COS20007

Object Orientated Programming

Learning Summary Report

Henry Nguyen
104249085

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	X			

Minimum Pass Checklist

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

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: **Henry Nguyen**

Portfolio Overview

This portfolio includes work that demonstrates that I have achieve all Unit Learning Outcomes for COS20007 Object Orientated Programming to a **Pass** level.
I am eligible for this grade as I have completed all the pass level tasks to an acceptable standard including the semester test.

Unit Learning Outcome 1 expects that students can explain the OOP principles of abstraction, encapsulation, inheritance and polymorphism. I have demonstrated this in tasks such as the 6.2 Key OOP concepts and the Semester Test section 2 where the questions ask to explain and provide examples of these concepts.

Unit Learning Outcome 2 asks to demonstrate that students can complete tasks or programs in a programming language and associated class libraries. I have demonstrated this through all tasks that I have completed in this unit through the C# language in Visual Studio. These tasks include the Clock program, iteration tasks, shape program which required students to write the programs in C# and Python in Clock in Another Language week 11 task.

Unit Learning Outcome 3 expects students to design, develop, test, and debug programs using OOP principles together with integrated development environment. This has been demonstrated through tasks such as the clock class, iteration tasks as they all require tests to be developed to ensure methods are returning the expected outputs. This is alongside creating the program and checking for errors and tests that have failed and correcting them.

Unit Learning Outcome 4 outlines that student be able to construct appropriate diagrams and textual descriptions to convey the behaviors of an object-orientated solution. This is demonstrated in the clock class, stack and heap, semester test. A component of these tasks required to construct UML diagrams that indicate the features of the program such as their private fields and public methods and properties. The stack and heap task required students to demonstrate their understanding of memory and what part of memory is used in a program through answering questions.

Unit Learning Outcome 5 expects students to describe and explain aspects of good OOP solution reflecting good accepted practices. This has been demonstrated through tasks that ask to explain the principles of OOP such as the week 6 Key OOP concepts task and the semester test as they ask to explain concepts such as abstraction which assist in code design and setting out features of a program and polymorphism and inheritance which supports good design and practices since it prevents code duplication and improves cohesion of code.

Task Summary

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

- 1.1P - Preparing for Object Orientated Programming (signed off as complete)
- 1.2P - Object Orientated Hello World (signed off as complete)
- 2.1P - In Person Check-in 1 – Tools (signed off as complete)
- 2.2P - Counter Class (signed off as complete)
- 2.3P - Drawing Program - A Basic Shape (signed off as complete)
- 2.4P - Case Study Iteration 1 - Identifiable Object (signed off as complete)
- 3.1P - Clock Class (signed off as complete)
- 3.2P - The Stack and Heap (signed off as complete)
- 3.3P - Drawing Program - A Drawing Class (signed off as complete)
- 4.1P - Drawing Program - Multiple Shape Kinds (signed off as complete)
- 4.2P - Case Study - Iteration 2 - Players Items and Inventory (signed off as complete)
- 5.1P - In Person Check-in 2 - Drawing Program (signed off as complete)
- 5.2P - Case Study - Iteration 3 – Bags (signed off as complete)
- 6.1P - Case Study - Iteration 4 - Look Command (signed off as complete)
- 6.2P - Key Object-Oriented Concepts (signed off as complete)
- 7.1P - Case Study - Iteration 5 - Tying it Together (signed off as complete)
- T1 - Semester Test (signed off as complete)
- 9.1P - In Person Check-in 3 - Case Study (signed off as complete)
- 11.1P - Clock in Another Language (signed off as complete)

Reflection

The most important things I learnt:

The most important thing I learn were the 4 OOP principles as they help build and design programs that are understandable and efficient way.

I found that UML diagrams are crucial in understanding what the task it asking for which helped me a lot during this unit.

I found that writing tests were also helpful in finding out if the method or class works as it was intended since sometimes the method may work but return unexpected results when called. Tests help identify these errors by comparing expected outputs to the actual output of the method that is called.

The things that helped me most were:

- Helpdesk: This saved me time as most questions or problems I had were already answered in the chats.
- W3schools and similar resources: Helped me understand C# language and commands
- Lectures: helped me understand the weeks topic and what I can expect when doing the tasks.

I found the following topics particularly challenging:

- Polymorphism and inheritance: these concepts I find are similar to each other and I find that I could get confused explaining 1 concept without referencing the other repeatedly which makes it sound like the other concept.

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

- Abstraction – I understand abstraction well it is building a visual diagram to map out an object's behaviors and collaborations.
- Encapsulation – concept that sets out what parts of program is accessible to other classes or private within its own class.
- UML diagrams – Diagrams that outlines the names, fields, method and properties of a class.

I still need to work on the following areas:

Polymorphism as I am not too confident in explaining this concept. In the feedback of key OOP concepts in week 6 I had to explain it more and also in the semester test section 2 where it asks to reference polymorphism which needed more material, I had to fix and resubmit so it could be marked off as complete.

My progress in this unit was …:

I feel my progression in this unit was good. The first few weeks were confusing as C# was a language that I have not used and trying to learn syntax while completing tasks was challenging but after a few weeks I was more comfortable with C# and the concepts being taught. Submitting work and getting feedback was consistent as I find it can be easy to fall behind since feedback is given on one day out of the week. I feel that I could have balanced this unit and other units better so I could get time to attempt some other credit level tasks for future reference.

This unit will help me in the future:

This unit will assist me furthering my course and units that have other programming languages as it provided me with the concept of OOP that could be applied and give me an understanding of the core concepts. This unit also taught me in searching for assistance through searching websites as learning C# syntax was minimal and researching independently was a big part of this unit for me.

If I did this unit again, I would do the following things differently:

Following the timeline/deadlines outlined allowed me to track my progress and see if I was behind.

I would definitely use the helpdesk as it would save me a lot of time researching on me own in finding answers since other students may have already brought up questions with problems that I was facing at the time.

I would keep on track with the lectures if I were to do the unit again since they were helpful when being introduced to new concepts instead of trying to start tasks without knowing what the weeks topics were.