

PROG2200 – Advanced OOP

Lab 1

Due Date: Sunday, Sept. 25th, 2023 @ 11:59pm

Assignment Value: 10% of overall course mark.

The goal of this assignment is to familiarize you with the basics of programming and Object-Oriented Programming in C#, git and Visual Studio.

This assignment relies on you following the **Learning C#** course on LinkedIn Learning (<https://www.linkedin.com/learning/learning-c-sharp-3/welcome?u=2149225>). You have an account with LinkedIn Learning associated with your NSCC account: See <https://subjectguides.nsc.ca/LinkedinLearning/getstarted>.

NOTE: this Assignment involves going through the first 4 modules of the **Learning C#** online course on your own, while completing the extra exercises that I have added below. I recommend watching a few videos, then do the exercises I provide before continuing to the next section/modules. Some of the modules in the course have Challenges that we will build on directly below. Otherwise, we will just build on the examples provided in the video. Each item will have a score associated with it which will accumulate to your final mark for this lab.

Note: Please organize your code so that each numbered section of the assignment is its own folder and Visual Studio project within your github repository. For example:

- Lab1
 - Module1
 - Section1
 - Section2

I recommend copying and renaming previous projects to build on what you did already. You may copy the resource files from LinkedIn Learning as a starting point. However, I recommend copying the starting files and not the solution so you get a chance to work through the code as it is explained in the video.

General

1. Demonstrate that you have made multiple commits as you went through the **Learning C#** course: at least 5. **(2 pts)** _____
2. Show that you have the right README and .gitignore for Visual Studio. Update the Readme file to state your name and your student number. Also, arrange your project according to the structure outlined above. **(2 pts)** _____

Module 1 (Basics)

1. Take the Challenge 1 (Multiple Choice) and add two more questions of your choices to it. **(2 pt2)** _____

PROG2200 – Advanced OOP

2. Alter Challenge 2 (Passcode) so that the user can change the password after authenticating.
Question: Is an else...if the best option for checking the secret? Simplify the code to not use else...if as described. (2 pts) _____
3. Change Challenge 3 (Loop Counting) to be only 2 loops (instead of 3) with a conditional. (2 pts) _____
4. In Challenge 4 (Student Manager) there are two arrays: studentNames and studentGrades. How could you replace these with a single two-dimensional array? I.e., Each row in the main array is itself an array of two elements: one for name, one for grade.
Hint: A 2D array is declared in C# as: `var array = new string[<size>,<size>]`.
You will need to remove parsing of ints for this exercise and store everything as a string as 2D arrays are of a specific type. (2 pts) _____

Module 2 (OOP in C#)

1. Challenge 5:
 - Change the Survey console app to be a full Zodiac Calculator. Requirements: Also take the day of the month as an input. Use a switch statement to calculate what star sign matches the month and date. You should validate that the user entered a number for month and date. (4 pts) _____

Module 3 Exception Handling & Debugging

1. Enhance the StudentManager program to have a custom error class called MyCustomException that extends FormatException. Make sure the main program code correctly handles the custom error class in a try-catch and reports the proper error message. (2 pts) _____
2. Demonstrate your knowledge of the debugger to your instructor. **(See the recording part)**
 - a. Comment out any try-catch and force a debug on exception.
 - b. Re-instate the commented-out code and demonstrate the following:
 - i. A breakpoint
 - ii. Step Over
 - iii. Step Into

Module 4

1. **(Enums – Module 4(sect. 1) -** Enhance the survey code to use an enum for the months: Jan = 0, Feb = 1, etc. Show it in the switch statement as well. (2 pts) _____
2. **(out vs ref – Module 4 (sects. 2-5)) –** Update the outreference class to have a static parseDouble function. This will take an input String and a double out. The function will explicitly return true if parsed successfully or false if not. In the Main method provide 2 calls one that sets the value properly, another which passes in your name and returns false. See below for the calling code: (2 pts) _____

```
var test = "1.5";  
var outDouble = 0.0;  
Console.WriteLine("Result: {0}", parseDouble(test, out outDouble));  
Console.WriteLine("Test type is {0}", outDouble.GetType());  
test = "Ronan";  
outDouble = 0.0;
```

PROG2200 – Advanced OOP

```
Console.WriteLine("Result: {0}", parseDouble(test, out outDouble));  
Console.WriteLine("Test value is {0}", outDouble);
```

3. **(Interfaces – Module 4(sect. 7))** – After completing the code in this section, extend by doing the following: move Student & Member to their own class files. Next, create an interface called IGraduate with a method of graduate(). Make student implement IGraduate as well as extend Member. Make sure the graduate() method prints to the console when called. Add a new class called Registrar that takes the list of students as <IGraduate>s and calls the graduate() method on each one in a graduateAll() method. This should be called at the end of the Program main method. The constructor for the Registrar class should look like the following: **(4 pts)** _____

```
public Registrar(List<IGraduate> grads)  
{  
    this.grads = grads;  
}
```

4. **(Events – Module 4(sect. 8))** – After completing the code in this section, extend by doing the following: The marketing department also wants to be notified when the Survey is completed. Add a second class called Marketing that will act as a listener to the Posted event that will print to the console - "Thank you for completing the survey. You are now subscribed to ten of our newsletters." **Demonstrate to your instructor in the video recording part.** **(2 pts)** _____

Video Recording

You will demonstrate your solutions (from scratch) to the following exercises as part of a code review using video recording:

- | | |
|-------------------------|----------------------|
| 1. Module 1, section 2 | (4 pts) _____ |
| 2. Module 3, section 2. | (4 pts) _____ |
| 3. Module 4, section 4. | (4 pts) _____ |

Final Score (Out of 40)
