COS20007

OBJECT ORIENTED PROGRAMMING

Nguyen Manh Duc

103792724

Learning Summary Report

# 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 |  |  |  | x |

Self-Assessment Statement

|  |  |
| --- | --- |
|  | Included |
| Learning Summary Report | x |
| Test is Complete in Doubtfire | x |
| C# programs that demonstrate coverage of core concepts | x |
| Explanation of OO principles | x |
| All Pass Tasks are Complete on Doubtfire | x |

Minimum Pass Checklist

|  |  |
| --- | --- |
|  | Included |
| All Credit Tasks are Complete on Doubtfire | x |

Minimum Credit Checklist (in addition to Pass Checklist)

|  |  |
| --- | --- |
|  | Included |
| Distinction tasks (other than Custom Program) are Complete | x |
| Custom program meets Distinction criteria & Interview booked | x |
| Design report has UML diagrams and screenshots of program | x |

Minimum Distinction Checklist (in addition to Credit Checklist)

|  |  |
| --- | --- |
|  | Included |
| HD Project included | x |
| Custom project meets HD requirements | x |

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: **Nguyen Manh Duc**

# 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.

[Provide a justification for why you should receive this grade… Write this for the assessment panel – tell them why you should get this grade.

For Pass: you need to indicate how you have demonstrated all Unit Learning Outcomes to a minimal level.  
For Credit: you need to indicate how you have demonstrated all Unit Learning Outcomes to a good level.  
For Distinction: you need to indicate how you have been able to apply all of the Unit Learning Outcomes in achieving the distinction tasks.  
For High Distinction: you need to indicate how you have been able to extend beyond the material presented in the unit.

In this section, refer to the tasks you have completed. These will be attached by Doubtfire after this summary. Do not try to demonstrate the outcomes here, this is just a summary.

Think of this like a cover letter to a job application – here it is a cover letter to your grade application.]

# Reflection

## The most important things I learnt:

The operation of object-oriented abstraction. I had a rough sense of how programs should abstract away as much complexity as they could in the past, but I wasn't sure how to do it in practice. I like how well this class has explained these ideas to me and demonstrated how to apply them. The most important thing for me was probably being able to read and grasp object-oriented programs correctly; I was no longer misled by the phrase "new" or interfaces.

Observing TDD in action, even though it was at a rudimentary level, was also enjoyable. Additionally, I've heard about unit tests before, but I've never used them and had no clue how they functioned. It was fantastic to be introduced to them.

I also learned that designing and building UML class diagrams and sequence diagrams is highly beneficial in establishing and developing programs that accomplish effective object-oriented design, which includes a few elements such as strong cohesion and weak coupling.

[ Think about topics covered, but also other general things you may have learnt. Think about what you have learnt in this subject and reflect on what you think were key learning points, or incidents. Did you learn what you wanted/expected to learn? ]

## The things that helped me most were:

[ List and explain]

* The lecturer's or tutor's lab demonstrations boosted my knowledge of the content, allowing me to apply it to my programs.
* The input I received from the professor or teacher was helpful in ensuring that I was adhering to excellent practices and in identifying my flaws.
* The internet allows me to look for tutorials for instances of how to tackle a certain problem or activity, as well as examples of how to employ specific syntaxes and make present scripts more efficient.
* The integrated development environments (IDEs) that I use to help me create my programs have made error checking faster and easier. They also provide tooltips for the syntaxes used, which is tremendously useful for debugging and writing faster, cleaner code.
* UML diagrams are useful because they allow me to see the look of my code.

## I found the following topics particularly challenging:

[ List and explain – if none explain why]

Understanding relationships between objects is tough for me since I am unable to constantly relate it to the context or condition provided. This complicates the building of my UML diagram because the objects and their accompanying connections are required. Even the presented reality or scenario may be difficult to comprehend and accept at times.

## I found the following topics particularly interesting:

[ List and explain – remove if none ]

* Because unit testing enables automated testing, it eliminates the need to manually run the software to test certain procedures and evaluate the results. Unit testing has been quite helpful to me since it allows me to validate that the code, I'm writing is functioning as intended without having to repeatedly run my program to test a single line or block of code.

• I am learning C++ because I find it intriguing that a class must be divided into header and cpp files, and that there is a requirement for resource management, which is really challenging and needs a lot of time and practice to fully understand.

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

[ List and explain – if none explain why, refer to your pieces for evidence to support your claims ]

I was certain that I understood the inheritance OOP notion. Understanding and learning about the parent and child courses is beneficial and aids in the development of the program. I now understand how to make objects such that their features may be passed on to one or more other stuff. Because fewer applications must be reproduced, writing the software becomes easier for me.

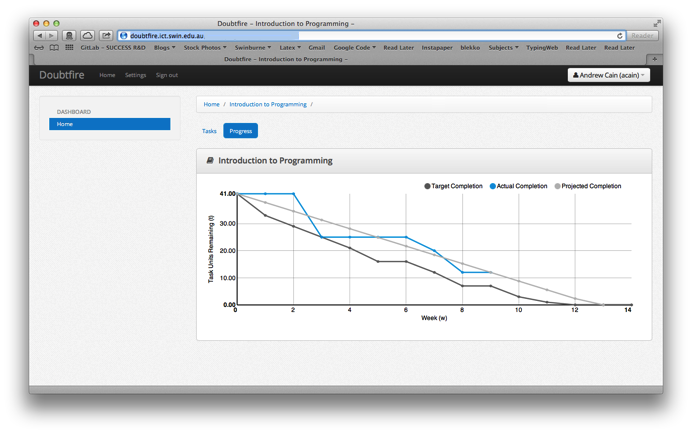
## I still need to work on the following areas:

[ List and explain – if none explain why, refer to your pieces ]

the actual framework of a software The tasks were quite straightforward, but a lot of genuine design effort was required for the succeeding C/D/HD projects. I think all I really need to do to get the hang of it is to write more real programs. My high-level design is tough to produce, even though my low-level reasoning is pretty good.

## My progress in this unit was …:

[ Include a screenshot of your **progress graph** from **Doubtfire**, and comment on what happened from your perspective… what does the graph say about how you approached the unit? (Login to Doubtfire to get your graph <https://doubtfire.ict.swin.edu.au)>]



## This unit will help me in the future:

[ How will the things you learnt relate to the rest of your studies, and career. What have you learnt that will be valuable for you in the future? ]

The object-oriented ideas I acquired in this unit will benefit me in both future job, which will require these principles, and in my future education, where I will be able to apply them easily if necessary. I'll be able to construct an object-oriented design with the knowledge I've received from this course.

## 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?]

I would start performing my chores early and not wait until the last minute. Aside from that, I would make certain that I comprehend everything in the laboratories and lectures, rather than simply copying what the speaker has printed on the projection screen. Following what the speaker performs in the lab and not trying to comprehend what you are doing might result in you failing this unit since you will not know how to execute the activities or tests when they are assigned. In addition, instead of keeping my concerns to myself, I would seek aid from my instructor, and when the time came, I would ask a friend or Google it. That is not a good practice.

## Other…:

[ Add any other reflections you think help you demonstrate your learning ]