INFO1111: Computing 1A Professionalism

2023 Semester 1

Self-Learning Report

Submission number: 01

Github link: `https://github.com/BlindTerran/INFO1111_Self-Learning_Report`

Student name	Hongting SU
Student ID	530554978
Topic	C#
Levels already achieved	None
Levels in this report	A

Contents

Instructions

Important: This section should be removed prior to submission.

You should use this LaTeX template to generate your self-learning report. Keep in mind the following key points:

- **Submissions**: There will be three opportunities during the semester to submit this report. For each submission you can attempt 1 or 2 levels. Each submission should use the same report, but amended to include new information.
- **Assessment**: In order to achieve level B, you must first have achieved level A, and so on for each level up to level D. This means that we will not assess a higher level until a lower level has been achieved (though we will review one level higher and give you feedback to help you in refining your work).
- **Minimum requirement**: Remember that in order to pass the unit, you must achieve at least level A in the self-learning (unless you achieve level B in both the skills and knowledge categories).
- **Using this template**: When completing each section you should remove the explanation text and replace it with your material.
- **Referencing**: You should also ensure that any resources you use are suitably referenced, and references are included into the reference list at the end of this document. You should use the IEEE reference style [**?**] (the reference included here shows you how this can be easily achieved).

1. Level A: Initial Understanding

1.1. Level A Demonstration

Note: This must be the same as was in your topic approval 1. Watch YouTube tutorial for setting up visual studio for C# development 2. Learning basic C# variables, control structures and loops on W3Schools 3. Build simple programs using concepts of variables, control structures and loops using C#

1.2. Learning Approach

Before learning I have to have some information about this language, I will find out how this language will be useful in the real world, and what scenarios C# is used but not other languages. Is C# an OOP language? What are the programming paradigms and how to choose the most appropriate paradigm?

Before developing any program using C#, I will have to learn the basics of C, which are the variable types, different control structures, if-else statements, loops, and functions.

Since C# is an OOP language, it is needed for me to learn the concepts of OOP such as class and object, inheritance, encapsulation, and polymorphism.

After I finish those learning processes, I will apply those concepts to building programs to solve some simple computational problems.

1.3. Challenges and Difficulties

I have been learning Python since this semester, there will be some challenges in transitioning from Python to C#. Python and C# have differences in syntax, paradigm, etc... I have to overcome the differences while learning. C# is more verbose than Python, such as variable and function declaration, and loops; In C# the method signature includes the method's return type; variables are explicitly declared before it is assigned values.

It always happens to me that I forget to initialise a variable without declaring its type and forget the format of creating a method. But once I have more practice I will be more familiar with the syntax of C#.

1.4. Learning Sources

I will learn C# primarily from Microsoft C# Docs, this is the official documents provided by Microsoft. It has official explanations of C# that cover almost every concept in C# and can be useful as a guide to the language.

The secondary learning resource will be W3Schools, different from Microsoft C# Docs, it has structured learning topics and uses more beginner-friendly languages rather than formal and jargon-heavy expressions that are widely used in C# Docs. Another resource I will use is YouTube since videos are better at helping me to visualise difficult concepts.

Learning Source - What source did you use? (Note: Include source details such as links to websites, videos etc.).	Contribution to Learning - How did the source contribute to your learning (i.e. what did you use the source for)?
Microsoft C# Docs	Understand how to setup C# for development learning the language with official guidance
W3Schools	Learning language basics (i.e. data types, control structures)
YouTube	Visualise difficult concepts (i.e. OOP in C#)

1.5. Application artifacts

A program.cs file is included in the GitHub repository, I apply the concept of print, user input, the string to integer manipulation, one-dimensional array and loop to build this simple program.

The project file contains a main method for executing the program, inside the main method it stores three user inputs into variables num, num2, and num3. an averageOfThree method which takes 3 integer values and inside the method it initialises an array with the 3 integers. then a for loop is used to iterate through the array to calculate the sum of the 3 integer values. Finally, the method will return the summing divided by 3 which is the average of the three numbers.

The main method will then call the averageOfThree method with three user input values.

2. Level B: Basic Application

Whilst level A is about doing something simple with the topic to just show that you have started to be able to use the tool or technology, level B is about doing something practical that might actually be useful.

2.1. Level B Demonstration

This is a short description of the application that you have developed in order to demonstration your understanding. (50-100 words).

2.2. Application artifacts

Include here a description of what you actually created (what does it do? How does it work? How did you create it?). Include any code or other related artefacts that you created (these should also be included in your github repository).

If you do include screengrabs to show what you have done then these should be annotated to explain what it is showing and what the application does.

3. Level C: Deeper Understanding

Level C focuses on showing that you have actually understood the tool or technology at a relatively advanced level. You will need to compare it to alternatives, identifying key strengths and weaknesses, and the areas where this tool is most effective.

3.1. Strengths

What are the key strengths of the item you have learnt? (50-100 words)

3.2. Weaknesses

What are the key weaknesses of the item you have learnt? (50-100 words)

3.3. Usefulness

Describe one scenario under which you believe the topic you have learnt could be useful. (50-100 words)

3.4. Key Question 1

Note: This question is in the table in the 'Self Learning: List of Topics' page on Canvas. (50-100 words)

3.5. Key Question 2

Note: This question is in the table in the 'Self Learning: List of Topics' page on Canvas. (50-100 words)

4. Level D: Evolution of skills

4.1. Level D Demonstration

This is a short description of the application that you have developed. (50-100 words). **IMPORTANT:** *You might wish to submit this as part of an earlier submission in order to obtain feedback as to whether this is likely to be acceptable for level D.*

4.2. Application artifacts

Include here a description of what you actually created (what does it do? How does it work? How did you create it?). Include any code or other related artefacts that you created (these should also be included in your github repository).

If you do include screengrabs to show what you have done then these should be annotated to explain what it is showing and what the application does.

4.3. Alternative tools/technologies

Identify 2 alternative tools/technologies that can be used instead of the one you studied for your topic. (e.g. if your topic was Python, then you might identify Java and Golang)

4.4. Comparative Analysis

Describe situations in which both your topic and each of the identified alternatives would be preferred over the others (100-200 words).