

## C# Developer- Home Task

### Scenario:

You've been hired to build the **backend logic** for a **smart parking system**. The system manages multiple parking lots, each with a set of parking spots. Users can park and unpark their vehicles, and the system should track spot availability and calculate parking fees.

---

### Your Tasks

#### 1. Design the Core Classes

Create the following classes with appropriate properties and methods:

- **ParkingLot**
    - Has a name, hourly rate, and a fixed number of parking spots.
    - Manages available and occupied spots.
  - **ParkingSpot**
    - Has a unique ID.
    - Can hold one vehicle at a time.
    - Tracks whether it's occupied.
  - **Vehicle**
    - Has a license plate (string).
    - Has a type (e.g., car, motorcycle).
  - **ParkingSession**
    - Tracks when a vehicle enters and exits a parking spot.
    - Calculates the total parking time and fee.
- 

#### 2. Implement the Following Logic

- When a **vehicle enters** a parking lot:
  - Assign it to an **available spot**.

- Start a **parking session**.
  - When a **vehicle leaves**:
    - End the parking session.
    - Calculate the **parking fee** based on the time spent and the parking lot's hourly rate.
- 

### 3. Constraints

Make sure your implementation follows these rules:

- A **parking spot** can hold **only one vehicle** at a time.
- Each **vehicle** has:
  - A **license plate** (string)
  - A **type** (e.g., car, motorcycle)
- Each **parking lot** has a **fixed number of spots**.
- Different parking lots may have **different hourly rates**.

### 4. Please answer the following questions:

1. What is the difference between **class** and **struct** in C#?
2. What is a try-catch-finally block, and when should you use it?
3. What is the difference between `Select`, `Where`, `FirstOrDefault`, and `Any`?
4. Explain the four principles of Object Oriented Programming: Encapsulation, Abstraction, Inheritance and Polymorphism.