BCPR283

## What is good about this course

This course is all about learning about how to be a programmer, and so is very useful.

I learned how to do unit testing in this course which was extremely relevant to my project at Telogis.

I enjoyed making the games in these courses: a fun and good way to learn about OO programming skills and the programming frameworks (C#). In BCPR283, I learned how to program in C# which very useful for the current project.

This course was helpful on a syntax and in a development point of view. We developed all prototypes in c#, as it was the language of choice for developing on windows devices. Aidan had not known how to program in c#, but I was able through my knowledge of setting up the Theseus vs Minotaur game how to make a system. He knew MVC, so we went from there. This was a buzz for me!

Also, the analysis, of thinking about the language from algorithms, patterns and dependencies was useful in determining how we were going to do certain things in the winforms version. It was through the need to be able to handle many things on screen at once that we worked on how to implement threading.

The experience I gained with C# .NET in this course helped greatly in understanding the back-end design of the Innovation Hub. I was able to confidently make changes to the code base, and understand the flow of data through the server.

The contents of this course definitely helped me in my internship project, from the very basics of C# to the advanced lessons that our tutor taught us like Flow Controls, Nested-loops, Arrays, Dictionaries, Methods, classes, Etc.

I build a game in windows form; that experience helps me a lot when I create game in this project.

## Recommendations

During this course we learnt how to write unit tests to ensure the quality of the code that was written. One thing we could be taught during this session is how to open up classes for testing. It was difficult to test certain aspects of our project with unit test due to the encapsulation settings. An explanation and experience with using the internal declaration could be helpful to students in the future who would like to open a class up for testing.

Also demonstrating how interfaces could be used to create “mock” classes that can behave exactly how you want them to when testing a class that has a dependency on one or more classes.

Larger focus on source control, Best practices, Different programs, and GUI’s, Git-flow?

These lessons will be nice to have in the following semesters:

* Setting up web client in C#.
* HTTP Post and Get request.
* Connecting C# and SQL Databases.
* Design Patterns like Factory pattern, Singleton, Etc.

Should have taught us how to develop web based applications with this language, would have been very helpful as asp.net is becoming more popular tool in developing web based applications.

I think it will be better if you can introduce more than one development platform during the class, for example Unity.