



Object Oriented Programming

Topic 5: UML Class and Sequence Diagrams

Resources

The following resources can help you with this topic:

- *Concept map depicting the principles of OOP presented during this week's lecture*
- [Wikipedia article on Concept Maps](#)
- [UML Class Diagram basics](#)

Topic Tasks

Before starting to work on the tasks, first read through this entire document to get a sense of the direction in which you are heading. Complete the following tasks and submit your work to Doubtfire for feedback before the deadline (check Doubtfire for the submission deadline).

Pass Task 13 - Principles of Object Oriented Programming	2
Credit Task 1 - Concept Map	3
Distinction Task 1 - Custom Program UML Class Diagram	4

Remember to submit your progress, even if you haven't finished everything.

After you have **discussed** your work with your tutor and **corrected any issues**, it will be signed off as complete.

Pass Task 13 - Principles of Object Oriented Programming

To make the most of object oriented programs you need to deeply understand the principles of object oriented programming.

For this task you must **explain** the principles of object oriented programming and **relate** these to the programs you have created. Imagine that you are writing the article for a friend who will be studying this unit next semester, and you want to help them get a head start with these principles.

Note: Explain is more than just a description, you want to convey a deeper understanding that can be achieved with a simple description. Try to relate the principles together and express the depth of your understanding.

Tip: A deep explanation does not need to be a long one. Aim to convey your understanding in a concise fashion.

1. Create an article / blog post / letter / comic / movie / podcast / etc that explains the principles of object oriented programming as you understand them.
 - 1.1. Keep it to about 2 pages of text (if text based)
 - 1.2. Videos or podcasts should be kept as short — but still demonstrate depth of understanding.
 - 1.3. Creativity will be valued

Tip: Including references to textbooks (not wikipedia) or research papers can help you strengthen your explanation. Remember to reference other peoples work.

Once you are happy with the report, create a cover page to relate it to the unit's learning outcomes.

Pass Task 13 - Assessment Criteria

Make sure that your task has the following in your submission:

- Explanation should clearly demonstrate a good understanding of the principles of object oriented programming and how they relate to the development of software.
- The report is your work and expresses your understanding in your own words. Where ideas and descriptions are related to other people they are appropriate cited and referenced.

Credit Task 1 - Concept Map

A concept map visually shows the relationships between concepts. This task aims to help you think through the various relationships between the object oriented programming concepts and the associated programming artefacts.

1. Create a concept map, or sequence of concept maps, that elaborate on the relationships between the following concepts and programming artefacts.

Note: You will need to think about this as you do it. Try to capture what you think is important and communicate it to the viewer.

2. Use this to help you prepare to write the article, and include some of the concept maps in the report itself.

Concepts and Principles		Programming Artefacts	
Abstraction	Encapsulation	Class	Method
Inheritance	Polymorphism	Field	Property
		Interface	

Tip: Be creative, box and line concepts maps are ok but if you can make it more visual go for it. We encourage you to use metaphors to communicate ideas.

Once you are happy with the concept map, create a cover page to relate it to the unit's learning outcomes.

Credit Task 1 - Assessment Criteria

Make sure that your task has the following in your submission:

- Concept maps should clearly show a deep consideration of the concepts represented.
- Relationships are explained, not just listed.
- Metaphors help communicate the ideas.

Distinction Task 1 - Custom Program UML Class Diagram

You have now completed tasks that demonstrate the four main principles of object oriented programming, and you are ready to work toward demonstrating these in your own program. If you are aiming for a Distinction or higher grade you should start working on this program now. Aim to create something of around the complexity of the Battleships program. Specifically it should:

1. Demonstrate sound use of the principles of object oriented programming - encapsulation, abstraction, inheritance, and polymorphism
2. Demonstrate the different forms of collaboration between classes
3. Demonstrate the use of collection classes
4. Demonstrate appropriate use of coding conventions (case, indentation, etc.)
5. Include extensive and thoughtful internal documentation

Here are some steps to get you started:

1. Think about what you want the program to do; write a paragraph or two that describes it to others. Drawing a picture of what you want it to look like is also a great idea.
2. Think about the roles and responsibilities of the objects required for your model.

Tip: Focus on the program model (the cake) and don't start working on the icing (e.g. detailed layout of the graphical user interface, game sprite designs, or a comprehensive variety of program features). Start with a small program with a sound design that demonstrates the four principles before writing code.

3. Draw some high-level UML class diagrams by hand - don't worry about every last attribute or method - then scan or take a picture and send to your tutor for feedback. Don't fiddle with the software for UML class diagrams until you are fairly clear about your design. The whole diagram need not fit on one page; e.g. you might have different class diagrams to describe different levels in the inheritance hierarchy.
4. Submit early! The design need not be perfect before you send it to your tutor.

Distinction Task 1 - Assessment Criteria

Include the following in your submission:

- Cover sheet with high-level description of your Distinction project (a couple of paragraphs).
- High-level UML class diagram of the relationships between objects (a single image).