

|  |  |
| --- | --- |
| name | yar muhammad awaim |
| id | CIIT/FA24-BCS-113/LHR |
| COURSE NAME | OBJECT ORIENTED PROGRAMMING |
| COURSE INSTRUCTOR | DR. MUHAMMAD SHAHID BHATTI |
| ASSIGNMENT TITLE | REVISION OF MIDTERM PROBLEM |
| SUBMISSION DATE | FEBRUARY 25, 2025 |
| DEPARTMENT | DEPARTMENT OF COMPUTER SCIENCE |
| FACULTY | FACULTY OF INFORMATION TECHNOLOGY |
| INSTITUTE | COMSATS UNIVERSITY ISLAMABAD, LAHORE CAMPUS |

Contents

[Mistake Review: 3](#_Toc197010683)

[Code: 5](#_Toc197010684)

[Person.java 5](#_Toc197010685)

[Owner.java 5](#_Toc197010686)

[Supervisor.java 6](#_Toc197010687)

[ParkingZone.java 7](#_Toc197010688)

[PermitHolder.java 8](#_Toc197010689)

[Vehicle.java 9](#_Toc197010690)

[ParkingSystem.java 11](#_Toc197010691)

[Main.java 13](#_Toc197010692)

[Console Output: 15](#_Toc197010693)

# Mistake Review:

**Singleton Problem in** ParkingSystem

* + The ParkingSystem class didn’t limit itself to one instance. It had a thisInstance variable but missed a private constructor and a static getInstance method to control it.

**Supervisor Constructor Missing Name**

* + The Supervisor constructor only took yearsOfExperience and left out name, even though Supervisor comes from Person, which needs a name. The main method tried to give both, so it didn’t work.

**ParkingZone Constructor Issue**

* + The ParkingZone constructor needed zoneID and vehicles, but the main method tried to use one with no arguments. It also couldn’t make unique zone IDs like Z1, Z2 on its own.

**No Duplicate Check for License Plates in** Vehicle

* + The Vehicle class didn’t stop duplicate license plates. It used "A1", "A2", etc., instead of the given license plate, allowing repeats like "LEA123" without showing an error.

**Wrong Array in** ParkingSystem

* + The addZone method put ParkingZone objects into the permitHolders array instead of the parkingZones array, mixing things up.

**No Auto IDs for** Owner **and Wrong** PermitHolder **IDs**

* + The Owner class didn’t create unique IDs like O001, O002. The PermitHolder class used a String ID starting at 1, but the example output wanted numbers starting at 1000.

**Missing** addVehicle **Method in** ParkingZone

* + The ParkingZone class had no addVehicle method to add vehicles to its array. It only used the constructor, which didn’t let me add vehicles later.

**Bad** toString **Methods**

* + The toString methods in classes like ParkingSystem, ParkingZone, and Vehicle didn’t show info right. They missed showing empty slots with null and added extra braces or wrong details.

**Vehicle Constructor Missing License Plate**

* + The Vehicle constructor only used vehicleType and owner, skipping the licensePlate parameter from main. This stopped setting the right license plate.

**No Copy Methods in** Vehicle

* + The Vehicle class didn’t have shallowCopy or deepCopy methods, which were needed to show how to duplicate objects.

# Code:

## Person.java

public class Person {

private String name;

public Person(String name) {

this.name = name;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

}

## Owner.java

class Owner extends Person {

private static int counter = 1;

private String ownerId;

public Owner(String name) {

super(name);

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

}

public String getOwnerId() {

return ownerId;

}

@Override

public String toString() {

return "name: " + getName() + ", owner ID: " + ownerId;

}

}

## Supervisor.java

public class Supervisor extends Person {

private int yearsOfExperience;

public Supervisor(String name, int yearsOfExperience) {

super(name);

this.yearsOfExperience = yearsOfExperience;

}

public int getYearsOfExperience() {

return yearsOfExperience;

}

@Override

public String toString() {

return "Name: " + getName() + ", Experience: " + yearsOfExperience + " years";

}

}

## ParkingZone.java

public class ParkingZone {

private static int zoneCounter = 1;

private String zoneID;

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

private int vehicleCount = 0;

public ParkingZone() {

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

}

public void addVehicle(Vehicle vehicle) {

if (vehicle != null && vehicle.getLicensePlate() != null && vehicleCount < 5) {

vehicles[vehicleCount++] = vehicle;

