COS20007

Object-Oriented Programming

Learning Summary Report

Nguyen Van Huy Quang
105027350

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 Van Huy Quang**

Portfolio Overview

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

Demonstration of Unit Learning Outcomes

1. **Advanced Object-Oriented Programming Principles:**
    - My project showcases a deep understanding of OOP principles through the thoughtful implementation of program classes. I applied refactoring techniques to enhance code readability, maintainability, and performance, optimizing class designs and employing static methods where appropriate to ensure robust and scalable code architecture.
2. **Effective Use of Design Patterns:**
    - I utilized various design patterns, such as Singleton, Factory, and Observer, to solve complex design problems in my project. These patterns were strategically implemented to ensure flexibility, reusability, and adherence to best practices in software design. This demonstrates my ability to apply theoretical knowledge of design patterns to practical scenarios, improving the overall structure and functionality of the application.
3. **Innovation and Problem-Solving:**
    - My project features innovative solutions for managing complex scenarios, such as Idimons, Player, interactive maps, and managing battle systems. The design prioritizes scalability, efficiency, and user-friendly features, showcasing my ability to think critically and solve problems effectively.

Extending Beyond the Material

- **Architectural Concepts:**
    - I applied knowledge of cloud services and serverless architecture in the project, designing scalable and high-performance applications. This demonstrates my ability to incorporate modern architectural concepts and extend the scope of traditional object-oriented programming to include advanced computing environments.
- **Software Engineering Practices:**
    - The project includes refactored code to improve structure, readability, and maintainability, adhering to high-quality development standards. This reflects my commitment to following best practices in software engineering, ensuring that the codebase remains clean, efficient, and easy to manage.

Conclusion

My portfolio reflects proficiency in Object-Oriented Programming, effective use of external libraries, advanced problem-solving skills, and the ability to extend beyond the standard curriculum requirements. These achievements justify the awarding of a High Distinction grade for my work in this unit.

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 (Completed)
- 1.2P - Object Oriented Hello World (Completed)
- 2.1P - In Person Check-in 1 – Tools (Completed)
- 2.2P - Counter Class (Completed)
- 2.3P - Drawing Program - A Basic Shape (Completed)
- 2.4P - Case Study Iteration 1 - Identifiable Object (Submitted)
- 3.1P - Clock Class (Completed)
- 3.2P - The Stack and Heap (Completed)
- 3.3P - Drawing Program - A Drawing Class (Completed)
- 4.1P - Drawing Program - Multiple Shape Kinds (UML Diagrams updated)(Submitted)
- 4.2P - Case Study - Iteration 2 - Players Items and Inventory (Submitted)
- 5.1P - In Person Check-in 2 - Drawing Program (Completed)
- 5.2P - Case Study - Iteration 3 – Bags (Completed)
- 5.3C – Drawing Program – Saving and Loading (Submitted)
- 6.1P - Case Study - Iteration 4 - Look Command (Submitted)
- 6.2P - Key Object Oriented Concepts (Completed)
- 6.3D - D Level Custom Program Initial Plan (Submitted)
- 6.4D - D Level Custom Program (Submitted)
- 6.5HD - HD Level Custom Program Initial Plan (Submitted)
- 6.6HD - HD Level Custom Program (Submitted)
- 7.1P - Case Study - Iteration 5 - Tying it Together (Submitted)
- 7.2C - Case Study - Iteration 6 – Locations (Submitted)
- 9.1P - In Person Check-in 3 - Case Study (Submitted)
- 9.2C - Case Study - Iteration 7 – Paths (Submitted)
- 9.3HD - Research Project Initial Plan (Submitted)
- 9.4HD - Research Project (Submitted)
- 10.1C - Case Study - Iteration 8 - Command Processor (Submitted)
- T1 - Semester Test (Completed)
- 11.1P - Clock in Another Language (Submitted)

Reflection

The most important things I learned:

The most important things I learned were the practice of Object-Oriented Programming (OOP) concepts and design patterns, which I applied to my custom program. These concepts helped my program become more structured, readable, and effective, ensuring it met the requirements while maintaining a clean and organized codebase.

The things that helped me most were:

The most helpful resources for me were information sources such as GeekForGeeks, YouTube tutorials, and the Canvas modules. These provided valuable guidance and examples that I could refer to while working on my assignments and projects.

I found the following topics particularly challenging:

I found printing code to PDF and drawing UML to be particularly challenging. This task was time-consuming and required careful formatting to ensure the code was presented clearly.

I found the following topics particularly interesting:

I feel I learned and practiced the use of SplashKit very well, especially in working with animations, image scaling, and memory management. These skills were crucial in making my program function smoothly and efficiently.

I still need to work on the following areas:

I still need to work on extending the program and enhancing the user interface to include more functionality. Currently, I feel it is still a bit simple and could benefit from additional features to improve user experience.

My progress in this unit was…:

I made significant progress in getting accustomed to the C# language and OOP coding patterns. I engaged with every task in this unit and tried to perform each task to the best of my ability. This consistent effort helped me grasp the core concepts more deeply.

This unit will help me in the future:

This course taught me advanced OOP concepts, which are widely used in software development. In the future, I will be able to apply these concepts and techniques to more complex projects, enhancing my ability to write clean, maintainable, and scalable code.

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

If I had the chance to study this course again, I would extend my custom program even further and refactor my code more carefully. By doing so, I could ensure that my code is not only functional but also optimized and well-structured for future development.