COS20007 Object Oriented Program

Learning summary report

John Hewitt
SI-103636725

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	✓			

Self-assessment statement

	Included
Learning summary report	✓
Test is complete in Canvas	✓
C# programs that demonstrate coverage of core concepts	✓
Explanation of OO principles	✓
All Pass Tasks are complete on Canvas	✓

Minimum Pass checklist

	Included
All Credit Tasks are complete on Canvas	

Minimum Credit checklist (in addition to Pass checklist)

	Included
Distinction Tasks (other than Custom Program) are complete	
Custom program meets Distinction criteria & video submitted	
Design report has UML diagrams and screenshots of program	

Minimum Distinction checklist (in addition to Credit checklist)

	Included
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.

John Hewitt
SI-103636725

Portfolio overview

This portfolio includes work that demonstrates that I have achieve all Unit Learning Outcomes for **COS20007 Object Oriented Program** to a **Pass** level.

The work and studies I have achieved in COS 20007 do meet all the criteria contained in the Syllabus to achieve a pass grade. I have developed code and C sharp designs which has given me a good knowledge of an OOP language. In my final report I will be expanding on this understanding and reflecting on the joys and challenges I have had learning C#.

Details of Portfolio

I have uploaded onto Canvas all the code and tasks which show that I have achieved all learning outcomes to a pass level. As well as this learning summary my portfolio uploaded onto Canvas includes the following items:

- Mission 4 Test and answers to questions. This test has also been graded satisfactory. Some of the key skills shown in the test were showing a basic understanding how to add an additional class to an existing program using Inheritance and Polymorphism and associated objects and methods. The test also developed my skills in unit testing in Visual Studio which is a very important part of OOP. As part of this test, I also had to produce my own UML diagram. Often in the Labs we were provided with a UML diagram, to be able to produce our own does make you understand how to plan a program better.

- I produced a document in Task 7.1 providing my perspective of OOP. It is important we at least try to put some of the OOP principles and concepts into words. The report has been graded as satisfactory. While it is hard to put some of the OOP concepts such as Polymorphism, encapsulation, inheritance into words it does help you to understand what they are actually doing in the code, and this is an important part of C# and OOP.

- The three pass tasks required for the Shape Drawing Program were very important for me in developing my C# and OOP skills. It was with this program that I finally got it. Easily being able to alter and add to code is very important and this was finally shown to me with the different stages of the shape drawing program. The Shape drawing program also helped me to get a better understanding of some of the OOP concepts such as Inheritance, abstraction and the benefit gained from OOP libraries. The shape drawing projects I have uploaded to canvas have all been graded as satisfactory.

- My Portfolio includes the Case study which helped me with unit testing as well as other parts of OOP. Initially I did struggle with some of the case study, perhaps because it took me a little while to visualize where it was heading. But now I do think I may even finish some parts of the case study to complete the command driven game console it reminds me of some of the earlier Video games I played.

- The clock timer in C# and the one I produced in JavaScript using classes also showed me how the principles of OOP can be used in many languages.

Reflection

The most important things I learnt:

The most important lesson I learnt was to take the initial effort and time to understand the concepts of classes and objects and also to make sure to follow the information provided in the mission UML diagrams and documents. The missions are designed in a certain order, and it is important to fully understand one before you go onto the next. Since my programming background is procedural or functional, some of the principles of OOP did initially seem strange to me and I thought in some cases making something simple unnecessarily complex. But by the end of the unit and working on larger code and additional libraries I could see how easily the code can be expanded and altered. Of course, the many features of OOP such as inheritance, polymorphism, abstraction, encapsulation, cohesion and etc were all very important concepts of OOP that I have learnt (or should I say learning).

The things that helped me most were:

The lecture videos were very good and helped me understanding OOP. I also in time understood the helpful and important information provided in the lab/mission documents. I realized over time just following the UML diagrams gave you nearly enough information to do all the code. The collaboration videos and other support provided by the ELA was very helpful for me. I must admit due to my initial challenges I did seek some code on the web to help me. This does not make me happy and in the end, I tried to develop more of the code by myself but to be honest I must admit some code sources on the web did help me with some parts of this unit at least initially.

I found the following topics particularly challenging:

I suppose the most challenging part for me initially was the concept of classes and the relationship between them. Once I got that building the methods and properties came easier for me. I wish I challenged myself more in the earlier modules. I realized that even though you may look at a piece of code you may think you understand the individual parts you need to make sure how these classes and objects are connected and working with each other.

I found the following topics particularly interesting:

I did find the whole unit interesting. The unit was presented in a good way and made it a more enjoyable and engaging to learn a language. The mission that I found the most interesting and beneficial for me was the Shape drawing project. It helped me to see the benefits of using OOP principles. From going to a simple rectangle shape to place different shapes on different points enabled me to see the progress made. I was happy I was also able to do a major part of the mission4 unit test. It was a relief to know I knew enough to complete this test.

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

There were many things about this unit which I think I learnt well, and they will be a great help for me. Even I now realize that Visual studio is an important application and it helped me with learning and developing C# and the use of libraries. The unit also helped me develop my skills in unit testing and understanding the importance of these tests especially with OOP. Also, by the end of this unit I think I can now say I understand objects and classes.

I still need to work on the following areas:

Yes, I understand objects and classes, but I am still learning on some of the concepts on how these classes are connected such as Polymorphism, abstraction, encapsulation, inheritance and association as well as other concepts. I am getting there but need to keep working on these concepts. Yes, for example I can prepare code for a class which is inherited from another class, but I need to fully grasp the full power of this concept as well as others, but I am getting there.

This unit will help me in the future:

Understanding an OOP program like C# will help me in the future. Already I have used some of the principles of OOP in another unit understanding PHP and JavaScript. OOP also has made me look at programming from a different perspective to try to visualize a full program before I even build the code. In the past I guess I would start writing the code and then build on it which sometimes can get complicated.

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

I think as I touched on before I would make sure I would fully understand the code as a whole in the earlier modules not just thinking when I look at the individual parts I understand it. Very important to understand those initial concepts.

Other…:

Overall, I have enjoyed this unit and I believe I have done enough to earn a good pass.
I think if I were to attempt this unit again or undertook a similar type of programming unit I would aim for a higher grade. I am pretty confident now in most of the principles and could do most of the tasks required. I know I still would find it difficult in developing my own complex program from scratch but at least now I would be confident in how to approach it.