SCIT
School of Computing and Information Technology

Spring 2020
CSIT121/821 — Programming Fundamentals

Assignment 3 (10 marks)

Due Time and Date:

Due by Sunday 8ᵗʰ November 2020 10 pm

General Requirements:

- You should create your programs with good programming style and form using proper blank spaces, indentation and braces to make your code easy to read and understand;
- You should create identifiers with sensible names;
- You should make comments to describe your code segments where they are necessary for readers to understand what your code intends to achieve.
- Logical structures and statements are properly used for specific purposes.
- Read the assignment specification carefully, and make sure that you follow whatever directed in this assignment. In every assignment that you will submit in this subject, you must put the following information in the header of your program:

```
/*----------------------------------------------------
My name:
My student number:
My course code: CSIT121
My email address:
Assignment number: 3
----------------------------------------------------*/
```

Objectives

This assignment requires you to write a program that designs and creates the graphic user interface (GUI) for the student enrolment system by using Java Swing.

Background

Assignment 3 is based on the Assignments 1&2 (please see Moodle for Assignment 1&2's solutions). Please refer to A1's description for the Bachelor of Computer Science course structure and A2's description for the Master of Computer Science course structure.

You are required to design, implement and test a student system with GUI to 1) load BCS or MCS course structure via the Java ObjectInputStream; 2) to enrol one student into either the Bachelor of Computer Science course or the Master of Computer Science; and 3) to save the student enrolment information into a text file. The detailed steps are as follows:

1. Upgrade your UML diagram.
You should upgrade your existing UML diagram for Assignment 3 by considering the new GUI class (named SEFrame extending JFrame).

2. Design a GUI.

The SEFrame class shall contain some basic Swing components such as JLabel, JTextField, JButton, JList and etc. With the new GUI, a user shall able to input his/her personal information, to select a course (either BCS or MCS), to select a major within the course and to select some elective subjects to complete the enrolment. An example of the GUI is displayed as follows. You can either use this GUI layout or design your own GUI.
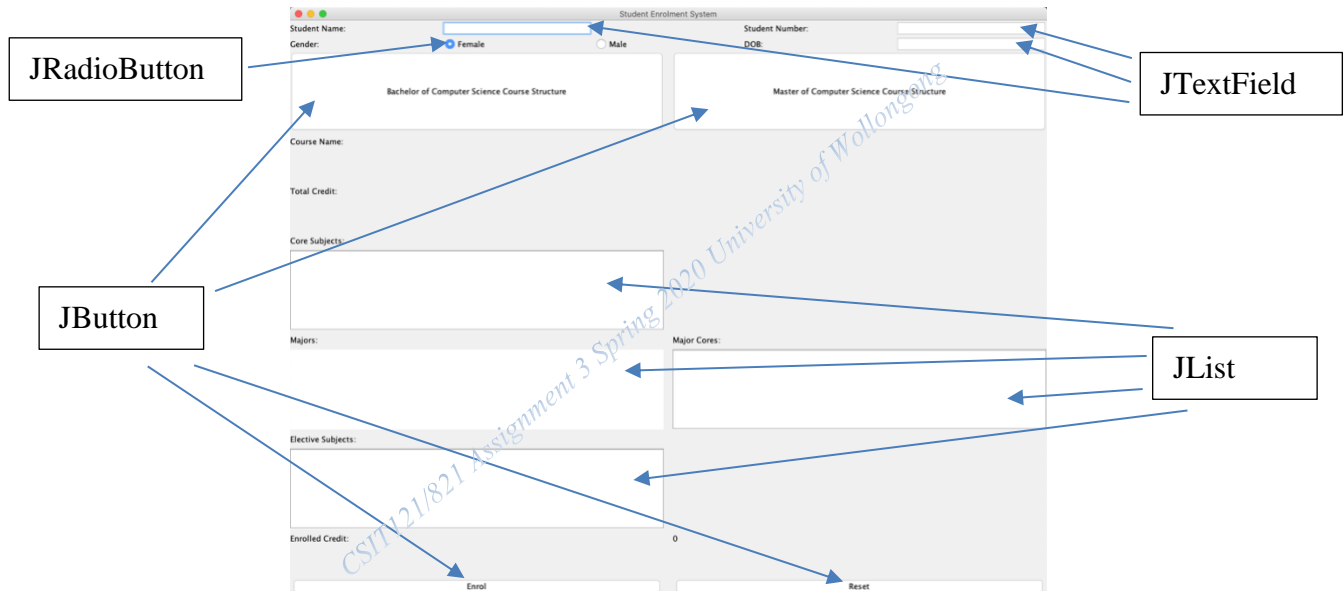
3. Load BCS or MCS course structure via Java ObjectInputStream

