L7b

Re-submit Assignment

Due Oct 23, 2018 by 11:59pm **Points** 10 **Submitting** a file upload **File Types** zip
Available after Oct 15, 2018 at 8am

In all the service classes that you are to implement please make sure that all methods use `this` keyword when referring to instance variables.

1. Complete Programming Activity 2 - section 7.14. Use these files: **Airport.java** **AirportClient.java** **Pause.java**
   - Draw UML Class diagram for the Airport class
   - After completing the program provide answers to the Discussion Questions 1, 2, and 3 on page 428 (section 7.14 below Programming Activity 2).
2. Class `Auto` that you studied in this chapter is a service class that contains data and methods for use by applications/clients that need functionality for an auto with three atributes: `model`, `milesDriven`, and `gallonsOfGas`. The client class has a `main` method that uses the methods defined in the `Auto` class and it is called `AutoClient`. Using these classes as an example write a service class called `Course` that implements a concept of a course and has the following defined:
   a. **three fields**: a *code* (which **must** be one of "CS1", "CS2", "CS3" or "CS4"), a *description* (`String`), and a *number of credits* as `int` (for example 3).
   b. **default constructor**
   c. **secondary constructor**
   d. **accessor** methods for each field
   e. **mutator** methods for each field
   f. `toString` method
   g. `equals` method
   h. a "**business**" method called `level`, that returns either 1,2,3, or 4 based on the value of *code*
   i. write a client class called `CourseClient`. The client class has a `main` method that should call all the methods defined in the `Course` class to make sure that they work properly. Please remember to use *this* keyword where appropriate. Submit both classes for grading.
3. Write a class encapsulating the concept of a circle, assuming a circle has the following:
   a. **two fields**: a *point* which is an object of type `Point` (from `java.awt` package) representing the center of the circle, and the *radius* (`int`) of the circle
   b. **secondary constructor** (default constructor should not be implemented)
   c. **accessor** methods for each field
   d. **mutator** methods for each field
   e. `toString` method

f. `equals` method
g. a "**business"** method calculating and returning the *primeter* of the circle (*2 * π * radius*)
h. a "**business**" method calculating and returning the *area* of the circle ($\pi * radius^2$)
i. write a **client class** to test all the methods in your class