1. Design a java program parking lot as the image given below.

The objects present in our design are the Vehicles, ParkingLot, Slots, and Levels.
ParkingLot: There are 'x' levels or floors and 'y' slots per floor.
Levels: Each level is an independent entity with a floor number, its slotsand the lanes within it.
The number of lanes is designed on the basis of number of slots. In our case, 10 Slots comprise 1 lane.
Slots: One slot is independent of the other slot. The slot size matters for a vehicle to fill the slot. For example, a small slot cannot be filled by a large vehicle.
Vehicles: Object with the company name, vehicle number, and their type. A vehicle has the number plate and the properties of the company it is from.
Objective is finding the parking slot(the image is only representation).
1. Work on Synchronization
2. Thread Priority

```
import java.util.ArrayList;
import java.util.List;

class Vehicle {
   String number;
   String type;

   public Vehicle(String number, String type) {
       this.number = number;
       this.type = type;
   }

   @Override
   public String toString() {
       return "Vehicle [Number: " + number + ", Type: " + type + "]";
   }
}

class ParkingSlot {
   int totalSlots = 5;
   List<Vehicle> parkedVehicles = new ArrayList<>();

   public void parkVehicle(Vehicle vehicle) {
       if (parkedVehicles.size() < totalSlots) {
           parkedVehicles.add(vehicle);
```

```
            System.out.println(vehicle.type + " Vehicle " + vehicle.number + " is
parked.");
        } else {
            System.out.println("No available slots for " + vehicle.type + " Vehicle " +
vehicle.number);
        }
    }

    public void unparkVehicle(String vehicleNumber) {
        parkedVehicles.removeIf(vehicle -> vehicle.number.equals(vehicleNumber));
        System.out.println("Vehicle " + vehicleNumber + " is unparked.");
    }

    public void displayParkedVehicles() {
        System.out.println("Parked Vehicles:");
        for (Vehicle vehicle : parkedVehicles) {
            System.out.println(vehicle);
        }
    }
}

public class ParkingLotSystem1 {
    public static void main(String[] args) {
        ParkingSlot parkingLot = new ParkingSlot();

        parkingLot.parkVehicle(new Vehicle("car1", "Small"));
        parkingLot.parkVehicle(new Vehicle("car2", "Medium"));
        parkingLot.parkVehicle(new Vehicle("car3", "Large"));
        parkingLot.parkVehicle(new Vehicle("car4", "Small"));
        parkingLot.parkVehicle(new Vehicle("car5", "Medium"));
        parkingLot.parkVehicle(new Vehicle("car6", "Large"));

        parkingLot.unparkVehicle("car2");

        parkingLot.parkVehicle(new Vehicle("car7", "Small"));

        parkingLot.displayParkedVehicles();
    }
}
Small Vehicle car1 is parked.
Medium Vehicle car2 is parked.
Large Vehicle car3 is parked.
```

```
Small Vehicle car4 is parked.
Medium Vehicle car5 is parked.
No available slots for Large Vehicle car6
Vehicle car2 is unparked.
Small Vehicle car7 is parked.
Parked Vehicles:
Vehicle [Number: car1, Type: Small]
Vehicle [Number: car3, Type: Large]
Vehicle [Number: car4, Type: Small]
Vehicle [Number: car5, Type: Medium]
Vehicle [Number: car7, Type: Small]
tarunssunadoli@Mac CIE %
```