



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*
- [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 through the Blackboard before the deadline.

Credit Task 1 - System Modelling

Distinction Task 1 - Custom Program UML Class Diagram

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

Credit Task 1 – System Modelling

* Compulsory to student aiming for Credit grade

In this task, you are required to develop a UML Class Diagram based on case study given below. You are expected to capture and model all relevant information provided into the class diagram. You may use any Computer-aided software engineering (CASE) tool i.e. MS Visio, [StarUML](#) or [draw.io](#) to produce the UML Class Diagram

Case Study – The Electoral System:

The ICT club in Kuching University holds an election every year to elect one candidate for the positions of President, Vice President, Secretary and Treasurer. This club has around 450 members as to date and the current club president position is held by Johnny. Usually, two Student Council members are appointed as Election Committee to help and carry out the election process. For this year, Christine was appointed as one of the election committee. The ICT club has its own election rules for example, only members who are active in the past 6 months are eligible to vote. Each position must be contested by at least one candidate and up to a maximum of four candidates.

On the actual election day, the voting begins at 10.00AM and closes at 4.00PM. All club members must cast their votes within the time frame given. To cast votes, voters must first register themselves at the voter registration counter and get four ballot papers for all four positions. The election committee will record all required information of a voter such as student ID, name, course, and contact number manually in the voter registration log book. Once the voters' information is recorded, the voter will enter the polling room and cast their votes in the ballot box. Finally, all the ballot papers must be counted by the election committee manually under the supervision of all candidates.

According to Johnny, conducting the election can be very frustrating for the club president, candidates and election committees for several reasons. He is looking for a more effective way in running the election, allowing voters to cast votes online and record all the required information for election in the club database. It would be fantastic if counting ballot papers can be done automatically.

The UML Class Diagram must fulfill the following criteria:

1. All requirements stated in the case study and relevant information are precisely captured.
2. Appropriately demonstrate the use of 4 OOP principles (Abstraction, Encapsulation, Inheritance & Polymorphism)
3. Appropriately demonstrate the use of UML Class Diagram notations

This task aims to help you think through the various relationships between the object oriented programming concepts and the associated programming artefacts. Once you are happy with the artwork, 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:

- Complete UML Class Diagram (Softcopy, not hand-drawn).
- The work is at its best quality as the number of attempt/submission allowed are limited.

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).