

COS20007

Object-Oriented Programming

Learning Summary Report

Truong Ngoc Gia Hieu
105565520

Part A: Self-Assessment Details

This session aims to provide a stunning chance for personal reflection on previous tasks and growth during the course. Notably, by enrolling in this fascinating course, I can evaluate my own performance, attitude, gain my strengths, and fields which I need to improve them. Consequently, this learning summary report not only allows me to foster self-awareness but also encourages accountability for strong actions and desired outcomes. In summary, the meaningful report offers me a great opportunity to look back what are my strengths and weaknesses to enhance for my academic performance in next courses and future career.

Appendix I: Completed Tasks and Tutors's Feedbacks

1.1 Week_01 (Task 1.1P and 1.2P)

1.1P - Preparing for OOP

Pass Tasks

16 May by 23:59

9 May at
10:11



Hey Hieu, it is nicely done! However, it would be better if you can attach source code in a code snippet instead of pure text or screenshots in future submissions.

Trung Doan , 23 May at 20:24

1.2P - Personalized Hello

World program

Pass Tasks

16 May by 23:59

10 May at
19:48



28 May at 22:38

It is nicely done! It would be appreciated if you can submit codes as pdf files in future submissions. The file "Creating PDFs of your code files for Task submissions" at the end of assignment submission section shows you how to do that.

- Trung Doan

1.2 Week_02 (Task 2.1P, 2.2P, 2.3P, and 2.4P)

2.1P - *In Person* Check-in 1 -
Checked-in tools and Prior
Experience Survey
Pass Tasks

23 May by 23:59

13 May at
21:00



Check-in task was successfully executed. Student could demonstrate the necessary kit installation.

- Trung Doan

2.2P - Counter Class and
Arithmetic Overflow-checking
Pass Tasks

23 May by 23:59

26 May at
16:23

late



It is nicely done! The report is thorough and well-structured. Keep up the great work!

P+

- Trung Doan

2.3P - Drawing Basic Shapes
with your own attributes
Pass Tasks

23 May by 23:59

22 May at
15:50



It is nicely done! However, the requirement for step 11 is not satisfied. P-

Trung Doan , 7 Jun at 16:23

2.4P - Identifiable Object with
your ID
Pass Tasks

23 May by 23:59

23 May at
19:47



It is nicely done!

- Trung Doan

1.3 Week_03 (Task 3.1P, 3.2P, and 3.3P)

3.1P - Clock Class with your
own hour format
Pass Tasks

30 May by 23:59

27 May at
10:05

Nicely done! However, there are a few issues with this work:

- Clock constructor is missing from UML diagram. And a clock uses 3 counters so please attach this idea in the association line.
- clock must be initialised with a studentID to determine clock format based on last digit.
- clock tests are missing.

P-

3.2P - Stack and Heap and Null
Pointer
Pass Tasks

30 May by 23:59

28 May at
10:16

Nice attempt. However, myCounters[2] is indeed a reference copy instead of an object.

- Trung Doan

3.3P - Drawing Program -
Drawing Class with your own
attributes
Pass Tasks

30 May by 23:59

28 May at
9:20

Nicely done! However, the personalised requirement is not met for MyRectangle constructor. IsAt function in MyLine could be improved by using PointOnLine from SplashKit.

There are no screenshots showing your running program as well.

P-

1.4 Week_04 (Task 4.1P and 4.2P)

4.1P - Drawing Program -
Multiple shape kinds with your
own attributes

6 Jun by 23:59

6 Jun at
10:04



Pass Tasks

Nicely done! However, the personalised requirement is not met for MyRectangle constructor. IsAt function in MyLine could be improved by using PointOnLine from SplashKit.

There are no screenshots showing your running program as well.

P-

4.2P - Case Study - Iteration 2:
Custom Players, Items, and
Inventory

6 Jun by 23:59

6 Jun at
10:14



Pass Tasks

Nicely done! However, Item Test Privilege Escalation is missing. P-

- Trung Doan

1.5 Week_05 (Task 5.1P, 5.2P, and 5.3C)

5.1P - *In Person* Check-in 2 -
Customized Drawing Program

13 Jun by 23:59

11 Jun at
13:37



Pass Tasks

Your answers in the answer sheet are very detailed! The student successfully demonstrated the Drawing Shape program in class and but couldn't answer the follow-up questions. P-

- Trung Doan

5.2P - Case Study -- Iteration
3: Bags
Pass Tasks

13 Jun by 23:59

11 Jun at
13:44



Nicely done! However, TestBagInBagWithPrivilegedItem() and screenshots showing your running test cases are missing. P-

- Trung Doan

5.3C - Drawing Program –
Saving and Loading with
Customized Payload
Credit Tasks

13 Jun by 23:59

11 Jun at
19:30



Nicely done! However, screenshots showing your running program, and TestDrawing.txt file are missing. P-

- Trung Doan

1.6 Week_06 (Task 6.1P, 6.2P, 6.3D, and 6.5HD)

6.1P - Case Study - Iteration 4:
Look Command with
customization
Pass Tasks

20 Jun by 23:59

18 Jun at
13:52



Nicely done!

