COS20007

OBJECT-ORIENTED PROGRAMMING

Learning Summary Report

Dang Quynh Chi

103805253

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	X

Minimum Distinction Checklist (in addition to Credit Checklist)

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

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

	Included
Custom project meets HD requirements	X

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:

Portfolio Overview

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

Pass Tasks:

I have successfully created classes and objects to model real-world entities, ensuring that the classes encapsulate their internal state and behavior. I've used inheritance to create specialized classes and overridden base class members as needed, allowing for code reuse and extensibility.

Credit Tasks:

In addition to the Pass tasks, I've defined and implemented interfaces to establish a contract for classes to adhere to, enabling polymorphism and loose coupling. I've also designed and used abstract classes to provide a common base for related classes and enforce a structure for derived classes.

Distinction Tasks:

Building on the Credit tasks, I've employed delegates and events to implement the observer pattern or to enable communication between components in my application. I've also utilized generic classes, interfaces, and methods to create reusable, type-safe components in my code.

High Distinction Tasks:

Taking it a step further, I've used reflection to inspect and manipulate types, members, and metadata at runtime, demonstrating a deep understanding of the C# type system. I've implemented 2 design patterns factory and dependency injection to address common software design challenges and improve the maintainability of my code.

By showcasing my work and understanding at each of these levels, I've demonstrated a comprehensive grasp of Object-Oriented Programming in C# and its advanced concepts.

Task Summary

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

- Learning Summary Report
- All Pass and Credit Tasks are marked as Complete
- Tests is marked as Complete
- Portfolio includes work demonstrating good coverage of all learning outcomes
- Custom Program — meets high distinction standards
- Design Report — meets high distinction standards
- High Distinction Project meets high distinction standards

Reflection

The most important things I learnt:

Using Nunit test and using SQL database.

The things that helped me most were:

The teacher's enthusisms, youtube, udemy.

I found the following topics particularly challenging:

Iteration 8 of Swinburne Adventure

I found the following topics particularly interesting:

Save and loading by writer

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

- Nunit test
- Encapsulation

I still need to work on the following areas:

- Code debug
- UI UX design

My progress in this unit was …:

I was able to learn it to the best of my abilities.

This unit will help me in the future:

Be more thorough with my code.

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

Learn SQL database first, leave 3 days for debugging, overall I should allocate more time to debugging.

Other…:

None.