The Java ObjectInputStream shall be used to load either BCS ("bcs.ser") or MCS ("mcs.ser") course structure to the system. The given "bcs.ser" and "mcs.ser" files are generated based on A2's solutions. If you make any modification of A2's classes, you shall create new "bcs.ser" and "mcs.ser" files by using the Lab5's program. The given example shows the BCS or MCS course structure after clicking the "Bachelor of Computer Science Course Structure" or the "Master of Computer Science Course Structure" button.
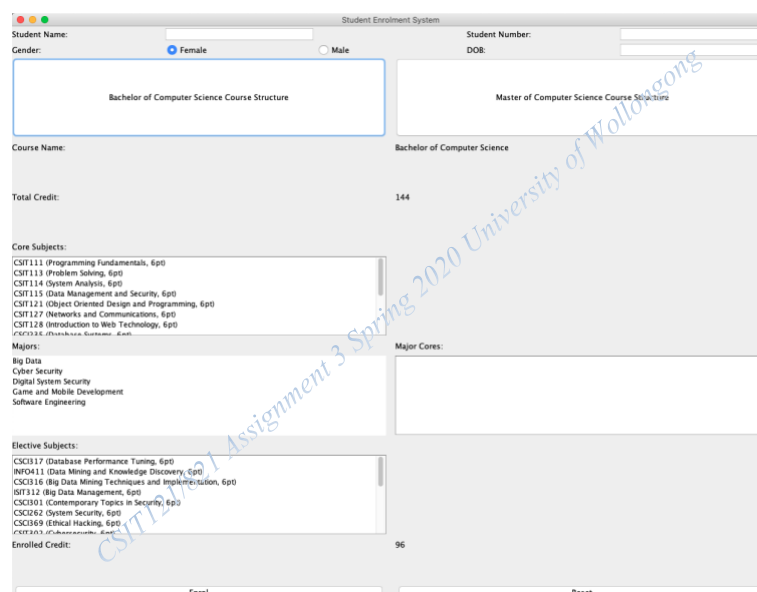
(by clicking the "Bachelor of Computer Science Course Structure" button)

(by clicking the "Master of Computer Science Course Structure" button)

4. Select a major for a course.

After a course structure is loaded.

i. The system shall automatically enroll all core subjects and update the total enrolled credits (96 cp for BCS, 54 cp for MCS).

ii. To allow the user to select a major and show the major core subjects. The system shall automatically update the total enrolled credits after the user selects a major (enrolling the user to all major core subjects automatically). The user is only allowed to select ONE major. For example, in MCS, if the user selects the Intelligent System Major or the Machine Learning and Big Data Major, the three major core subjects shall be displayed on the right side and the enrolled total credits shall be updated automatically. The elective subject list shall also be updated automatically by removing the selected major's core subjects.

(by clicking the Intelligent System Major)

(by clicking the Machine Learning and Big Data Major)

5. Select elective subjects

The user shall able to select elective subjects from the list. The user can make multiple selection and the system shall automatically update the total enrolled credits. For example, the user can select the top three subjects from the elective subject list which make the total enrolled credits to 90 cp.

(by clicking the top three elective subjects)

6. Enroll the student and create the enrolment record as a text file ("studentName_courseName.txt")

The user can click the "Enrol" button anytime to complete the enrolment. By clicking the "Enrol" button, the system will do the following things automatically.

i.   The system will check whether the user's personal information is provided and valid by using the exception handling (please see Lab 4's solution). The student's name must be a string, the student number must be an int. The DOB must be a string, use the format "dd/mm/yyyy" and the date must exist. If any information is not provided or invalid, a pop-up window will alert the mistakes and ask the user to complete/correct the inputs.

(The exception for not inputting the student name)

(The exception for not inputting the student number)

(The exception for inputting a wrong student number)

(The exception for not inputting the DOB)

(The exception for using a wrong DOB format "30-02-1999")

(The exception for inputting an invalid DOB "30/02/1999")

ii. The system will automatically check whether the user enrolls sufficient credits for the course. For BCS, the minimal requirement is 144 cp. For MCS, the minimal requirement is 96 cp. If not, the system will handle this as an exception and show a pop-up window to alert the user to enroll some extra subjects to reach the minimal requirement.