}

}

public String getZoneID() {

return zoneID;

}

@Override

public String toString() {

StringBuilder sb = new StringBuilder();

sb.append("Zone ID: ").append(zoneID).append(", Vehicles: [");

for (int i = 0; i < 5; i++) {

if (vehicles[i] != null) {

sb.append(vehicles[i].toString());

} else {

sb.append("null");

}

if (i < 4) sb.append(", ");

}

sb.append("]");

return sb.toString();

}

}

## PermitHolder.java

public class PermitHolder extends Person {

private static int counter = 1000;

private int permitID;

public PermitHolder(String name) {

super(name);

this.permitID = counter++;

}

@Override

public String toString() {

return "name: " + getName() + ", permit ID: " + permitID;

}

}

## Vehicle.java

class Vehicle {

private static java.util.List<String> licensePlates = new java.util.ArrayList<>();

private String licensePlate;

private String vehicleType;

private Owner owner;

public Vehicle(String licensePlate, String vehicleType, Owner owner) {

if (licensePlates.contains(licensePlate)) {

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

this.licensePlate = null;

this.vehicleType = null;

this.owner = null;

} else {

this.licensePlate = licensePlate;

this.vehicleType = vehicleType;

this.owner = owner;

licensePlates.add(licensePlate);

}

}

public String getLicensePlate() {

return licensePlate;

}

public Vehicle shallowCopy() {

return new Vehicle(this.licensePlate, this.vehicleType, this.owner);

}

public Vehicle deepCopy() {

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

return new Vehicle(this.licensePlate, this.vehicleType, newOwner);

}

@Override

public String toString() {

if (licensePlate == null) {

return "license plate: null, type: null, owner: [null]";

}

return "license plate: " + licensePlate + ", type: " + vehicleType + ", owner: [" + owner.toString() + "]";

}

}

## ParkingSystem.java

class ParkingSystem {

private static ParkingSystem instance;

private String campusName;

private Supervisor supervisor;

private ParkingZone[] parkingZones = new ParkingZone[2];

private PermitHolder[] permitHolders = new PermitHolder[2];

private int zoneCount = 0;

private int permitHolderCount = 0;

private ParkingSystem() {}

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

if (instance == null) {

instance = new ParkingSystem();

instance.campusName = campusName;

instance.supervisor = supervisor;

} else {

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

}

return instance;

}

public void addZone(ParkingZone parkingZone) {

if (zoneCount < 2) {

parkingZones[zoneCount++] = parkingZone;

}

}

public void addPermitHolder(PermitHolder permitHolder) {

if (permitHolderCount < 2) {

permitHolders[permitHolderCount++] = permitHolder;

}

}

@Override

public String toString() {

StringBuilder sb = new StringBuilder();

sb.append("Campus: ").append(campusName).append("\n");

sb.append("Supervisor: ").append(supervisor.toString()).append("\n");

sb.append("Zones:\n");

for (ParkingZone zone : parkingZones) {

if (zone != null) {

sb.append(zone.toString()).append("\n");

}

}

sb.append("Permit Holders:\n");

for (PermitHolder ph : permitHolders) {

if (ph != null) {

sb.append(ph.toString()).append("\n");

}

}

return sb.toString();

}

}

## 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);

}

}

# Console Output:

C:\Users\yarmu\Desktop\FA24-BCS-113 Y.M. AWAIM\Assignment-1-Theory>java Main

Warning: ParkingSystem instance already exists. Returning existing instance.

Error: Duplicate license plate 'LEA123' is not allowed.

Campus: CUI Lahore

Supervisor: Name: Dr. Ayesha, Experience: 5 years

Zones:

Zone ID: Z1, Vehicles: [license plate: LEA123, type: Car, owner: [name: Ali, owner ID: O001], license plate: LEB456, type: Bike, owner: [name: Zara, owner ID: O002], null, null, null]

Zone ID: Z2, Vehicles: [null, null, null, null, null]

Permit Holders:

name: Sara, permit ID: 1000

name: Hassan, permit ID: 1001

C:\Users\yarmu\Desktop\FA24-BCS-113 Y.M. AWAIM\Assignment-1-Theory>

## Console Output Image:

