**CSCI 2120**

**Homework : Serialization**

**Introduction**

For this assignment you will, starting with some implementation of classes for Students and a StudentDatabase, write classes that will serialize and deserialize an entire StudentDatabase (with very little code, and with very little concern for formatting.

**Procedure**

1) You are provided an implementation of a class that represents a Student, and a class that represents a StudentDatabase. (NOTE: the StudentDatabase is implemented as a Singleton). I have also provided you with a class, MakeRandomStudents, that can be run to produce random Student data, in a specified format, and print it out to the screen. For example, running the following from the command line:

java MakeRandomStudents 100 > students.txt

will cause the program to generate 100 random students, and the last big means “redirect standard output to the following file”. “>” is the redirection operation and we can send text that would normally appear on the screen to a file from within the shell. A single “>” will overwrite the file if it exists, or create it if it does not. A double “>>” will APPEND to the file if it exists, or create it if it does not. Use this command to produce some test data to work with, and have a look at the file produced.

2) You are also provided with a few other, fully implemented classes, StudentDatabaseCSVFileReader (which can read in the data you just generated from a file and build a StudentDatabase object), and a class that holds a main method BuildAndSortStudentDataFromCSVFile which takes two arguments: the formatted text file to read in, and the formatted text file to write out. After reading, the Student objects in the StudentDatabase are sorted based on their GPA, and the data is written out to the file listed as the second argument. For example:

java BuildAndSortStudentDataFromCSVFile students.txt sortedStudents.txt

will read in the formatted text file above, and write out the data to a file called sortedStudents.txt.

The writer part, the class StudentDatabaseCSVFileWriter, is where the “writing out” happens. You’ll notice that the write() method is not fully implemented (search for the string YOUR CODE HERE in the comments). You’ll finish writing this method, so that the output is in exactly the same format as the input file was. Use both the Reader, and the MakeRandomStudents classes as a guide to what the format is…. **(30 points)**

3) Write two new classes, StudentDatabaseSerializedFileWriter, and StudentDatabseSerializedFileReader, that, in a similar way, will serialize and deserialize, the entire StudentDatabase in one write operation, respectively. You’ll need to make both Student and StudentDatabase subtypes of Serializable to accomplish this. Write another class, BuildAndSerializeStudentDatabase that starts as a copy of BuildStudentDatabase, which reads in a formatted file, and then serialized the database to a file. Finally, write another class like BuildStudentDatabase that reads in a serialized StudentDatabase from a file, and writes the data out to a formatted text file. **(70 points)**

**Submission**

You will add, commit, and push your program to Gitlab.

**Grading**

Your code should provide fully functional object serialization and should be able to easily serialize and deserialize the entire database at once.