(The exception for not selecting sufficient subjects)

iii. When the user's personal information is completed and valid and the user selects sufficient subjects (the user can enroll more subjects than the minimal requirement), the system will complete the course enrollment (with a pop-up window) and create a text file to keep the enrolment record. The text file's name must be "studentName_courseName.txt".

(The user enrolls MCS course successfully)

```
1    -----------------
2    Student: Bob (1234567, Male, 28/02/1999)
3
4    Course Name: Master of Computer Science (ACTIVE)
5
6    Cores:
7    CSCI814 (IT Project Management, 6pt)
8    CSCI851 (Advanced Programming, 6pt)
9    CSCI803 (Algorithms and Data Structures, 6pt)
10   CSCI835 (Database Systems, 6pt)
11   MTS9302 (Corporate Network Management, 6pt)
12   CSIT940 (Research Methodology, 6pt)
13   CSCI920 (Contemporary Topics in Computer Science, 6pt)
14   CSCI992 (Professional Project, 12pt)
15
16   Major: Machine Learning and Big Data
17   CSCI933 (Machine Learning Algorithms and Applications, 6pt)
18   CSCI935 (Computer Vision Algorithms and Systems, 6pt)
19   CSCI946 (Big Data Analytics, 6pt)
20
21   Electives:
22   CSCI862 (System Security, 6pt)
23   CSIT826 (Human Computer Interaction, 6pt)
24   ISIT925 (Strategic Network Design, 6pt)
25   CSCI964 (Computational Intelligence, 6pt)
26   CSCI924 (Reasoning and Learning, 6pt)
27
28   Total Enrolled Credit: 102pt
29
30   Expected Master Graduation: Spring/2022
31   -----------------
```

(The enrollment record is kept by a text file, named "studentName_courseName.txt". The expected graduation date can be specified manually in the program.)

7. Reset the system

The user shall able to click the "Reset" button to reset all inputs, course structure, user's selection and the total enrolled credit anytime.

(The reset button is clicked)

Tasks (10 marks)

Task 1 (2 marks): Update the UML class diagram by adding the SEFrame class. Complete the GUI design and implementation.

Task 2 (6 marks): Implement the system according to the specified steps. The program shall

- be consistent with the UML class diagram;
- follow the conventions for naming all classes, variables, and methods;
- provide sufficient comments;
- use proper blank spaces, indentation and braces to make your code easy to read and understand;
- follow the steps specified above to enrol one student into either the Bachelor of Computer Science course or Master of Computer Science course;
- be able to load BCS or MCS course structure via Java ObjectInputStream;
- be able to response to user's selection;
- be able to automatically update the enrolled total credits and calculate the remaining elective subject number for the minimal credit points requirement;
- be able to do the validation checking and handle the exceptions;
- be able to create a text file to keep the user's enrolment record;
- be able to reset the system anytime.

Task 3 (2 marks): Compilation and test.

- You shall compile and test your program by using two different cases. The first testing case must enrol a student (bob) to the Bachelor of Computer Science course and the second testing case must enrol another student (Amy) to the Master of Computer Science course.
- You shall also test the exception handling of your program.
- Please carefully compile and test your program and make sure your program can pass the compilation by using "javac" command. Please do not define the package in your program (a special alert for students who use IDE to complete the assignment).

Submission:

- Please submit your solution to Moodle (Assignment 3). Email submission is not accepted.
- Please submit an **individual PDF document (SEFrame.pdf)**, **a Java program** (**SEFrame.java) and both 'bcs.ser' and 'mcs.ser' files.**
- The SEFrame.java shall **contain all classes** of the system. The Java program's name must be **SEFrame.java**.
- The PDF document shall contain 1) the UML class diagram; 2) the GUI design and explanations; 3) the snapshots clearly shows the compilation and the execution of your program with the two testing cases (including all the exception handling); 4) the content of the student enrolment record txt files.
- Please DO NOT include the java code in the PDF document.

NOTES:

Enquiries about the marks can only be made within a maximum of 1 week after the assignment results are published. After 1 week the marks cannot be changed.

Mark deductions: compilation errors, incorrect result, program is not up to spec, poor comments, poor indentation, meaningless identifiers, required numeric constants are not defined, the program uses approaches which has not been covered in the lectures. The deductions here are merely a guide. Marks may also be deducted for other mistakes and poor practices.