COS10009: Introduction to Programming

Learning summary report

LUONG TRAC DUC ANH (103488117)

Self-assessment details

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

	Pass	Credit	Distinction	High Distinction (Band 1)	High Distinction (Band 2)
Self-assessment (please tick)				X	

Project task checklist

Check the boxes for each task you have completed.

Pass tasks

☒ 1.1P Hello world

☒ 1.2P Desk check the Bill Total program

☒ 1.3P Hello User

☒ 2.1P Debug coding task

☒ 2.2P Hello user with functions

☒ 2.3P Write your own functions

☒ 3.1.1P Name tester first stage

☒ 3.1.2P Name tester second stage

☒ 3.2P Simple menu program

☒ 4.1P File handling

☒ 4.2P Gosu major cycle

☒ 5.1P Track file handline

☒ 5.2P Music records

☒ 6.1P Array search

☒ 6.2P Album file handling

☒ 7.1P Text music player

☒ Skill tester 1

☒ 8.1P Concept map

☒ 9.1P Fix it

☒ Skill tester 2

☒ 10.1.1P ‘Hello world’ in C

☒ 10.1.2P ‘Hello world’ in Python

☒ 11.1P Python program

Credit tasks

☒ 3.3C Drawing shapes using Gosu

☒ 4.3C Gosu shape moving

☒ 5.3C Hover button

☒ 7.2C GUI music player

☒ 10.2C Recursive factorial

☒ 11.2C Python shape moving

☒ 12.1C Minitest test harness

Distinction tasks

☒ 4.4D Maze creation

☒ 6.3D Custom program design

☒ 7.3D Extended GUI music player

☒ 8.2D Food hunter

☒ 9.2D Custom code

☒ 11.3D C name tester

☒ 11.4D Custom code video

High distinction tasks

☒ 9.3HD Custom code

☒ 10.3HD Maze search

☒ 10.4HD custom project

☒ 11.4HD Custom code video

Portfolio Overview

This portfolio includes work that demonstrates that I have achieve all Unit Learning Outcomes for **COS10009 Introduction to Programming** to a **High Distinction** level.

I have been able to extend beyond the material presented in the unit by:

- Finishing all the tasks with satisfaction of every requirements in those tasks on time (on due date).
- Being collaborative and help out my teammates whenever they are stuck on a task.
- Go in-depth on research tasks.
- Learn to program with Python from YouTube and LinkedIn Learning after being briefly introduced to the language in class sessions.
- Manage to finish both of the Skill Testers within the first try, even though I came across many errors and only got to submit when the time almost ran out.

I have finished all the tasks on Ed and have marked them in the Project Task Checklist section.

Reflection

The most important things I learnt:

The instrumental things that I learn from this course are fundamental concepts of programming, how to set up an environment, how to use IDEs like Sublime Text or Visual Studio Code for text editing, how to run the terminal, how to not only write functional code, but also learn to format it with appropriate commenting to make the code clean, easy to read, and easy to reuse by others – a factor that needs to be taken serious of when working with a team or in my future workspace.

The things that helped me most were:

Video lectures were particularly helpful, as they cover every week's topic in detail, with clear demonstrations and explanations. Coding platforms like Stack Overflow was also useful, as I get to make questions for the bugs that I ran across, see different answers, and get a more comprehensive view on the tasks that I was doing.

I found the following topics particularly challenging:

The most challenging topic to me was file handling, as I did not fully understand the use and application of it initially, so I had to look up different sources, and was lucky enough to came across the tutorial videos of Mr. Mike Dane on YouTube, as his channel cover many topics of many different languages like Ruby, SQL, Java, etc. with comprehensive explanations and practical exercises. The language C that we got to learn in Week 11 was also not easy to get a grasp of, as the syntax is very different to Ruby in comparison to Python. While Python and Ruby are interpreted languages, which means you can run the program from the terminal using *ruby file_name.rb* or *python file_name.py*, in C we have to download a compiler like MinGW to our computers, execute the command *gcc file_name.c* to create an executable file like *a.exe* and execute the file separately, making C very hard to debug without the help of the built-in terminal in Edstem.

I found the following topics particularly interesting:

There are many interesting topics that this unit cover, one of which is GUI – graphical user interface, together with the introduction of libraries like Gosu or Ruby2d, as I find it exciting to make game applications and user interfaces, rather than just do coding and execute the program through the terminal.

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

I believe I am competent at the concept of array, as I can complete the array tasks fairly well in both of the portfolio and Skill Tester 2. The tools that I can use well are IDEs, which I mostly used Visual Studio Code and Sublime Text 4 simultaneously, with Visual Studio Code for big projects like Custom Code and

Sublime Text 4 to handle smaller tasks that only requires me to work on 1 single file.

I still need to work on the following areas:

The areas that I need to improve further are the concepts of coupling and cohesion, which are program designing principles. I believe that I need a better grasp of the application and the practical use of the two. Also, there are not many exercises on Edstem that focus on the areas of coupling and cohesion, as in Week 8, there is only a revision task (8.1) and a Gosu task to customize the original Foodhunter program.

My progress in this unit was …:

In this unit, my initial progress was a bit slow in comparison to other students, as I have little to no prior experience in back-end programming. I did try to learn some Python on YouTube but did not really understand it. However, I still managed to catch up with the lectures in later weeks and finish the course in time with the help of some tutorials from sources outside the school materials.

This unit will help me in the future:

In this unit, , I got exposed to a beginner-friendly language – Ruby, which helped me build the foundation of knowledge and skills for programming in general, understand the different in syntax between languages. Therefore, I could go in depth in other programming languages for a more specialised field. For example, Python for data analysis or facial recognition, which there are numerous libraries for us to use. To go in detail, I wanted to become a cyber security analyst, and Python – a language that was briefly covered in the materials, are often used to make malware for ethical hacking, and that's something that I can make use of in the future career path.

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

[List and explain, how will you approach learning in the future? What things worked well, but what could you change to make sure you did better next time?]

Overall, I believe I finish the course well and can go for the High Distinction level, however, time-management and prior schedule planning for the course will be needed, as there are many tasks which I took to much time to finish, therefore not utilising my time for practicing programming principles as much as I want to. I believe that if I took this unit again, finishing the school work would not be the sole objective, but doing things efficiently.