





Workshops 4 & 5

Description:

This assignment lets you practice concepts such as Object Serialization, and Swing GUI.

In this assignment, you will be working with some **objects of a Student class** (which should be **serializable**.)

-  This class has fields such as **stdID** (int), **firstName** (String), **lastName** (String), and **courses** (an array or preferably an **ArrayList** which contains the **names of the courses the student currently has**). Please provide **constructors, getters, and setters**, where appropriate.
-  In the **second class**, you will need to **enter some information** for some **student objects from the console** and **save** these student **objects in a file**.
-  And finally, in the **third class** of this project, you need to **read those student objects from the file**, and then **show their info in the console**, to verify that they had been saved (and serialized) correctly.
-  Lastly, design a **Swing GUI-based Java application** to do the **same job**. It could be as simple as (at least) a **JFrame** with some **JTextFields** and a **JButton** (to let the user enter the information of the students and save them), and a **JTextArea** and another **JButton** (to let the application to read the information of all saved students from the file and show them to the user.)

Note that we should be able to run the third class and retrieve the data (student objects) already stored in the file independent of running the second class first (assuming we have serialized some objects by running the second class in advance). These two classes should be treated and/or run as two independent processes/programs in your project. The same applies to the GUI version; you could have an interface which lets you choose whether you want to read/load or save students' info.

Marking Criteria and Tasks:

Please note that you should:

- a- have appropriate indentation.
- b- have proper file structures and modularization.
- c- follow Java naming conventions.
- d- document all the classes properly.
- e- not have debug/useless code and/or file(s) left in assignment.
- f- have good intra and/or inter class designs.

in your code!

- Task 1: Developing and running the desired solution (The three classes mentioned above,) correctly, in the console: **5 marks**.

- Task 2: Developing and running the desired solution (The three classes mentioned above,) correctly, using Swing GUI: **5 marks**.

Deliverables and Important Notes:

You are supposed to show up AND hand in your solution in person (run the solution and/or answer related Qs) in the labs 6 and 7. **Task 1 should be delivered in lab 6 and task 2 should be delivered in lab 7.**

In case you don't show up OR hand in/run the required task in the lab, you could submit your final solution (described below) on the same due date but note that there would be a 50% penalty! Late submissions would result in additional 10% penalties for each day or part of it.

In this case, you should zip *only the Java files* to a file named after your Last Name followed by the first 3 digits of your student ID. For example, if your last name is **Savage** and your ID is **354874345** then the file should be named **Savage354.zip**. Finally email your zip file to me at reza.khojasteh@senecacollege.ca

Remember that you are encouraged to talk to each other, to the instructor, or to anyone else about any of the assignments, but the final solution may not be copied from any source.