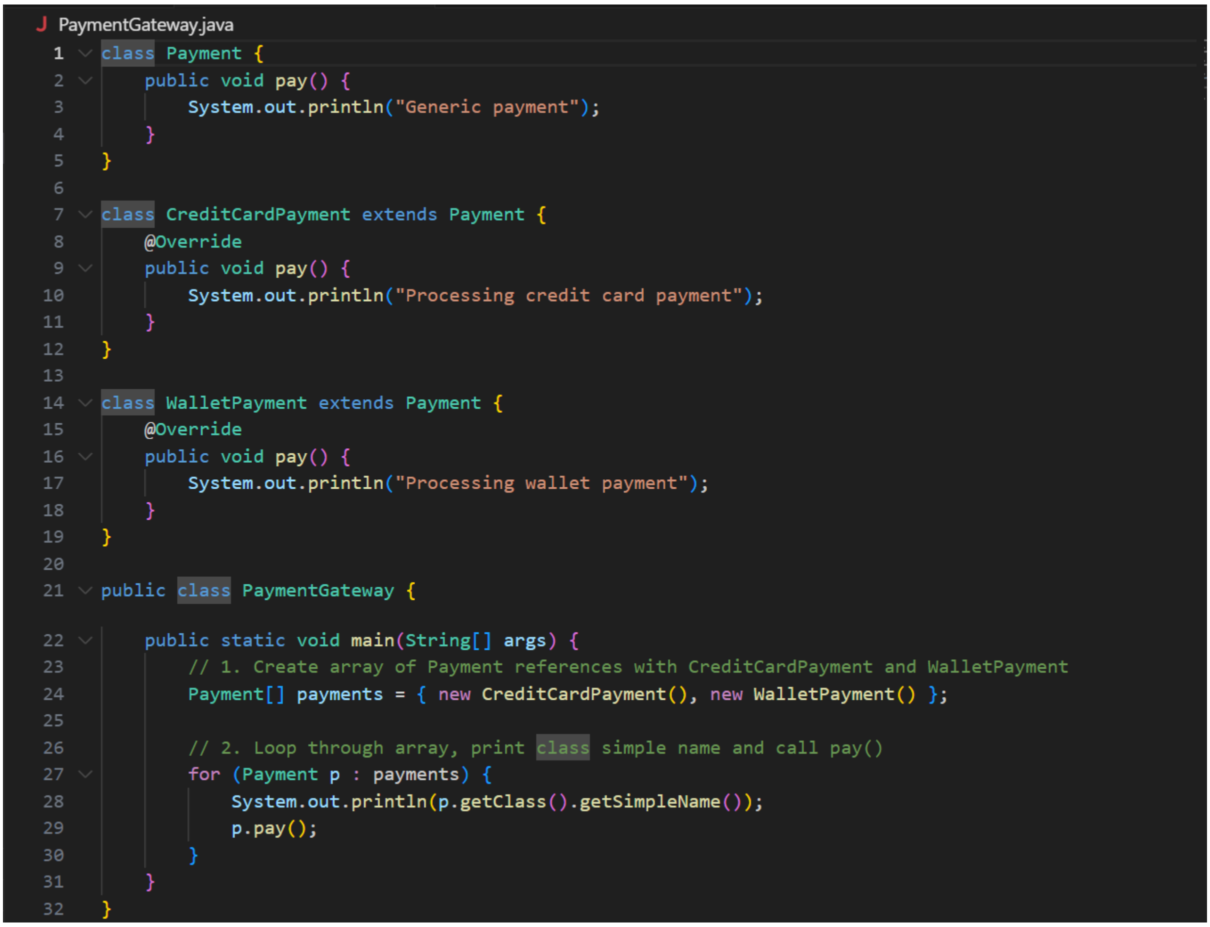
```java
public class EmployeeAuth {
    public static void main(String[] args) {
        // 1. Create employees
        Employee e1 = new Employee("BL001", "Ritika");
        Employee e2 = new Employee("BL001", "Ritika S.");
        Employee e3 = new Employee("BL002", "Amit");

        // 2. Reference equality vs logical equality
        System.out.println("e1 == e2: " + (e1 == e2));
        System.out.println("e1.equals(e2): " + e1.equals(e2));

        // Print the objects (toString)
        System.out.println("e1: " + e1);
        System.out.println("e2: " + e2);
        System.out.println("e3: " + e3);

        // 3. Test usage in a HashSet
        HashSet<Employee> set = new HashSet<>();
        set.add(e1);
        boolean addedE2 = set.add(e2); // should be false because empCode same as e1
        boolean addedE3 = set.add(e3); // should be true

        System.out.println("Added e2 to set? " + addedE2);
        System.out.println("Added e3 to set? " + addedE3);
        System.out.println("HashSet size: " + set.size());
        System.out.println("HashSet contents: " + set);
    }
}
```

OUTPUT→



```
PS C:\Users\Ramesh\Personal Folders\MISCELLANEOUS\ENTRANCE EXAMS\SRM\SEMESTERS\SEMESTER-3\JAVA-STEP\Weeks\Week 9\
ctise Problems\Program2> cd "c:\Users\Ramesh\Personal Folders\MISCELLANEOUS\ENTRANCE EXAMS\SRM\SEMESTERS\SEMESTER
JAVA-STEP\Weeks\Week 9\Practise Problems\Program2\" ; if ($?) { javac EmployeeAuth.java } ; if ($?) { java Employ
uth }
e1 == e2: false
e1.equals(e2): true
e1: Employee{empCode='BL001', name='Ritika'}
e2: Employee{empCode='BL001', name='Ritika S.'}
e3: Employee{empCode='BL002', name='Amit'}
Added e2 to set? false
Added e3 to set? true
HashSet size: 2
HashSet contents: [Employee{empCode='BL002', name='Amit'}, Employee{empCode='BL001', name='Ritika'}]
```

QNO3→



```java
class Payment {
    public void pay() {
        System.out.println("Generic payment");
    }
}

class CreditCardPayment extends Payment {
    @Override
    public void pay() {
        System.out.println("Processing credit card payment");
    }
}

class WalletPayment extends Payment {
    @Override
    public void pay() {
        System.out.println("Processing wallet payment");
    }
}

public class PaymentGateway {

    public static void main(String[] args) {
        // 1. Create array of Payment references with CreditCardPayment and WalletPayment
        Payment[] payments = { new CreditCardPayment(), new WalletPayment() };

        // 2. Loop through array, print class simple name and call pay()
        for (Payment p : payments) {
            System.out.println(p.getClass().getSimpleName());
            p.pay();
        }
    }
}
```