COS10009 – Introduction to Programming

Learning Summary Report

Show Wai Yan (105293041)

Self-Assessment Details

The following checklists provide an overview of my self-assessment for this unit.

	Pass (D)	Credit (C)	Distinction (B)	High Distinction (A)
Self-Assessment - enter the mark you want under the grade				✓

Self-assessment Statement

	Included (please tick)
Learning Summary Report	✓
Test 1 and Test 2 are Compete in Ed	✓
All Pass level tasks completed (including tutorial tasks)	✓

Minimum Pass Checklist

	Included (please tick)
All Credit Tasks are Complete in Ed	✓

Minimum Credit Checklist, in addition to Pass Checklist

	Included (please tick)
Distinction tasks (other than Custom Program) are Complete	✓
Custom program meets Distinction criteria & Interview booked	✓
Design report has structure chart and screenshots of program	✓

Minimum Distinction Checklist, in addition to Credit Checklist

	Included (please tick)
HD Project included	✓
Custom project meets HD requirements	✓

Minimum High Distinction Checklist, in addition to Distinction Checklist

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 COS10009 Unit Title to a **High Distinction** level.

I believe I have successfully demonstrated all Unit Learning Outcomes at a high level, achieving the distinction tasks with careful attention to detail.

For the Music Player project, I successfully implemented the foundational functionality, including playing, and implementing a basic user-friendly interface. This demonstrates my understanding of basic coding principles and user interface design, as outlined in the unit and my ability to apply the theoretical knowledge gained from the unit into practical, working solutions. While there may be areas for improvement, I ensured all core requirements were met and tested to function correctly.

Furthermore, my code reflects good practices, such as commenting, modularization, and appropriate testing. I have gone beyond the minimal requirements and believe my submission aligns with the expectations for a high distinction grade performance.

Skills and Knowledge Gained from the Unit:

1. **Programming Mindset and Logical Thinking**
2. **Fundamentals of Programming**
    - Learned various data types: Integer, String, Boolean, Float, Array, Enumeration, Record, etc.
    - Learned to create and use functions and procedures.
    - Gained knowledge of parameters, arguments, and variables.
    - Applied design principles: Coupling, Cohesion, Decomposition, Abstraction.
    - Worked with core programming statements: Conditional, Repetition
3. **Advanced Topics in Programming**
    - Implemented game programming and GUI design using Gosu.
    - Developed the ability to research and understand documentation for new libraries.
    - Learned about algorithms, including complexity analysis and recursion.
4. **Testing and Debugging**
    - Gained skills in identifying and resolving errors in programs.
5. **Clean and Consistent Coding Style**
    - Adopted practices to ensure code readability and maintainability.
6. **Code Comprehension**
    - Enhanced ability to read and understand code written by others.

How I Developed Myself

Throughout this journey, I developed myself by learning efficient and active learning methods. I prioritized reading blogs and official documentation to sharpen my understanding, as I found this approach more time-effective and engaging than watching videos. While short videos provided a broad overview, longer ones required significant time and offered limited

opportunities for active learning. By focusing on reading, I was able to delve deeper into topics and retain information more effectively. This is the blog website, Odin Project, that organizes information in step by step way and engages you to learn in learn by doing method.

When solving portfolio tasks, I build a structured approach to problem-solving. I set a 30-minute timer to concentrate fully on the task and, if stuck, took a short break of 10 minutes before returning with a fresh perspective. To resolve challenges, I relied on resources like Stack Overflow and ChatGPT. Instead of immediately asking for code from ChatGPT, I seek and demand explanations and concepts, which allowed me to write my own code first. I then ask code from ChatGPT and compared my implementation with ChatGPT's suggestions to improve my understanding and learn alternative approaches. Additionally, I used Code Wars to practice problem-solving in Ruby, which improves my skills further.

I also learned the importance of clear and strong mind when faced with unclear instructions or complex tasks. I refused to give up, carefully analysed the problem, searched for solutions, and experimented until I succeeded. This experience helped me push beyond my boundaries and gain confidence in my abilities.

One of the most transformative lessons I gained from this unit was a chance to change in my mindset. Previously, I struggled with perfectionism, delaying projects until I felt I had mastered every detail. Through this unit, I realized that as a software engineer, it is more effective to focus on learning what is needed in the moment and applying that knowledge immediately. This process of learning, doing, and relearning allowed me to overcome my hesitation and expand beyond my familiar technology stacks like JavaScript, React.js, and Node.js. I now approach learning with a practical mindset, taking only what is necessary and building on it through experience.

