Module title: Object Oriented Programming in Java and UML
Name of the assignment setter: Dr. Sardar Jaf
Submission deadline: 14:00; 25/11/2016.
Method of submission: DUO.

Scenario

The assignment for this module is to implement a student record system in Java. The system should include Java classes for students, programmes, and modules.

Tasks

- **Students:** Define and implement Java class to do the followings: **(10%)**
  1. A way to create a new student record with the following details:
     - ID, name, gender, date of birth.
  2. Different methods to set the following student details:
     - email, postal address, and finish year.
  3. Different methods to return the following student details:
     - name, gender, date of birth, email, start year, and finish year.
  4. A method to return all the male/female students that are older/younger than a given age.
  5. A method to return all the male/female students living in a given postcode.
  6. The class should include error checking.
- **Modules:** Define and implement a Java class to do the followings: **(5%)**
  1. A way to create a new module.
  2. A methods to return a module name.
  3. A method to change a module name.
  4. A method to add a student to a module.
  5. A method to remove a student from a module.
  6. A method to count the number of students added to a given module.
  7. A method to return all the modules for a given student.
  8. The class should include error checking.
- **Programmes:** Define and implement a Java class to do the followings: **(10%)**
  1. A way to create a new programme.
  2. Methods to get and set a programme name.
  3. A method to add a module to a programme.
  4. A method to remove a module from a programme.
  5. A method to return the modules that are shared between two programmes.
  6. A method to return the yearly tuition fee for a particular programme.
  7. A method to set the yearly tuition fee for a particular programme.
  8. A method to show all the modules with the number of students studying each module.
  9. A method to return a programme that has the highest number of modules on it.
  10. A method to return a list of modules with all students added to each module.
  11. The class should include error checking.

- **Extras:** Extend your program in one or more of the following ways: **(15%)**
  - Allow a user to enter and update the details of one or more of the followings via a textual or graphical user interface: students, modules, programs.
  - Allow details of students, programmes, and modules to be saved to a file and loaded from a file so that a user may edit and use the records over many sessions.
  - Allow details of students, programmes, and modules to be saved in a database and loaded from a database.
  - Create a login page for administrators to use the programme. Administator must be authenticated and successfully logged before they can use the system.

Assessment weightings

- Basic Programming Skills: the main emphasis is on proficiency in Java programming, including definition of classes, use of methods and constructors, inheritance, exceptions, input and output.
- Use of Java APIs: including collections.
- Design: including use of appropriate UML notation.

Submission mode

- Project Implementation: you will electronically submit your working implementation plus instructions for running it. Include all Java source codes and classes for Students.java, Modules.java, Programmes.java, and any other classes you have created for your system. **(45%)**
- Project Report: you will electronically submit a commentary on your implementation, written in HTML, with relevant links to a javadoc description of the classes you have written. Relevant UML diagrams, which may be generated from BlueJ, should be included. Any discussion of the performance of your code should be included here. The report and the implementation should be submitted by 14.00 on Friday 25th November. **(10%)**
- Bench Test: a short written test of basic Java skills and API material will be set at 10.00am on Monday 21st November. It will be a closed-book test which will last two hours. Short structured answers will be required. **(45%)**

Assessment weightings

Assessing	**Implementation**	**Report**	**Bench Test**	**Total**
Java programming Skills	40%		40%	80%
Use of Java APIs	5%		5%	10%
Design		10%		10%
Total	45%	10%	45%	100%