COS20007

Object Oriented Programming

STUDENT NAME

STUDENT ID

Learning Summary Report

# 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:

**LUAN NGUYEN**

# Portfolio Overview

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

Thanks to COS20007 - Object Oriented Programming - a fundamental course that provide me knowledge of OOP concepts of Abstraction, Encapsulation, Inheritance and Polymorphism. Before I can apply the OOP concepts, my code was complex with functions and variables that I need to use the same format all over again, but thanks to the Inheritance and Encapsulation concepts that help me my code be more readable and much easier for debugging and update. Especially, in Clock Task - the very first Task of the course that gave me an insight into the abstraction concept that uses the UML class to assign roles and responsibilities for properties for functions that I don’t need to know the implement inside but I can understand how to use these properties and methods. Then, in Drawing Class Task, I was familiar with the Splaskit library - which is a very essential tool for visualizing things. After labs, I learnt how to implement the polymorphism concept to implement different kinds of shapes such as Lines, rectangles and Circles. However, I was challenged by installing the Splash Kit library. Thanks to the help from the tutor and the given resources, I overcome it and did well in the class. Moreover, The Swin Adventure Tasks were helped me a lot in understanding the concepts of OOP from the UML class diagram, sequence diagram, and 4 mains concepts that help me put what I learn into practice such as using Inheritance to inherit all the properties from the Items class to the Bag and Player class. During working on the Swin Adventure, Testing and debugging my code by using NUnitTest Framework were also important I will specifically point out in which of the implementation I did it in the wrong way. To say, it helps me easier to fix the code.

And in the very last week of the semester, I m have a chance to apply OOP structure to the Clock Task in Python, which make me the very first step in applying the OOP structure to another language.

I am believe that I am qualified to get the Pass for this unit for my understanding of the OOP concepts by showing what I have done so far

# Task Summary

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

# Pass tasks:

Completed

* 1.1P - Preparing for Object Oriented Programming
* 1.2P - Object Oriented Hello World
* 2.1P - In Person Check-in 1- Tools
* 2.2P - Counter Class
* 2.3P - Drawing Program - A Basic Shape
* 2.4P - Case Study - Iteration 1 - Identifiable Object
* 3.1P - Clock Class
* 3.2P - Stack and Heap
* 3.3P - Drawing Program - A Drawing Class(not sign-off yet but checked by tutor)
* 4.1P - Drawing Program - Multi Shape Kinds(not sign-off yet but checked by tutor)
* 4.2P - Case Study - Iteration 2 - Players Items and Inventory
* 5.1P - In Person Check-in-2 - Drawing Program
* 5.2P - Case Study - Iteration 3 - Bags
* 6.1P - Case Study - Iteration 4 - Look Command
* 6.2P - Key Object Oriented Concept
* 7.1P - Case Study - Iteration 5 - Tying it Together
* 11.1P - Clock in Another Language

Semester Test

Semester Test 1

# Reflection

## The most important things I learnt:

In my opinion, the most important thing that I have learned in the course was the UML class diagram itself was the implementation of the Abstraction concept that show the roles and responsibility of properties and function that we can understand and use and do not need to know further implement inside this. The arrows between the class in the UML class diagram also show the relationship between classes and know whether this class is inherited from other classes.

## The things that helped me most were:

I think the things that help me most was the given resources like slides and video in echo 360 that help me understand the fundamental concepts of OOP and how it is apply through videos. Moreover, feedback from tutors and the convenor is also an important source that help me improve my code.

## I found the following topics particularly challenging:

The UML sequence diagram was particularly difficult for me as I had to outline all the elements and their execution order, including commands and their return values. However, with the help of the materials from topic 3 and the SwinAdventure project, I was able to learn how to create a sequence diagram.

## I found the following topics particularly interesting:

I think the topic that I was interested in is the Clock Task which not only provide me with the fundamental knowledge of the OOP concepts such as UML, Abstraction, Inheritance, Encapsulation, but also I can apply it into another language

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

From my perspective, the four primary concepts, namely Abstraction, Encapsulation, Inheritance, and Polymorphism, significantly minimize the amount of effort required by allowing the reuse of code blocks, functionality, properties, and referencing one class to impact others.

## I still need to work on the following areas:

I think I still need to work a lot more in the OOP concepts that is the UML sequence diagram that I need to have a better understanding of the implementation and the workflow of the code. Moreover, I can apply the OOP concept in any other object-oriented language.

## My progress in this unit was …:

My progress in this unit was not as good as I expected during the semester, it took me quite a lot of time to get the Pass Tasks signed - off. However, I did my best to achieve the final result. Although I was challenged by understanding the basic concepts, I was helped a lot by the tutor and the given resources in Canvas.

## This unit will help me in the future:

I think this unit is extremely helpful for me in the future in that I can apply the OOP structure to my future project to make my code be readable and flexible so that I can save my memory as well as make it easier for me to debug and update code in the future especially is apply them to Python which is a crucial language for my major - Data Science.

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

If I did this unit again, I would do my best to achieve that higher band score.

## Other…:

[ Add any other reflections you think help you demonstrate your learning ]