CORNERSTONE
International Community College of Canada

**Module 2**
**Java Programming - MADP 202**

Activity 6

**Due: 10:30pm, Monday, July 8rd, 2019**

# *Problem0*

Create a Java project with the sample code we did in the lecture in the last session (About Delegation). Then make the following changes:

1. Create a new service provider called, CarRepairServiceProvider which does a car repair service. The car repair service contains a) description and b) car model. Add all required classes to let the client request for a car repair service. ApplicationDriver class and also write at-least one unit test for it in a test class.
2. Create a new service provider called, HomeServiceProvider which performs three (3) types of home services 1) CookingService 2) WashingService 3) BabySittingService. Add 2 arbitrary properties for each source and then add other required classes.
3. Create a new service provider called, StudentServicesProvider which provide one student service. Add two arbitrary properties for the StudentService. In addition to the StudentService, the provider also provide an optional service which is informing the client know when the requested service is completed.

   For all 3 items above, write some test scenarios in the ApplicationDriver class (No need for writing unit test).
   Also please do not add any extra properties in the Client class. The only new properties (instance variables) you need to add to the Client class are the references to the providers (similar to the CleaningService and PaintingService references already available in the class Client).

# *Problem1*

In Problem 1 you are going to improve the code you have done in Problem0.

1- Create a new project with the existing code from Problem 0.
2- Create a Service class as a parent (base) class and make all your service classes (CleaningService, PaintingService, CarRepairService,…) inherits from this parent class. Make any other necessary
3- Create an interface which is the parent of all interfaces you have defined in your Problem0. Now think about this: Now, do you need the other interfaces at all anymore? If No, then remove them and make corresponding changes. If Yes, then make corresponding changes.

4- Create a parent (base) class called Provider which is the parent of all providers you have defined above. Make the corresponding changes.

Your changes should make the application more maintainable which means if you are going to add a new ServiceProvider and a Service, it should take less time and effort compared to the time and effort needed in the Problem0.

After the making the above changes your application should still be able to deliver the same functionalities as Problem 0.

# *Problem2*

Imagine you design a mobile view or a webpage which contains a table (Similar to the picture below). The table lists the students in our program. Create an application using the delegate approach which given a list of students it create the table. At the minimum you need two classes View and Table.

In order to create the table, the Table should know about the following information. Also Imagin the View has a list of students and it uses a Table object to show the list of Students.

- The View class has the list of students.
- The list of students is not fixed and could be arbitrary. In other word, if there are 10 students in the list, then the table has 10 rows. If there are only 2 students in the list, then table shows 2 rows and so on.
- The Table class is responsible for showing the information in the table.
- The Table requires the following information to be able to show the information
-
    - Student data. Each row represents one students
    - How many students are in the list
    - The fixed height of each row

| |
|---|
| Ali David, 80 |
| Peter Jackson, 79 |
| Donald Trump, 81 |
| William Chen, 76 |
| Ana Leu, 82 |

- As mentioned, each row in the table represents an item which contains the following properties 1- firstName 2-LastName 3- GPA