Completed Tasks and What They Taught Me

1. **1.1T, 1.2T**
    - Introduction to Ruby and helped me get familiar with the language.
2. **1.3T**
    - Understanding how to use input and output in programming.
3. **2.1T**
    - Introduction to functions and their applications.
4. **3.1T, 3.2T**
    - Learning to use functions, procedures, and the concept of Records (classes).
5. **4.1T, 5.1T**
    - Reading from and writing to external files.
6. **5.1T**
    - Operating a program using loops and switch statements.
7. **6.1T, 7.2T**
    - Techniques for debugging by hand.
8. **6.3C, 8.2C,**
    - Understanding the Gosu library and learning to apply it effectively.
9. **9.1.1P. 9.1.2C**
    - Combining fundamental programming concepts, complex data types, and file reading/writing.
10. **8.3D, 9.3D, 9.1.3D**
    - Advanced use of the Gosu library to create GUI programs.
11. **8.1T**

- Revision of all fundamental programming concepts.

12. **10.2P, 10.3P, 11.2P, 11.4HD**
    - Introduction to and practice with other programming languages.
13. **7.3P, 9.4HD, 10.1P**
    - Algorithms and applications using arrays and recursion.
14. **12.1C**
    - Testing and debugging techniques.

This list summarizes the tasks I completed and highlights the key knowledge and skills I gained from each.

How did I go beyond the requirements

I went beyond the requirements by completing all tasks within deadlines and overcoming challenges independently without seeking help from the tutor. I successfully passed all tests on the first attempt and actively supported other students by helping them solve problems in the Discord channel. These efforts reflect my dedication to expectations of my grade.

Reflection

The most important things I learnt:

The most important things I learned include developing a programming mindset and logical thinking, mastering fundamental concepts like functions, loops, and data types, and gaining hands-on experience with advanced topics such as GUI design and game programming using Gosu. I also learned the value of perseverance, problem-solving, and debugging, as well as adopting a clean coding style. Most importantly, I shifted my mindset from striving for perfection to practise learning by doing and focusing on what is needed to solve problems effectively.

The things that helped me most were:

The things that helped me most were reading blogs and documentation to deepen my understanding, using resources like Stack Overflow and ChatGPT for guidance, and practicing problem-solving on platforms like Codewars. Setting a timer to focus on tasks and taking breaks when stuck allowed me to stay productive. Additionally, engaging with the Discord community, which is my friends' group to help and learn from peers provided valuable insights and my understanding of the material.

I found the following topics particularly challenging:

One of the most challenging aspects was developing my custom code with Gosu. I worked on creating a tower defence game. Due to Gosu's limitations and lack of built-in features, I had to develop everything from scratch, including animations, object detection, and collision handling. I also animated the user interface and manually translated each UI page, which required attention to detail and significant effort. These tasks were incredibly time-consuming, and leaving me limited time to work on the D and HD tasks because I decide to build this game for production level(not for only assignment) with great code, great design for my future portfolio to apply job. Despite these challenges, the experience greatly enhanced my programming and problem-solving skills.

I found the following topics particularly interesting:

Developing custom code with Gosu, especially while creating a tower defence game, as it allowed me to explore game mechanics and animations. I enjoyed implementing object detection, collision handling, and animating the user interface, as these tasks combined creativity with technical problem-solving. Additionally, working with recursion and exploring algorithms to optimize solutions challenged my thinking and deepened my understanding of programming concepts. These topics were both engaging and rewarding, as they pushed me to apply my knowledge in creative and practical ways.

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

The fundamentals of programming, including functions, loops, and data types, as well as more advanced concepts like recursion and algorithms. I gained strong skills in debugging and testing, ensuring clean and consistent code. Working with the Gosu library helped me understand GUI design, game mechanics, and custom animations, while file reading and writing tasks give me about external data handling. Additionally, practicing problem-solving on platforms like Codewars improved my logical thinking and problem solving skill.

I still need to work on the following areas:

My tower defence game currently has some limitations, including only one tower type, one spell, and one map. These restrictions are not due to coding challenges but rather limitations in art design. Since high-quality art assets often require a financial investment, I had to work with free and limited resources. Improving the visual and artistic aspects of the game would enhance its overall quality and player experience.

This unit will help me in the future:

This unit will help me in the future by providing a strong foundation in programming concepts, problem-solving, and logical thinking, which are essential for any software development role. The skills I gained in debugging, testing, and working with libraries like Gosu will be valuable for building interactive applications and games. Additionally, learning to adapt and work within constraints, such as limited resources for art assets, has taught me creativity. The experience of challenges independently and collaborating with peers will further support my growth as a software engineer.

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

If I did this unit again, I would focus on better time management to allocate more time for the HD custom project and explore additional features for my tower defence game. I would plan ahead to incorporate more diverse tower types, spells, and maps, possibly by seeking affordable or alternative art assets earlier in the process. Additionally, I would dedicate more time to refining the user interface and experimenting with advanced Gosu features to further enhance the game's quality. This approach would allow me to push the boundaries of my project and achieve a more effective and efficient final product.