

Section B | FA24-BCS-068

Muhammad Hammad Sarwar

Assignment # 1



Submitted to: Muhammad Shahid Bhatti  
Course Title: Object Oriented Programming

May 1, 2025

**Mistake review**

**Logical Errors:** I incorrectly made **Vehicle inherit from Owner**, which is wrong.

 I passed **owner ID and permit ID manually**, but the requirement was to auto-generate them using static counters.

 I allowed **multiple instances of ParkingSystem**. The correct design required singleton.

 I didn’t check for **duplicate vehicle license plates**.

 I didn’t **add an array in ParkingZone** to store up to 5 vehicles.

 I didn’t implement **object cloning like shallow and deep copy in Vehicle.**

**Syntax Errors:**

 I initialized **Id = 0000**, which uses a leading zero.

 In **PermitHolder.toString(),** I left the return string with incomplete output.

 I didn’t **properly format the output strings** to match the sample.

 I accessed **super.name directly**.

**Missed Requirements**:  
 I didn’t implement **the singleton pattern** in the ParkingSystem class, which is user to ensure only one instance exists.

 I missed **displaying a warning** when a second attempt to create the ParkingSystem was made.

 I didn’t **print an error message** when adding a duplicate license plate.

 I used manual zone IDs **instead of generating auto-named ones like Z1, Z2**.

 I didn’t include **a proper vehicle storage mechanism in ParkingZone**.

**Updated Code:**

// Person.java

public class Person {

    String name;

    Person(String name) {

        this.name = name;

    }

}

// Supervisor.java

public class Supervisor extends Person {

    int yearOfExperence;

    Supervisor(String name, int yearOfExperence) {

        super(name);

        this.yearOfExperence = yearOfExperence;

    }

    @Override

    public String toString() {

        return "Supervisor: Name : " + this.name + ", Experence: " + this.yearOfExperence;

    }

}

// Owner.java

public class Owner extends Person {

    private static int counter = 1;

    String Id;

    Owner(String name) {

        super(name);

        this.Id = String.format("O%03d", counter++);

    }

    @Override

    public String toString() {

        return "Owner : [ Name: " + this.name + ", Owner Id: " + this.Id + " ]";

    }

}

C:\Users\ahmad raza\AppData\Local\Microsoft\Windows\Clipboard\HistoryData\{21FDC235-38A0-468E-AC60-F37B710554D7}\{8CDBE17E-44C8-47A9-B3F3-8F3819A12C6D}\ResourceMap\{B5259DC7-F97B-4490-BC56-850323D12791}

// ParkingSystem.java

import java.util.ArrayList;

public class ParkingSystem {

    private static ParkingSystem instance = null;

    private String campus;

    private Supervisor supervisor;

    private ArrayList<ParkingZone> zones = new ArrayList<>();

    private ArrayList<PermitHolder> permitHolders = new ArrayList<>();

    private ParkingSystem(String campus, Supervisor supervisor) {

        if (supervisor == null) {

            throw new IllegalArgumentException("System must have a supervisor assigned.");

        }

        this.campus = campus;

        this.supervisor = supervisor;

    }

    public static ParkingSystem getInstance(String campus, Supervisor supervisor) {

        if (instance == null) {

            instance = new ParkingSystem(campus, supervisor);

        } else {

            System.out.println("Warning: ParkingSystem instance already exists. Returning existing instance.");

        }

        return instance;

    }

    public void addZone(ParkingZone z) {

        zones.add(z);

    }

    public void addPermitHolder(PermitHolder p) {

        permitHolders.add(p);

    }

    @Override

    public String toString() {

        String output = "Campus: " + campus + "\n" + supervisor + "\n";

        for (ParkingZone z : zones) {

            output += z.toString();

        }

        output += "Permit Holders:\n";

        for (PermitHolder p : permitHolders) {

            output += p.toString() + "\n";

        }

        return output;

    }

}

C:\Users\ahmad raza\AppData\Local\Microsoft\Windows\Clipboard\HistoryData\{21FDC235-38A0-468E-AC60-F37B710554D7}\{8CDBE17E-44C8-47A9-B3F3-8F3819A12C6D}\ResourceMap\{B5259DC7-F97B-4490-BC56-850323D12791}

// PermitHolder.java

public class PermitHolder extends Person {

    private static int idCounter = 1000;

    private int id;

