COMP1010 Major Assignment

You will write a text-based Java application of your choice which fulfills the requirements set out in the marking criteria. The application must include test cases, classes and objects, ArrayLists, recursive data structure(s), and file input / output.

Assignment guidelines

- You must submit a **single .zip file** containing all the necessary files to run your program and your documentation.
- The project must not have any machine-based or path-based dependencies.
- Your project must not have any external dependencies, except for JUnit. Your project must include the JUnit jar in the *lib* subfolder of the project and be referenced and work correctly.
- The submission must work on lab computers to be marked. You are recommended to test your submission on the lab computers before the due date. If a submission does not work on the lab computers, a mark of 0 will be given. So make sure you test your program on one of the lab machines. Only the latest submission on iLearn is marked, and earlier copies are not saved on iLearn. If you re-submit after the deadline, standard late submission penalty, as per unit guide, will be applied.
- We discourage heavy use of Generative AI to do the assignment, but we do encourage you to seek guidance from it when stuck. That said, we do realise that students do use GenAI extensively. Given that you are making the submission as *your* work, you must be able to understand and explain any code that you submit. We advise that you only use topics covered in the lectures. We cannot be confident that the structures not taught in the unit, and used in the assignment, are your own work or not. Therefore, make sure we have covered the structures you are using. You may be asked to attend an in-person interview based on your submission if there are any suspicions that a satisfactory level of understanding has not been achieved. If you are asked to attend an in-person interview, your marks will be finalized only after the interview is held. If you choose to not attend the interview, your mark will be finalised as a 0.

Marking criteria

Your submission will be based on the following criteria:

- Functionality (3 marks)
    - What does your program achieve, and what can it be used for. The extent of the services / features provided by your program. Think about the quality of features, not just the quantity. Your program must include a Client / Driver class that demonstrates the capability of your application.
- Scope (5 marks)
    - (1 mark) At least one class should contain object(s) of another user-defined class in the project (but not an ArrayList of objects)
    - (1 mark) At least one class should contain an ArrayList of object(s) of another user-defined class in the project
    - (2 marks) At least one satisfactory instance of recursive data structure being used.
    - (1 mark) File I/O. Example provided with the specs and demonstrated in week 8 lecture (**studyPlanner.zip**).
- Unit testing (2 marks)
    - Whether you wrote good-quality (a diverse variety of data including some edge or boundary cases) unit tests for your classes or not.
    - You don't need to write tests for every single method. Quality and diverse tests for five non-trivial functions is deemed as sufficient coverage
- Code quality (5 marks) Refer to the code style guide
    - (1 mark) Commenting
    - (1 mark) Indentation
    - (1 mark) Variable names
    - (1 mark) Delegation
    - (3 marks) Class Design
- Documentation (3 marks)
    - Provide a README.md file of somewhere between 200 and 500 words as part of the zip file you submit that includes the following clearly labelled sections:
        - What problems does your application solve?
        - A description of the structure of your program.
        - Clear instructions on how to run your application in bullet points

Submission Timeline

- Assignment released: Sunday 23:55pm Week 7
- Checkpoint submission: Sunday 23:55pm mid semester break week 1
    - Online via the “Major Assignment checkpoint submission” box
    - You can submit your current progress on your project to receive feedback.
    - You should submit a zip file which contains your project code and documents required for the submission.
    - Your submission must compile for us to provide feedback.
- Consultations: in-person consultations will be provided in weeks 10 and 11.
    - Sign up using the session selectors on iLearn
    - You can sign up for one consultation session per week.
    - If you do not show up to a session you have registered for, you may be banned and/or removed from future sessions.
- Final submission: Sunday 23:55pm Week 11
    - Online via the “Major Assignment” iLearn submission box
    - Submit a single zip file containing all your code and documentation
    - Refer to the unit guide regarding late penalty rules
    - Grade based on rubric provided
    - Sample solutions are provided and discussed in the week 8 lecture.

Guidance and support available

This assignment covers the following topics:

- Problem solving
- ArrayLists
- Classes and objects
- Recursive data structures
- Unit testing

The lectures that cover these topics are from weeks 1 to weeks 10. While the drop-in centre staff will not directly help with your assignment, you can go to the drop-in centre for help with all the lecture, practical, and practice content. These directly apply to the assignment and give you the knowledge required to solve the assignment.

You can find the computing drop-in centre schedule here: https://students.mq.edu.au/study/faculties/science-and-engineering/drop-in-centre

Checkpoint submissions and consultations are also available for the assignment. See the above submission timeline for the dates that these are available.

Project ideas

Some past project ideas are listed below. Note that you are not allowed to directly implement these ideas. You can come up with your own idea using these as a starting point, or you can come up with your own completely new idea.

MQTube

A video streaming service for Macquarie students to use. This site will allow courses, university staff, and students to create educational videos which can be viewed by others. The platform includes channels which can have videos and playlists of videos.

MQGram

A social network aimed to help students share study tips and talk about how great university is. Students will be able to make posts, which can have comments from other students. Posts can be purely text based or include media. Students can follow other students that they would like updates from.

Masters of MQ

A typical turn-based RPG involves two teams, each consisting of one or more characters, fighting until one team has had all their characters reach 0 "health points". On their turn, characters can perform two or more actions, such as attacking or defending. Often, characters cannot take actions if they do not have any health points left.

Information on the outcome of an action is delivered to the player via an easy-to-read, sometimes exciting string, such as "Aang attacked Deadpool! Deadpool took 5 points of damage!"

Characters have a few base "stats" such as strength, intelligence and defence that dictate the outcome of an action. For instance, if character A attacks character B, compare A's strength against B's defence. “Dice rolls” or other randomness is often factored into an outcome to give the game some unpredictability and excitement. E.g., a number between 0-5 is added to attack damage, or a dice roll determines if the attack hits or not.