- Trung Doan

6.2P - Key Object Oriented
Concepts and Self-Reflection
Pass Tasks

20 Jun by 23:59

18 Jun at
13:54



Nicely done! The answers with examples are detailed and on point. However, three appropriate examples where Exception Handling should be used are missing.

P+

6.3D - Custom Program Initial
Plan
Distinction Tasks

27 Jun by 23:59

27 Jun at
10:30



The idea sounds good enough to implement. I can see the usage of Inheritance, Polymorphism and Abstraction in your program.

However, your report is currently missing "Provide supporting arguments and real-world evidence to show that your custom program is practical and significantly useful to its target community and end-users". Please pay attention to this, as you may be asked about it during the interview. Good luck!



6.5HD - HD Level Custom
Program Initial Plan
High Distinction Tasks

27 Jun by 23:59

27 Jun at
10:29



25 Jul at 16:29

The newly added features are complex enough to elevate your program. The updated UML is well done. However, I can't see the usage of the design pattern in this report.

- Trung Doan

1.7 Week_07 (Task 7.1P and 7.2C)

7.1P - Case Study - Iteration 5
- Console Application typing
prior iterations
Pass Tasks

4 Jul by 23:59

2 Jul at
11:41



Nicely done! However, a screenshot showing your program running (by executing the look command) in the terminal is compulsory for this task.
P-

7.2C - Case Study – Advanced
Iteration 6: Locations
Credit Tasks

4 Jul by 23:59

2 Jul at
15:29



Nicely done!

1.8 Week_08 (Test)

T1 - Semester Test
Hurdle Tasks

28 Jun by 19:00

28 Jun at
10:54





The overall work satisfies the test requirement:

- Class FileSystem, Thing, Folder and File are properly implemented.
- The personalised requirement is met.
- The UML diagram is on point. However, zero-to-many relationship couldn't be found. Additionally, inheritance annotation is not correct.
- The answers in task 2 are precise and detailed.

Trung Doan , 7 Jul at 18:55

1.9 Week_09 (Task 9.1P and 9.2C)

9.1P - *In Person* Check-in 3 - Case Study wrap up Pass Tasks	18 Jul by 23:59	7 Jul at 15:45	
9.2C - Case Study - Advanced Iteration 7: Paths Credit Tasks	18 Jul by 23:59	8 Jul at 16:26	

Nicely done!

Appendix II: Summary of Task Corrections

Based on the tutor's feedback on each specific tasks, I revise and update my code which adapt requirements. Therefore, during code revising phase, I frequent compare the previous code with updated code to figure out what funtions, methods, anad attributes that I had missed. However, due to busy learning schedule, I cannot submit the updated code while I have already upgraded them. Notably, tutor's feedbacks allow me to not only know what requirements that I have missed but also greatc chances for me to learn new knowledge/ In summary, I ensure that task submissions from previous weeks up to Week 12 still remains unchanged, highlighting clarity and transparency regading the status of each task and commiment to continuous improvement.

Appendix III: Uploaded Task Submission in PDF Format**Appendix IV: Source Code for Task Submissions in Compressed Zip format**

Based on the descriptions in the template. I decided to press on code files at the beginning of the course until right now into four .zip files including ***Drawing_Shape.zip***, ***Preparation_Hello_Word_Clock.zip***, ***SwinAdventure.zip***, ***Custom_Program_Code.zip***, ***Clock_own_project.zip***, and ***SwinAdventureTest.zip***. Firstly, the '***Drawing_Shape.zip***' file contains all code files for task submission related to Shape Drawing. Secondly, '***Preparation_Hello_Word_Clock.zip***' includes all source code for the Hello World program, The Counter, Clock projects, and Preparation at the beginning of the course. Thirdly, '***SwinAdventure.zip***' holds code implementations in the SwinAdventure game. Then, the '***Custom_Program_Code.zip***' provides code for custom program which aims for D (Distinct) level. Next, the '***SwinAdventureTest.zip***' include all the C# source code (.cs files) and test case implementations in my task submissions. Finally, the '***Clock_own_project.zip***' uses another language (Python) to code.

