COS10009 – Introduction to Programming

Learning Summary Report

Ekrar Uddin Mohammed Efaz (103494172)

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 (please tick) |  |  |  | ✔ |

*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 | ✔ |

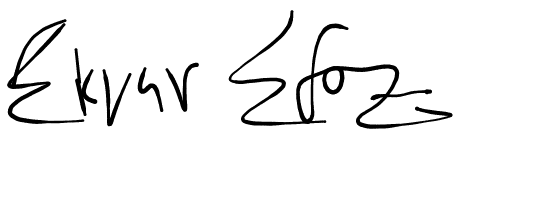
*Minimum Distinction Checklist, in addition to Credit Checklist*

|  |  |
| --- | --- |
|  | 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 think I deserve an HD grade because I have completed all my Portfolio Tasks without severe difficulty. I have made an Extended Music Player as my custom program which certainly goes a bit beyond the complexity this unit provides. I have implemented concepts such as: modular programming, reusable code, good design, recursion and program testing. I have successfully completed my Maze Search Program for HD and attempted a Custom Project which is an extensive literature on “Garbage Collection in Python and Ruby”.

***For Pass:*** I have thoroughly grasped the concepts of structural programming, using conditional branching statements and loops. Successfully completed all the Pass Task without any difficulty. For the first time, 7.1(Text Music Player) seemed be a very difficult task because taking pieces of old code and mashing them together at the same time understanding how each component of code contributed to the completion of the whole program was confusing. Task7.1 was one of the most important building blocks for me to understand how development works and how modular code is an advantage.

***For Credit:*** For the first few weeks Credit tasks introduced us to basic *Gosu* usage and I was able to complete all my credit tasks without difficulty. Right after the pass task Text Music Player I attempted the basic GUI Music Player for credit. I became familiar with GUI design and programming through that. That’s also when I had an interesting idea to make a further extended music player as my custom program. Later weeks were pretty simple regarding the credits tasks. I learned python in week 11 and also made a similar shape moving program in Python using *pygame*. I understood that *pygame* was much smoother and easier to use that *Gosu*. In the final week for credit we did a simple program testing using a ruby gem named *Minitest.* I was confused about the concept at first but later with some Youtube videos and Matt’s lectures understood about program testing.

***For Distinction:*** Distinction tasks were my stepping stones of understanding how to read code and understand the structure of the program. Most of the distinction tasks except the (7.3D)Extended GUI Player were pre-written code I had to understand and implement some additional functionality. First D task was the maze creation problem that made me read and practice the whole “**Learn Game Programming with Ruby**”. Then I was able to solve to the Maze Creation task(4.4D) with a little bit of Matt’s help. After, in week 7 I was excited to work on my 7.3D extended Music Player because I knew I would extend this program and make a custom Music Player. (7.3D) wasn’t much difficult as I had written most of the code myself and didn’t have to read pre-written code.Next D task was in the following week. Task (8.2D) was already a simple game we had to add some extra functions to the game where the food in the game had to change directions randomly and just before it changed direction we had to cover it in smoke for sometime. Implementing wasn’t the biggest obstacle in this task but read ing and understanding a whole program that was written beforehand was difficult. But constant reading of code made me more comfortable in ruby and gosu whereas only writing code wouldn’t take me too far.

