Lab 7

To pass this exercise you must:

- Complete the exercise below.
- After completing the task, submit your zipped file to Canvas for assessment.
- Discuss your work with your tutor in the tutorial lab class for feedback.
- Submit by the **end of the tutorial** or by the **due date Sunday** of this tutorial week.

Individual Work

You may talk to other students about the work but must write your own code.

Task: Below are the updated project 1 requirements. Read it again and design a class diagram for project 1 including appropriate attributes, methods, relationships and multiplicities between classes.

Project 1 requirements: A small system is required that will help manage cars at a parking site for a company. You are to develop a system that includes the following three classes:

- CarPark class
- ParkingSlot class
- Car class

CarPark is responsible for maintaining a list of available parking slots. You should be able to find a slot, add a slot, delete a slot, and provide a list of all slots included in the car park (better to use toString). To find a slot, you should have *findBySlotID* & *findByCarRegNo* methods (they all return a ParkingSlot object).

A **parking slot** must have an identifier, which starts with a capital letter, followed by a three-digit number e.g. “D001”, “E123”. A parking slot also should know if a car and what car is parked in the slot. You must be able to add a car to the slot and remove a car from the slot. A toString method is needed for supporting list of slots (or called toString) in CarPark.

A **car** will be identified by its registration number. A registration number always starts with a capital letter, followed by a four-digit number e.g. “T1234”. A car should have its make, model, year, etc.

For this assessment, you should NOT maintain a list of parked cars in any of your classes.

Submission

You may use whatever tools to draw the class diagram or draw by hand on a piece of paper and then take a photo. To submit a zip of your file (a picture, a Word, or a PDF file) to the Canvas for assessment.

Marking scheme

1. A proper class diagram representations (6 marks)
   a: correct class names
   b: proper attributes (visibility, meaningful names, and types)
   c: proper methods (visibility, meaningful names, parameters, and return types)
2. Proper class relationships (2 marks)
3. Proper multiplicities (2 marks)