# Java OOP 2016: Assignment Brief

## Scenario

You’ve just made a contract with VGU Academy International to develop its Student Management System.

In this Student Management System, you are required to create an application to store list of students and list of lecturers. Following information are to be stored for each student:

- stdId: The student ID of the form like GTxxxxx or GCxxxxx (x: is a digit)

- stdName: Student name

- stdDoB: Student date of birth

- stdEmail: Student email

- stdPhone: Student phone number

- stdAddress: Student address

- stdBatch: The batch (class) of the student

Following information are to be stored for each lecturer:

- lecId: Lecturer ID with 8 digits (fixed)

- lecName: Lecturer

- lecDoB: Lecturer date of birth

- lecEmail: Lecturer email

- lecPhone: Lecturer phone number

- lecAddress: Lecturer address

- lecDept: Lecturer department (e.g., Computing, Business, etc)

This application will need to provide following functionalities via a menu

=======================

1. Manage Students
2. Manage Lecturers
3. Exit

=======================

Please choose:

When user selects 3, the program will exit.

When user selects 1, the program will display submenu for managing students:

=======================

1. Add new student
2. View all students
3. Search students
4. Delete students
5. Update student
6. Back to main menu

=======================

Please choose:

When user selects 2, the program will display submenu for managing lecturers:

=======================

1. Add new lecturer
2. View all lecturers
3. Search lecturers
4. Delete lecturers
5. Update lecturer
6. Back to main menu

=======================

Please choose:

For the submenu:

When user chooses 1, program will prompt user to input student’s/lecturer’s information (specified previously). After that, program will validate the input data and if they are all valid, program will add a new student/lecturer to the current list of students/lecturers. Program should inform to the user corresponding messages.

When user chooses 2, the program will list all the students/lecturers to the screen, each student/lecturer in a row and student’s/lecturer’s data fields are separated by ‘|’.

When user chooses 3, the program will ask user to input student’s/lecturer’s name to search for, the user can just type part of the name in order to search for complete student/lecturer information.

When user chooses 4, program will ask user to input student/lecturer id to delete the student/lecturer with the specified id if it exists, otherwise, it will display a message to inform users that the student/lecturer with such id doesn’t exist.

When user chooses 5, program will first ask user to input student/lecturer id to update, once inserted and a student/lecturer with the inserted id exists, it will display current data for each field of the student/lecturer and user can type in new data to update or just press enter to keep the current data for the field.

When user chooses 6, program will back to the main menu.

## Tasks

### Task 1 – Problem Analysis

For the scenario above, you need to produce a design for the requirements given. The design must include

* Class diagrams for all objects identified as well as class relationships.
* Pseudo-code or flowchart for the Algorithms for the main functionalities (listed in the scenario e.g., add/view/search, etc).

Besides these you need to state what features have been included/excluded and why certain features have been used over others (where applicable). Example code can be used to help clarify OOP features.

### Task 2 – Implementation

In this task you will need to code the application and add sufficient sample data for following tasks. The data have to be stored persistently and submitted together with the solution.

### Task 3 – Presentation

Please prepare a presentation (about 15 minutes) with the following points

* Coded UML Class diagram and explanation about Relationships among classes.
* How did you implement main functionalities (Add, update, delete, etc.) with main flow and code snippets.
* Application demonstration (slides, screencast or live)

Explain what IDE used and how using the IDE has assisted the development process. E.g. debugging, intellisense, code snippets, etc.

### Task 4 – Software Testing

* Produce a test plan (strategy + test cases) and execute it. Your test cases must include:
* Normal scenarios
* Data validation.
* Extremes of data.
* You will need to critically review the actual outcomes of the tests against your expected outcomes; were your tests adequate and appropriate; did they cover all user requirements.

## Deliverables

1. A **Report** containing all the artefacts produced for Task 1 and 4 (eg Class diagrams, pseudo-code or flowchart, test log, etc.). The report should include your student detail as well as short introduction about yourself. This should be well organized using Headings, Tables, TOC, etc.
2. A **Runnable Solution** for Task 2. The solution should include sufficient sample data.
3. A **Presentation** and **Screencast** for Task 3. The presentation should be pitched at a non-technical audience to try to persuade them to purchase the product; the screencast should demonstrate the functionality of the system.

## Assessment Breakdown

Functionality 40%  
Expect: Student and lecturer management, store and load data, user interface and error handling, class diagram, OOP features, pseudo-code/flowchart, code snippets.

Testing 25%  
Expect: Test plan, test log, sufficient data to fully test, evidence of testing finding errors, critical evalutaion.

Screencast and Presentation 25%  
Expect: Professional standard of presentation promoting the product, screencast demonstrating all the main features of the product.

Quality of documentation 10%

Expect: NO SHARED CONTENT, professional standard, header page, page numbers, table of contents, headings, cropped images, figure captions, no spelling or grammatical errors.