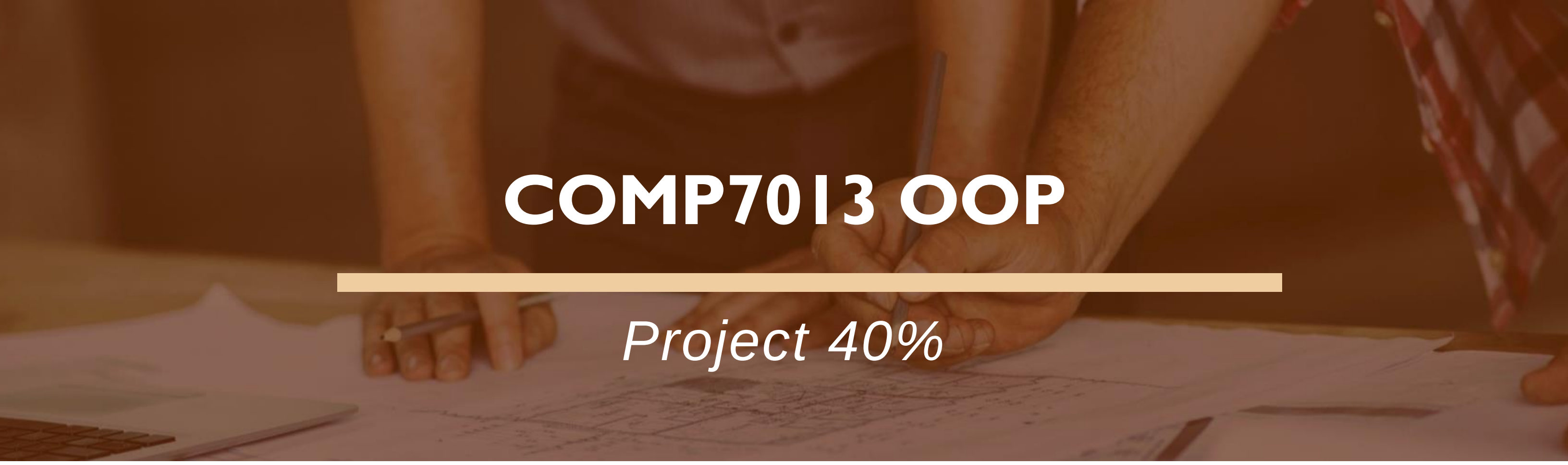

COMP7013 OOP

Project 40%

Submission details

This project is due on the 6th of May 2022 at 23:59.

Submissions after that date will incur late penalties

- 10% deduction for submissions less than 1 week late
- 20% deduction for submissions less than 2 weeks late
- No submissions will be accepted after the 20th May

Submissions must be uploaded to Canvas via the associated Canvas Assignment. Please include all code, files, and screen recordings in a zip file.

My advice to you is to utilize the code that you created in your Practical Assessments as a starting point for your Project.

Please ensure that the code you submit is clean and well documented code. If classes, methods, or attributes have become obsolete in your updated project then they should be removed.

Project outline – student records GUI

I want you to develop an application that stores student records at MTU; the records consist of information about each student and information about the modules they have completed and the grades they received. This will allow the university to stay informed about student performance.

You will use a database (**not an ArrayList**) to store Student objects.

Write a JavaFX desktop application with a GUI (**do not use Scene Builder**).

Write the classes required to implement the above functionality and use them in an application that has JavaFX GUI as the front-end and a database as the back-end. You can create any tables you want to make the application function properly.

Project requirements

Your application should allow users to:

1. Use the GUI to add a new Student to the database (each student should have Name, ID, and DateOfBirth attributes)
2. Use the GUI to remove an existing Student from the database
3. Select a student in the database and edit their details (Name, ID, or DateOfBirth)
4. Update a Student's record to show the modules they completed and the grades they attained in each module
5. If a Student is removed from the database their module information should also be removed
6. Search for a particular student in the database and display that student's information and their module results
7. Display all of the modules where a student attained a first-class honor (grade equal to or greater than 70%) – based on a database query
8. Display all of the students in the system
9. Quit

Additional project requirements

1. Use objects in the application, not just Strings. The methods of the class which connects to the database should take objects as parameters, where appropriate.
   - ❖ Instead of having Name be a String attribute for each Student, create a Name class which has FirstName, MiddleInitial, and LastName attributes.
   - ❖ Have DateOfBirth be a specialized Java Object instead of a String.
2. Use the MVC pattern for this application and use a package structure to reflect the MVC pattern.
3. Write Junit test cases (completely covering at least 3 classes) and a test-suite to test some elements of your code in your project.
4. A good package structure; you should consider how your different classes can be separated into packages to improve code readability. Consider how this will help your MVC design pattern structure.

5. Document your project using JavaDoc, generate the JavaDoc documentation.
6. Add an extra button that creates a loop that creates dummy people objects and adds them to some collection (stored in the Heap Space) until the application runs out of memory. Note how long it took and the memory at the point of exception. Set the VM size to half of normal then use the same button and observe what happens/how long it takes to get the out of memory exception. Attach a Word document which shows your analysis of this memory leak and explain in your own words what is happening.
7. 15% of the marks for this assignment will be reserved for students who use the JPA API to interface with the database. This will require you to research, install, and utilize a Java API yourself. Reference links: Oracle, TutorialsPoint