

# COMP212 – Programming 3

## Section 005 - Winter 2024

### Test 1

**Instructions: Be sure to read the following general instructions carefully:**

- This test should be completed individually by all the students.
- Please submit your solution through the “Test 1” drop box and name your submission according to the following pattern: studentID(yourlastname)\_MidTerm.zip.  
e.g., 300123456(smith)\_MidTerm.zip
- This is an open-book test, and you can use any resources (e.g., lecture notes, examples, your assignments, etc.) that you feel helpful.
- You are not allowed to communicate with anyone else during the test.
- Please bear in mind that you are responsible for securing your own code. You will be penalized if another person has your code, or similar to your code.
- Download the attached “MidTermTemplate.zip” file. It includes two projects for Question 1 and 2. Use these projects to answer the following questions.
- Note: You will be asked to demonstrate your solution.

### Rubric

Functionality	Marks
<b>Question 1</b> <ul style="list-style-type: none"><li>• Declare and initialize a dictionary of <b>Courses</b> and Obtain a dictionary collection of <b>String - Course</b> pairs by calling the static method <b>GetDictionary()</b> of the <b>Course</b> class. (1 mark)</li><li>• List all the courses from the dictionary. (1 mark)</li><li>• Add a new Course to the dictionary. Do not allow duplicates and print an error message if user tries to add a duplicate course in the dictionary. (1.5 marks)</li><li>• Remove course from the dictionary, based on the Course Code. (1 mark)</li><li>• Find a course (by Name). Notify the user if the course name does not exist. (2 marks)</li><li>• List all courses which have PreReq equals to “COMP123 COMP125” or “COMP123 COMP301” or “COMP247”. (1.5 marks)</li><li>• List of all the elective course (Hint: Core = false) (1.5 marks)</li><li>• List all courses, sorted by Course Code. Implement the compareTo method in the Course class to compare the course based on Course Code. (2.5 marks)</li></ul>	<b>12</b>
<b>Question 2</b> <p>In the Car.cs class, implement the following methods:</p> <ul style="list-style-type: none"><li>• Set a new price and notify all the customers with the information about the car and price change. (1.5 marks)</li><li>• Add a customer to the subscriber’s List. (1 mark)</li><li>• Remove a customer from the subscriber’s List. (1 mark)</li></ul>	<b>12</b>

<ul style="list-style-type: none"> <li>• Notify each customer if the car's qty is updated to be greater than zero. (1.5 marks)</li> </ul> <p>In the Customer.cs class, implement the following methods:</p> <ul style="list-style-type: none"> <li>• Implement a method that displays information when the Car is available, or Car's price is updated. To notify the customer, you can use Console.WriteLine(). (1.5 marks)</li> </ul> <p>In the Main method, add the following:</p> <ul style="list-style-type: none"> <li>• Create a new Car object, keeping the qty value "0" initially. (1 mark)</li> <li>• Create 2 customers' objects and make them subscribe to receive notification when the car's price or qty changes. (1 mark)</li> <li>• Call the method that sets the new price for the car and notify the customers. (1 mark)</li> <li>• Call the method that sets the new qty for the car and notify the customers only when qty is more than 0. (1 mark)</li> <li>• Remove 1 customer from the waiting list which means unsubscribe a customer to no longer receive notification related to above car. (1 mark)</li> <li>• Call the method again that sets the new price for the car and notify the customers. (0.5 mark)</li> </ul>	
<b>Overall (output, code readability, app usability, clarity of the code, etc.)</b>	<b>1</b>

### Question 1:

To complete this question, you must download the attached zip file. This file contains a **Course** class.

Declare and initialize a dictionary of **Courses** and Obtain a dictionary collection of **String - Course** pairs by calling the static method **GetDictionary()** of the **Course** class. Note, the type of your collection should be **Dictionary<string, Course>** and use Course Code as the key. (1 mark)

Implement the following code to make the application completely functional to allow users to:

- List all the courses from the dictionary. (1 mark)
- Add a new Course to the dictionary. Do not allow duplicates and print an error message if user tries to add a duplicate course in the dictionary. (1.5 marks)
- Remove course from the dictionary, based on the Course Code. (1 mark)
- Find a course (by Name). Notify the user if the course name does not exist. (2 marks)
- List all courses which have PreReq equals to "COMP123 COMP125" or "COMP123 COMP301" or "COMP247". (1.5 marks)
- List of all the elective course (Hint: Core = false) (1.5 marks)
- List all courses, sorted by Course Code. Implement the compareTo method in the Course class to compare the course based on Course Code. (2.5 marks)

## Question 2:

Complete the console app that solve the following problem using the delegates/events:

Two customers come to a Car Dealership for buying a Car. Unfortunately, at that time the Car is out of stock.

At the dealership, the customers are registered to be notified when the car is available. Customers are registered with a maximum acceptable price of the car. So, when the car is available, the dealership will send a notification to them.

When the car is available, the quantity changes, the dealership sends notifications to all the customers who are registered and are willing to accept the current price of the car.

In the Car.cs class, implement the following methods:

- Set a new price and notify all the customers with the information about the car and price change. (1.5 marks)
- Add a customer to the subscriber's List. (1 mark)
- Remove a customer from the subscriber's List. (1 mark)
- Notify each customer if the car's qty is updated to be greater than zero. (1.5 marks)

In the Customer.cs class, implement the following methods:

- Implement a method that displays information when the Car is available, or Car's price is updated. To notify the customer, you can use Console.WriteLine(). (1.5 marks)

In the Main method, add the following:

- Create a new Car object, keeping the qty value "0" initially. (1 mark)
- Create 2 customers' objects and make them subscribe to receive notification when the car's price or qty changes. (1 mark)
- Call the method that sets the new price for the car and notify the customers. (1 mark)
- Call the method that sets the new qty for the car and notify the customers only when qty is more than 0. (1 mark)
- Remove 1 customer from the waiting list which means unsubscribe a customer to no longer receive notification related to above car. (1 mark)
- Call the method again that sets the new price for the car and notify the customers. (0.5 mark)