Tut-Lab Week 4

Aims:

- Understand the object-oriented design process
- Learn to evaluate different system designs
- Gain familiarity with tools for object-oriented design

Preparation:

- Review the material on CRC cards and UML class and sequence diagrams
- Review the API documentation for library classes: **ArrayList<E>, LinkedList<E>, Iterator<E>**

Object-Oriented Design

- Consider a university enrolments system with the following requirements
    - Students enrol in courses that are offered in particular semesters
    - Students receive grades (pass, fail, etc.) for courses in particular semesters
    - Courses may have prerequisites (other courses) and must have credit point values
    - For a student to enrol in a course, s/he must have passed all prerequisite courses
    - Course offerings are broken down into multiple sessions (lectures, tutorials and labs)
    - Sessions in a course offering for a particular semester have an allocated room and timeslot
    - If a student enrols in a course, s/he must also enrol in some sessions of that course
- Design an object-oriented system to implement the above requirements
    - Define one or more use cases, e.g. for a student enrolling in a course that has a prerequisite that s/he has passed
    - Provide CRC cards for your main classes
    - Draw a UML class diagram for your initial design
    - Clearly explain how your design distinguishes between a course and an instance of a course in a particular semester
    - Identify any assumptions and design tradeoffs you have made
- Consider the above use case for a student enrolling in a course that has a prerequisite that s/he has passed

    - Describe using a walkthrough how this use case would be handled in your system
    - Define a UML sequence diagram corresponding to this walkthrough

- (Partially) implement your design using Java, **making sure that your code conforms to your design**
    - If you did your design properly, you should not have to change the design once you have started coding