Practical Task 8.1

(Credit Task)

Submission deadline: 10:00am Monday, May 25
Discussion deadline: 10:00am Saturday, May 30

General Instructions

In this practical task, you will need to document the design of the final version of the banking system, which you have completed in Task 7.1. Spend some time reflecting on what you have learnt with the focus on the concepts and ideas, how they work, and how they are related in the final program. You need to demonstrate that you have achieved a solid understanding of the OOP concepts implemented in your code.

Your specific task is to create a Universal Modelling Language (UML) class diagram representing the current state of the banking system. Ensure that you place together all the classes that you have developed along with their attributes and methods. You must link the classes identifying the proper relationships between the classes. The existing examples of separate classes provided as part of the previous tasks should guide you in this exercise. We also recommend you to explore Section 7 of the Workbook to refresh your knowledge and learn details of designing using UML diagrams.

To build the required class diagram, use a tool such as LucidChart (you though may choose any software that you prefer). In LucidChart, once you have the diagram finished, you can export and download your work as a PDF document. Remember to keep a copy of the image in your backup as LucidChart is a web-application.

Prepare to discuss the following with your tutor:

- How do you represent classes, fields, methods, and relationships in terms of the Universal Modelling Language?
- Describe the relationship between the diagram and your code.
- How could you use this to think through a solution before you write the code? What would be the advantage to doing this?

Further Notes

- Study the connection between object relationships and class relationships, their differences, and how to implement them in code and show in UML diagrams by reading Section 4.2 of SIT232 Workbook available in CloudDeakin → Resources.
- Read Section 7 of the Workbook to learn about how to interpret UML class diagrams.
- The following links will give you more ideas about class/object relationships, and specifically about composition:
  - https://creately.com/blog/diagrams/class-diagram-relationships/
  - https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-aggregation-vs-composition/
  - https://www.c-sharpcorner.com/UploadFile/ff2f08/association-aggregation-and-composition/
- The following is the link to LucidChart.
  - https://www.lucidchart.com

Marking Process and Discussion

To get your task completed, you must finish the following steps strictly on time.

- Make sure that your program implements the required functionality. It must compile and have no runtime errors. Programs causing compilation or runtime errors will not be accepted as a solution. You need to test your program thoroughly before submission. Think about potential errors where your program might fail.
- Submit the expected code files as an answer to the task via OnTrack submission system. Cloud students must record a short video explaining their work and solution to the task. Upload the video to one of accessible resources, and refer to it for the purpose of marking. You must provide a working link to the video to your marking tutor in OnTrack.
- On-campus students must meet with their marking tutor to demonstrate and discuss their solution in one of the dedicated practical sessions. Be on time with respect to the specified discussion deadline.
- Answer all additional questions that your tutor may ask you. Questions are likely to cover lecture notes, so attending (or watching) lectures should help you with this **compulsory** interview part. Please, come prepared so that the class time is used efficiently and fairly for all students in it. You should start your interview as soon as possible as if your answers are wrong, you may have to pass another interview, still before the deadline. Use available attempts properly.

Note that we will not accept your solution after the submission deadline and will not discuss it after the discussion deadline. If you fail the deadline, you also fail the task and this may impact your performance and your final grade in the unit. Unless extended for all students, the deadlines are strict to guarantee smooth and on-time work throughout the unit.

Remember that this is your responsibility to keep track of your progress in the unit that includes checking which tasks have been marked as completed in the OnTrack system by your marking tutor, and which are still to be finalised. When marking you at the end of the unit, we will solely rely on the records of the OnTrack system and feedback provided by your tutor about your overall progress and quality of your solutions.