COS10009 – Introduction to Programming

Learning Summary Report

Jake Scott (102581840)

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)	X			

Self-assessment Statement

	Included (please tick)
Learning Summary Report	X
Test 1 and Test 2 are Compete in Doubtfire	X
Ruby programs that demonstrate coverage of core concepts	X
A C program that demonstrates coverage of core concepts	X

Minimum Pass Checklist

	Included (please tick)
All Credit Tasks are Complete on Doubtfire	

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: Jake Scott (4/06/2019)

Portfolio Overview

This portfolio includes work that demonstrates that I have achieve all Unit Learning Outcomes for COS10009 Unit Title to a **Pass** level.

I came into this unit with no programming experience and I originally aimed for a credit grade but due to the lack of understanding on my part, I have decided to aim for a pass grade. I have completed all the **pass** grade tasks as well as a few credit tasks, such as (6.4C, 5.4C).

Throughout my completion of my tasks I have successfully demonstrated all the Unit Learning Outcomes, while technically all the tasks meet these outcomes some demonstrate it better than others. Below are listed:

Apply code reading and debugging techniques:

- 7.1P, this was my greatest challenge in this course as I spent hours going over my code figuring out where I went wrong and how to fix, some were simple like a missing end other were major logic errors within the program.
- Throughout all my tasks I have never completed a program perfectly there has always been errors that I've had to debug in my code

Principles of structured programming:

- 8.1T – A concept map of the principles of structured programming in ruby.
- 7.1P – Using: Sequence, Selection, Iteration and Coupling and explaining where they are present within my music player.

Construct small programs:

- 11.T – A very small program that prints out if a name is silly or not.
- 5.4C – Creating a button within ruby that creates an outline of a box when the mouse is hovering over it.

Use modular and functional decomposition to break problems down functionally:

- 7.1P as there are 5 menu options each menu option has its own related functions and before the program is made, I had to break the problem down into its simplest form in order to create a functioning program.
- 9.1T Breaking down the program into its functions in order to find where the program isn't working properly, by breaking the program down the to its most basic form I was able to fix the program successfully.

Reflection

The most important things I learnt:

Time management skills, that I need to allocate more time to my tasks and in the future of my degree I will plan out what I need to do before I do it.

The things that helped me most were:

- My Peers, since we were all in the same boat together it was important to consult with my peers and seek advice and occasionally give advice.
- My Tutor has helped me enormously throughout this course from simple questions to errors I couldn't solve in my programs, also in the early weeks the way he explained the topics were critical to me understanding more about programming.

I found the following topics particularly challenging:

- Gosu Ruby, because of the lack of resources around it and the odd syntax.

I found the following topics particularly interesting:

- Loops and how critical they are to programming
- Arrays how storing similar data is performed.

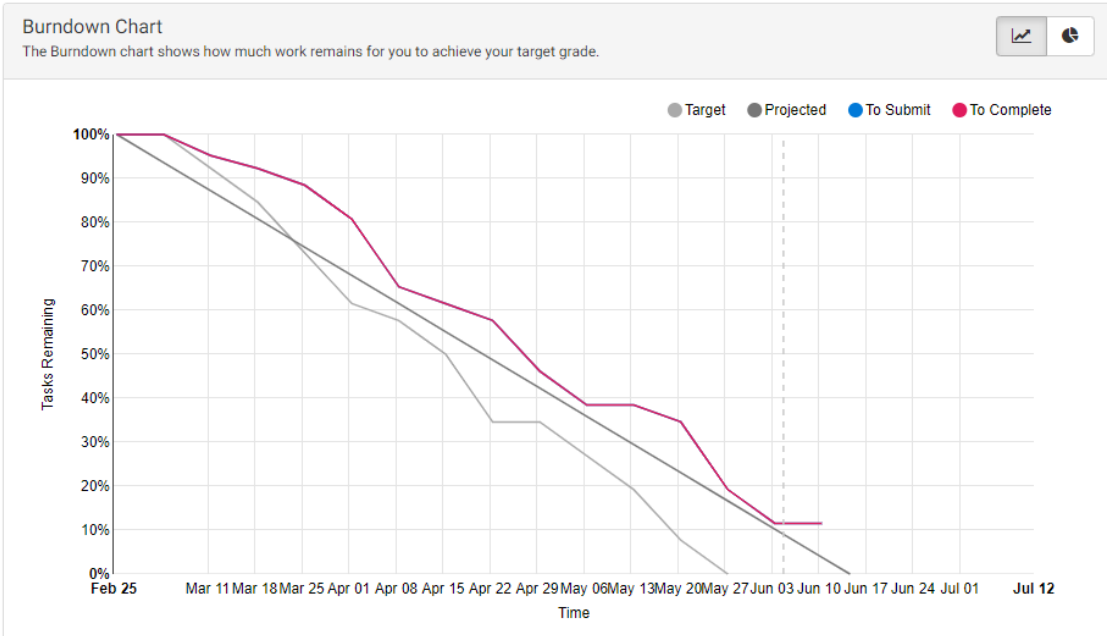
I feel I learnt these topics, concepts, and/or tools well:

- Loops, because of the lecture notes but mainly because of my tutor
- Arrays, looking at the lecture notes I had a bare understanding of them, but my Tudor went into great depth about arrays and how to implement them into my programs correctly.

I still need to work on the following areas:

- Gosu Ruby, I understand the basics of Gosu and how to draw basic objects but when it comes to 7.2C I have a difficult time outputting my tracks as a Gosu image.

My progress in this unit was …:



From the start the system of DoubtFire was new and confusing to me as I've never used it before. The progression of tasks about in week 5 I thought that ever task would be not that complicated as these first five weeks haven't been to difficult. I was thoroughly under prepared for the week 7 tasks, and this led to me quickly falling behind in my tasks.

This unit will help me in the future:

This unit will most definitely help me in the future as I'm studying a degree of Computer Science and next semester, I am taking Technical Software Development as a subject, and I will only further build upon my knowledge that I have learnt in Introduction to Programming.

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

I would have completed every week's tasks on time and how important it is to stay up to date with the course because you can fall behind quickly.