COS20007

Object Orient Programming

Learning Summary Report

Nguyen Gia Binh
104219428

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: **Nguyen Gia BInh**

Portfolio Overview

This portfolio includes work that demonstrates that I have achieve all Unit Learning Outcomes for COS20007 Object Orient Programming to a **Distinction** level.

I think I should get Distinction because I have finished all the P and C with confirmation that they are complete to standard. Furthermore, I also participate in the task of making a custom program. My custom program is a simplification of the famous board game “Monopoly”, in this program I have include all 4 principle of OOP and the program is running according to plan. I have also made a UML for the program including detail explanation of each class in the program. In addition, I have also passed the semester test with only minor typo in the UML.

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 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 - The Stack and Heap
- 3.3P - Drawing Program - A Drawing Class
- 4.1P - Drawing Program - Multiple Shape Kinds
- 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
- 5.3C - Drawing Program - Saving and Loading
- 6.1P - Case Study - Iteration 4 - Look Command
- 6.2P - Key Object Oriented Concepts
- 7.1P - Case Study - Iteration 5 - Tying it Together
- 7.2C - Case Study - Iteration 6 - Locations
- 6.3D - D Level Custom Program Initial Plan
- 6.4D - D Level Custom Program
- 9.1P - In Person Check-in 3 - Case Study
- 9.2C - Case Study - Iteration 7 - Paths
- 10.1C - Case Study - Iteration 8 - Command Processor
- 11.1P - Clock in Another Language
- T1 - Semester Test

Reflection

The most important things I learnt:

I have learned how to apply OOP principle into my program, understand the basic of C#, learnt how to create Unit test for testing the program

The things that helped me most were:

Unit Test because it allow to understand what is wrong with my program or element that I didn't paid much attention to and is now causing problem.

I found the following topics particularly challenging:

The Pathing of SwinAdventure game was especially challenging for me, most of it is because the instruction doesn't have instruction at all and only include result that the program should achieved.

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

The use of inheritance concept save so much time as a whole, though it can be tricky sometimes to implement all of what before to the derived class

I still need to work on the following areas:

During the creation of the custom program, I got a lot of trouble with enum as program constantly deny to change from 1 type in enum to another even though I have set it

My progress in this unit was …:

Before starting the course, I knew nothing about C# and the way of OOP, I gradually picked up pace after the fourth week as I ask the lecturer more. In future units, I think I could bring what I have learned in C# into it

This unit will help me in the future:

This unit has provided me with basic fundamentals and a sense of object orient programming as I just started out with it. Before if you asked me about my skill, I probably can only code basic stuff and would have to re-code a lot of functions but now I can just make it some sort of polymorphism and save a lot of time without much concern of security as well.

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

If I have to take this unit again then I'd say I would implement all of what knowledge I had learned better as my code is still very lengthy and unnecessary and definitely need some help with it on the first few try