

## CSIT121/821 Lab Exercises

### Lab 4

**Deadline: 11 October 2020, 10 PM**

#### Objectives

- Familiar with Java exception handling process.

#### Task:

In Assignment 1, the user was required to input some personal information to initialise the course enrolment process and select the major and elective subjects to complete the enrolment. However, the requirement of A1 (please check Moodle for A1's requirement and solution) does not consider the exception handling. It is possible that the user inputs invalid information during the enrolment process. In this task, you are asked to design and implement the exception handling solutions on top of A1's solution (you can use your own solution or the solution provided by me). Please carefully consider the possible invalid information the user may input during the enrolment process, and design and implement relevant solutions to handle those possible exceptions.

It is required that the updated program shall be robust to any possible inputs provided by the user and shall not have the runtime error. The exception handling process can detect the user's invalid inputs and provide the chance to allow the user to re-input the information until it is valid. The possible (but not limited) invalid inputs could be:

1. No inputs (the user just press the Return button);
2. Invalid DOB format (correct format shall be dd/mm/yyyy. All other formats such as dd-mm-yyyy, mm,dd, yyyy and ect are invalid);
3. Invalid DOB (the date does not exist, such as 29/02/1999, 31/04/2000, and etc);
4. Invalid major index (input 0 or 6 for BCS major selection);
5. Invalid elective subject code (input not existing subject code such as CSIT191), and
6. Others (optional, you can do other validation checking based on your own assumptions)

You may need to modify A1's solution (fields and methods) to complete this task based on the following requirements.

- You can use the build-in exceptions predefined by Java (Exception and/or sub-class of Exception). However, your program must contain at least one custom exception defined by yourself via extending the Exception class.
- For each exception handling, you may use the "throw", "try...catch", "finally" and "Assertion" blocks.
- Your program must contain at least one multiple "catch" blocks.
- Your program must contain at least one "finally" block.
- Your program must contain at least one "Assertion" block. (note: assertions are not enabled by default. In order to execute the programs contain assertion, you shall run your program as `"java -ea ClassName"`).
- Please test the updated program with some possible invalid inputs listed above (Items 1~5 are essential, and Item 6 is optional).

#### **Submission:**

- Please submit your solution to Moodle. Email submission is not accepted.
- Please submit an **individual PDF document** to contains all your solutions for the task.
- In the pdf document, please include your updated java program and the snapshots which clearly shows the compilation and the execution of your program.
- The execution of your program shall contain the program tests with invalid inputs listed above (Items 1~5 are essential, and Item 6 is optional).

Note: Turnitin will be used to check similar submissions. Plagiarism will be reported to the school, and all involving students will receive the zero mark or other penalties based on the university's regulation.