



COS20007

Object-Oriented Programming

Learning Summary Report

Dao Khanh Nga Thi
104177393

Self-Assessment Details

The following checklists provide an overview of my self-assessment for this unit.

Self-Assessment Statement

	Pass (D)	Credit (C)	Distinction (B)	High Distinction (A)
Self-Assessment	✓	✓	✓	

Minimum Pass Checklist

	Included
Learning Summary Report	✓
Test is Complete	✓
C# programs that demonstrate coverage of core concepts	✓
Explanation of OO principles	✓
All Pass Tasks are Complete	✓

Minimum Credit Checklist (in addition to Pass Checklist)

	Included
All Credit Tasks are Complete	✓

Minimum Distinction Checklist (in addition to Credit Checklist)

	Included
Custom program meets Distinction criteria & Interview booked	✓
Design report has UML diagrams and screenshots of program	✓

Minimum Low-Band (80 – 89) High Distinction Checklist (in addition to Distinction Checklist)

	Included
Custom project meets HD requirements	

Minimum High-Band (90 – 100) High Distinction Checklist (in addition to Low-Band High Distinction Checklist)

	Included
Research project meets requirements	

Declaration

I declare that this portfolio is my individual work. I have not copied from any other student's work or from any other source except where due acknowledgment is made explicitly in the text, nor has any part of this submission been written for me by another person.

Signature: **Dao Khanh Nga Thi**

Portfolio Overview

This portfolio includes work that demonstrates that I have achieved all Unit Learning Outcomes for COS20007 Unit Title to a **Distinction** level.

I have fulfilled all the Pass requirements for this unit, including passing the Semester Test and completing the Pass assignments. Additionally, by Week 12 of this Unit, I was able to create a class or function in a number of programming languages, including Python, C#, and Ruby, and I had a solid understanding of the four object-oriented programming principles of polymorphism, abstraction, inheritance, and encapsulation.

Additionally, completing the case study (C tasks) gives me a fundamental understanding of how to design, create, test, and debug programs using object-oriented principles. Additionally, it teaches me how to produce a proper UML class diagram and UML sequence diagram. For additional information, see the SplashKit drawing application or the case study assignments that helped me get adjusted to an integrated development environment.

Last but not least, having finished D tasks enables me to develop my own project, practice researching, designing, and programming, and receive comments from my tutor and friends.

Task Summary

To demonstrate my learning in this unit, I would like the following tasks to be considered part of my portfolio:

a. Pass Tasks:

- 1.1P - Preparing for Object Oriented Programming: **Completed**
- 1.2P - Object Oriented Hello World: **Completed**
- 2.1P - Counter Class: **Completed**
- 2.2P - Drawing Program - A Basic Shape: **Completed**
- 2.3P - Case Study Iteration 1 - Identifiable Object: **Completed**
- 3.1P - Clock Class: **Completed**
- 3.2P - Drawing Program - A Drawing Class: **Completed**
- 3.3P - Case Study - Iteration 2 - Players Items and Inventory: **Completed**
- 4.1P - The Stack and Heap: **Completed**
- 4.2P - Drawing Program - Multiple Shape Kinds: **Completed**
- 4.3P - Case Study - Iteration 3 – Bags: **Completed**
- 5.1P - Case Study - Iteration 4 - Look Command: **Completed**
- 6.1P - Case Study - Iteration 5 - Tying it Together: **Completed**
- 7.1P - Key Object Oriented Concepts: **Completed**
- 11.1P - Clock in Another Language: **Completed**

b. Credit Tasks

- 5.2C - Drawing Program - Saving and Loading: **Completed**
- 7.2C - Case Study - Iteration 6 – Locations: **Completed**
- 9.1C - Case Study - Iteration 7 - Paths **Completed**
- 10.1C - Case Study - Iteration 8 - Command Processor: **Completed**

c. Distinction Tasks

- 6.2D - D Level Custom Program Design: **Completed**
- 6.3D - D Level Custom Program: **Completed**

Reflection

The most important things I learnt:

The most important knowledge after this unit for me is I am get used to a new programming language (C#), and understand about how an OO program was developed. Specially, I was able to apply the OOP principles into my project.

The things that helped me most were:

When I was studying for this subject, the tutor was the most helpful. Even though the lectures were time-limited, I was able to ask my tutor any questions, gain his input, and get further suggestions to improve my program. As a result, I was aware of any errors or misunderstandings and addressed them before coming up with better solutions for my programs.

I found the following topics particularly challenging:

When I initially started using the Testing program and learning how to construct a UML diagram, I found it to be quite difficult. Since the NUnit Test required some installation work, I tried to create a case study that included the test component. Regarding UML diagrams, I had done research and learned how to represent OOP principles or relationships between classes, such as inheritance or association, in a UML class diagram. The UML sequence diagram, however, was trickier for me because I had no prior knowledge of actors, lines, arrows, or actions.

I found the following topics particularly interesting:

Although I initially found NUnit challenging, once I got the hang of it, I actually found it to be pretty interesting. It enables me to test a variety of scenarios that may arise during the course of my program's execution and obtain anticipated results. Additionally, SplashKit increases my interest in programming because it gives me a means to put my creations to use rather of growing bored with a unit with a lot of data and lines of code. By simply writing code, I was able to build graphics, interfaces, and other things.

I feel I learnt these topics, concepts, and/or tools really well:

The topics in which I feel the most confident are SplashKit and UML diagram. It's only that I was allowed to freely design and produce a variety of shapes or figures. Additionally, my understanding of my software is demonstrated in my UML class.

I still need to work on the following areas:

Even though I am only a novice programmer, there are numerous conventions, skills, and knowledge bases that are necessary to become a professional. Additionally, I should work on learning more programming languages because doing so will provide me access to a wider range of problem-solving options and a deeper grasp of how programs operate.

My progress in this unit was ...:

I discovered that I was more motivated than a unit like it. There seem to be more intriguing things than previously that advance my development. As a result, I am submitting assignments more frequently and using tutor comments as a fix for any issues I encounter.

Unfortunately, because my laptop was an older model, it took me a lot longer to complete jobs than it did for everyone else, which occasionally resulted in a late submission and prevented me from finishing the HD tasks in time. As a result, the intended grade is lower than I had anticipated, but that's okay with me.

[This unit will help me in the future:](#)

First and foremost, the knowledge I gained from this unit will be extremely helpful to me in the future, such as in a subsequent unit or job. Since programming languages like C# and Python are currently in demand and modern OOP languages, I will have more resources at my disposal to conduct study, hone my skills, and locate employment in the future.

[If I did this unit again I would do the following things differently:](#)

I'll take extra time to conduct research, design, and development on a more cutting-edge and captivating application for my unique work. Find a better way to configure functions and methods while at the same time creating a program that is both shorter and more efficient. to my own program, then use the knowledge from the references to produce a better project for the custom program.