***For High Distinction***: Aiming for HD and completing it’s tasks was by far the best experience that helped me grow as a beginner coder. I had to dive head in to my custom program without understanding how I would make a functional game or program with my limited abilities. Talking about Portfolio tasks, I successfully completed my Maze Search Task for HD. In my custom Music Player program, I have imported external libraries and used them to add more functionality to my Program. I have gained the skill to use libraries by reading documentation. I have achieved fluency in Ruby syntax and can implement complex program functionalities. I can use “GOSU” comfortably and even learned to use “RubyTk” to some extent. I used a ruby gem named ‘optparse’ to parse command line instructions when opening my program to add more flexibility to switch between GUI mode and CLI mode. I can handle the concept of recursion very well and am able to make recursive calls comfortably. Although I initially struggle to grasp the concepts of good program design using coupling and cohesion I somehow managed to understand the concept. I have learned to use Python as well and able to display skills in Python on par with Ruby. I also attempted a custom project which is a literature on Garbage Collector. Garbage collection process although is a concept that exceeds the content of this course, I was very intrigued by the process of automatic freeing and allocation of memory in modern programming languages having used C earlier in my high school, where automatic memory allocation and freeing doesn’t happen. Therefore, I made abstract descriptions of garbage collection process and algorithms used in Ruby and Python. My love for programming wouldn’t let me aim for anything less than HD.

# Reflection

## The most important things I learnt:

Modular Program design was one of the best things I learned. Now I feel sick to put all my code in main() function, that’s what I used to do before. I am very comfortable with use of if-else branching statements and loops which help achieve most solutions to any problem. Arrays were also an important data type that helped me a lot in my portfolio tasks. Complex data types such as Classes and Enumerations were a life saver when working with Music Player and my custom program.

## The things that helped me most were:

Early morning lectures are supposed to be dreadful but Matt’s Lectures were the best source for me. Matt releasing next weeks topic on Ed was very helpful because I would read up on what’s coming next week watch a few Youtube videos and attend the lectures with specific goals set which might be some topic I found challenging that needed more attention. I also found a few external resources that helped me learn ruby in this fast-paced course. TheOdinProject has a very good ruby resource.

## I found the following topics particularly challenging:

I found the topic of Good Program Design challenging because I found good program design to be a pretty vague topic. I also struggled to make sense of Coupling and Cohesion. I re-watched Matt’s Lectures numerous time on this which helped clear a lot of aspects but still many things were unclear, so I watched a lot of videos on program design and coupling/cohesion. I watched videos of an Indian guy(channel: Gatesmashers) on Youtube who explained the concepts very well.

## I found the following topics particularly interesting:

Recursion is the topic that peaked my interest. I felt like this is the most interesting piece of code I have seen when I saw recursion. Matt explained recursion very well in his lecture and I attempted both tasks that required Recursion (10.3HD) and (10.2C) in which I was successful 10.3(HD) was the most interesting task in this unit that I solved using recursion and had a blast when I finally traversed the maze without any error.

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

I think I am most comfortable in functional programming and I have learned to use *GOSU* very well. Besides those, I have put more effort into understanding recursion because I found it to be the most interesting. Overall, I have grasped the Ruby and Python syntax very well and I can solve simple problems with my programming skills.

## I still need to work on the following areas:

I still struggle with good program design and the concepts of coupling/cohesion. I feel like I have to put in more work to understand these. I plan to work on researching on these topics in my semester break.

## This unit will help me in the future:

I have learned a lot from this unit. I plan to to work with Application Security in future and programming is one of the core aspects of the role I want to take. I was a little familiar with programming when I started this unit but soon realised this unit is going further in its 4th week that I knew about programming. After that I put a lot of effort into this unit because I loved programming a lot. I plan to further my knowledge *pygame* and *python* and work on interesting projects in my semester break with the skills this unit has equipped me with.

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

If I did this unit I would definitely start working on my Custom Program earlier. When I started my Custom Program in week8 I was already being crushed by deadlines from other units and kept putting it off as my optional work for the day. Thus I ended up rushing my custom program and couldn’t make it more exciting as I wanted to or had planned to in week7. Furthermore I would put more mind into my program design because I think I have missed out on implementing good program design. Otherwise I have no regrets for this unit. I learned a lot.

## Other…:

I learned and got comfortable with few of the most important skills a programmer must have. Firstly I can learn anything and solve any problem by googling and being persistent enough, Secondly, I can sleep on the problems I have been stuck at and come up with a solution next morning. I solved problems that I found impossible at first just because of this. This approach has saved my sanity and I have been successful in completing all my portfolio tasks along with my custom program.