**Assessment Details**

**Functional**

Build a Java console app that manages student enrolment

1. Student class: id, name, birthdate

2. Course class: id, name, number of credits

3. StudentEnrolment class that has 3 properties: student , course, semester (2021A, 2021B etc. This is a string)

This is an associate class. To keep all enrolments, there must be a list that keeps all enrolments.

You are supposed to define an interface StudentEnrolmentManager to take care of this list. Refer to the lectures on class relationship and interface.

This interface must have the following methods: add, update, delete, getOne, getAll.

4. Build functionality to enroll students in courses.

This functionality will basically **add/update/delete** into StudentEnrolment list.

There will be an academic assistant who will use this functionality to enroll students in courses.

**Enroll a student for 1 semester**

The system should ask id of students, semester, and courses that students need to enroll (use Scanner(System.in))

**Update an enrolment of a student for 1 semester**

The system should list all courses of a student in a semester and ask whether delete or add new courses from the list.

5. Print all courses for 1 student in 1 semester. Print all students of 1 course in 1 semester. Prints all courses offered in 1 semester.  
Allow to save these reports to CSV files.

6. When starting the program, sample students and courses should be populated. There is no need to build CRUD functionality for student and course.

   Default file: [default.csv](https://rmit.instructure.com/courses/86345/files/17561723?wrap=1)

7. Write unit tests for all methods used in this program and ensure at least 100% code coverage.

**Write a report**

1. Students need to write a report 3-4 pages to describe their program, following best practice on software engineering.  The report should include a class diagram, use case diagram and other diagrams that students think they are important.

**Github code repo**

1. Students during their project must use Github to maintain their project source code.

2. Students must have at least 10 commits to their Github.

**Nonfunctional**

1. Proper use of naming convention for class, members, local vars, packages

2. Good organization of package, class, and methods

3. Code quality reduces numbers of critical, proper use of try/catch, if-else to prevent error.