    PermitHolder(String name) {

        super(name);

        this.id = idCounter++;

    }

    public int getId() {

        return this.id;

    }

    @Override

    public String toString() {

        return "[ Name: " + this.name + ", Permit ID: " + this.id + " ]";

    }

}

C:\Users\ahmad raza\AppData\Local\Microsoft\Windows\Clipboard\HistoryData\{21FDC235-38A0-468E-AC60-F37B710554D7}\{8CDBE17E-44C8-47A9-B3F3-8F3819A12C6D}\ResourceMap\{B5259DC7-F97B-4490-BC56-850323D12791}

// ParkingZone.java

public class ParkingZone {

    private static int zoneCounter = 1;

    private String Id;

    private Vehicle[] vehicles = new Vehicle[5];

    private int count = 0;

    ParkingZone() {

        this.Id = "Z" + zoneCounter++;

    }

    public void addVehicle(Vehicle v) {

        if (count < 5 && v != null) {

            vehicles[count++] = v;

        }

    }

    @Override

    public String toString() {

        String result = "Parking Zone: " + Id + "\n";

        for (int i = 0; i < count; i++) {

            result += "  " + vehicles[i] + "\n";

        }

        return result;

    }

}

C:\Users\ahmad raza\AppData\Local\Microsoft\Windows\Clipboard\HistoryData\{21FDC235-38A0-468E-AC60-F37B710554D7}\{8CDBE17E-44C8-47A9-B3F3-8F3819A12C6D}\ResourceMap\{B5259DC7-F97B-4490-BC56-850323D12791}

// Vehicle

public class Vehicle {

    private static String[] existingPlates = new String[100];

    private static int plateCount = 0;

    private String plate;

    private String type;

    private Owner owner;

    Vehicle(String plate, String type, Owner owner) {

        if (isDuplicatePlate(plate)) {

            System.out.println("Error: Duplicate license plate " + plate + " is not allowed.");

            return;

        }

        this.plate = plate;

        this.type = type;

        this.owner = owner;

        existingPlates[plateCount++] = plate;

    }

    private boolean isDuplicatePlate(String plate) {

        for (int i = 0; i < plateCount; i++) {

            if (existingPlates[i].equals(plate)) {

                return true;

            }

        }

        return false;

    }

    public Vehicle shallowCopy() {   // Shallow copy

        return new Vehicle(this.plate + "\_COPY", this.type, this.owner);

    }

    public Vehicle deepCopy() {     // Deep copy

        Owner newOwner = new Owner(this.owner.name);

        return new Vehicle(this.plate + "\_COPY", this.type, newOwner);

    }

    public Owner getOwner() {

        return owner;

    }

    @Override

    public String toString() {

        return "Vehicle: [ License plate: " + this.plate + ", Type: " + this.type + ", " +

                (owner != null ? owner.toString() : "No owner") + " ]";

    }

}

C:\Users\ahmad raza\AppData\Local\Microsoft\Windows\Clipboard\HistoryData\{21FDC235-38A0-468E-AC60-F37B710554D7}\{8CDBE17E-44C8-47A9-B3F3-8F3819A12C6D}\ResourceMap\{B5259DC7-F97B-4490-BC56-850323D12791}

// Main.java

public class Main {

    public static void main(String[] args) {

        Supervisor supervisor = new Supervisor("Dr. Ayesha", 5);

        ParkingSystem system = ParkingSystem.getInstance("CUI Lahore", supervisor);

        ParkingSystem secondAttempt = ParkingSystem.getInstance("ShouldNotWork", supervisor);

        ParkingZone zone1 = new ParkingZone();

        ParkingZone zone2 = new ParkingZone();

        Owner owner1 = new Owner("Ali");

        Owner owner2 = new Owner("Zara");

        Vehicle v1 = new Vehicle("LEA123", "Car", owner1);

        Vehicle v2 = new Vehicle("LEB456", "Bike", owner2);

        Vehicle v3 = new Vehicle("LEA123", "Car", owner1);

        zone1.addVehicle(v1);

        zone1.addVehicle(v2);

        zone2.addVehicle(v3);

        system.addZone(zone1);

        system.addZone(zone2);

        PermitHolder ph1 = new PermitHolder("Sara");

        PermitHolder ph2 = new PermitHolder("Hassan");

        system.addPermitHolder(ph1);

        system.addPermitHolder(ph2);

        System.out.println(system);

    }

}