Self-Assessment Statement

	Pass (D)	Credit (C)	Distinction (B)	High Distinction (A)
Self-Assessment	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Minimum Pass Checklist

	Included
Learning Summary Report	<input checked="" type="checkbox"/>
Test is Complete	<input checked="" type="checkbox"/>
C# programs that demonstrate coverage of core concepts	<input checked="" type="checkbox"/>
Explanation of OO principles	<input checked="" type="checkbox"/>
All Pass Tasks are Complete	<input checked="" type="checkbox"/>

Minimum Credit Checklist (in addition to Pass Checklist)

	Included
All Credit Tasks are Complete	<input checked="" type="checkbox"/>

Minimum Distinction Checklist (in addition to Credit Checklist)

	Included
Custom program meets Distinction criteria & Interview booked	<input checked="" type="checkbox"/>
Design report has UML diagrams and screenshots of program	<input checked="" type="checkbox"/>

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. Failure to meet this requirement will result in a failing grade for the unit.

Failure to provide the source code for any task submission will result in that task not being assessed, even if the task is included in PDF format.

Signature:



Portfolio Overview

This portfolio includes work that demonstrates that I have achieved all Unit Learning Outcomes for COS20007 Unit Title to a **Distinction** level

In this comprehensive unit, I have immersed myself in each lecture to obtain valuable knowledge about the object-oriented programming field, specifically four distinct principles including Encapsulation, Abstraction, Polymorphism, and Inheritance. Furthermore, throughout weekly exercises, I not only have a deep understanding of strong concepts but also the practical applications in the real world. Based on criteria on the Distinction (D) level, I decided to implement these Object-Oriented Programming (OOP) principles into my custom program named Tetris Game. Although the complexity of integrating all four principles is challenging and requires lots of effort and contributions, I believe that this hands-on custom program is suitable for me as a beginner to dive into the object-oriented field which allows me to apply my understanding of these principles to provide clear code, clear UML design, and well-structured.

In order to achieve the Distinction grade, I decide to focus on applying all Unit Learning Outcomes to complete required tasks and integrate all previous four principles in my custom program:

- **Modular Design:** A structured boardgame foundation 20 x 10 (20 rows and 10 columns) and clear UI elements for the game interface.
- **Hierarchical Block System (Inheritance):** In the Tetris Game, there are seven different block types such as Z, J, L, O, S, T, and I which will be seven classes in the program respectively. Notably, these block classes derive from a common base Block class (Block.cs).
- **Friendly UI Design and Core Mechanics:** In the game, there are some specific functions including "Hold", "Next", and a real-time 'Score' display during the game. Furthermore, the "Hold" block will be on the left side of the boardgame while "Next" is displayed on the right side. Finally, the 'Score:' is shown on the top of the boardgame.
- **Game Flow and State Management;** The game starts with dropping blocks from top of the boardgame, then if a row is full, the score will increase precisely based on the updated gamestate. Then, player will play until the block reaches top. After that, the finish screen will display for asking that player want to play again or not.
- **Player Control:** Players control blocks intuitively using keyboard directional inputs for movement and space for rotation, demonstrating robust **Encapsulation** within the block and game state logic.

In conclusion, based on my previous explanations and descriptions, my custom program (project) strongly showcase my application of four principles of OOP and Unit Learning Outcomes to create a complex program, detailed UML Design, and well-structured program that adapt the requirements of Distinction (D) level.

Task Summary

To demonstrate my learning in this unit, I would like the following tasks to be considered part of my portfolio:

- Week_1 (Task 1.1P and 1.2P)
- Week_2 (Task 2.1P, 2.2P, 2.3P and 2.4P)
- Week_3 (Task 3.1P, 3.2P, and 3.3P)
- Week_4 (Task 4.1P and 4.2P)
- Week_5 (Task 5.1P, 5.2P, and 5.3C)
- Week_6 (Task 6.1P, 6.2P, 6.3D, 6.4D, and 6.5HD)
- Week_7 (Task 7.1P and 7.2C)
- Week_8 (Semester Test – T1)
- Week_9 (Task 9.1P and 9.2C)
- Week_10 (Task 10.1C)
- Week-11 (Task 11.1P)
- Custom Program (D level)