CSCI 201 – Computer Science 1

Final Project: An interactive program that processes student data.
Part II. Due on Friday April 26.

Stage 3. *Processing the data and printing reports.* Study the files in `CourseFiles/FinalProject/Stage3.cpp` as you work on this. Add the following features:

1. *Processing the data.* To process a student record, we have a method `void processStudent()` in the class `Student`,
which processes the data by calling additional functions to compute aggregate scores, letter grades etc.
2. *Generating a report of all the student grades.*
In this case, the menu function gets the output file, opens the output stream and invokes a method on the `studentList`, passing the output stream as a parameter. This method calls a function on each student in the list.
3. *Searching the collection. (Note: For the employee case study the* `search()` *operation has been implemented in* `FinalProject/Stage4.cpp` *)* This involves the following steps:
(i) getting the "key" for the search, which could be a name or an id
(ii) searching the collection
(iii) displaying the result
The first step requires interaction with the user and is therefore done in a menu function. The second one requires searching the collection; we add new methods for doing this by name or ID. The third step requires displaying the individual item (i.e., a student, if found) and is is done in a separate function. If you used arrays of character, then searching requires comparison of C-strings; read sections 10.3 and 10.4 in the text. (In the samplecode, id is a string, whereas name is an array of characters, i.e., C-string; you can contrast the way in which we use the two.)

Q 1 Create a design document in the 5-column format explaining the following: (i) how a report will be generated, and (ii) how the collection will be searched.
Q 2 Generate a script file showing source, compilation and test.
What to turn in (Stage 3): Design documents for Q 1, source files, and script file (Q 3.2) into folder `FinalProjStage3`.

Stage 4. *Modifying and saving the data.* Add the following features:

1. *Adding a student.* The menu function will prompt the user and get all the information about the student; user types in the entire data on one line, just as it appears in the file. Rest of this is similar to what we did in Stage 2 to add the student.
2. *Modifying the final exam score.* User provides first the ID, and if student found, main program asks for new final score. This will invoke the search-by-ID feature to get the index(i) of the cell; we need a method that will return a reference to the ith student, so that we can modify the student object that was found.
3. *Saving the modified data.* Menu function gets the file name(s) from the user into which the data will be written back. Menu function then calls a studentList method to write all the students to the file, which in turn calls a method to write each student to the file.

Q 3 Create a 5-column design document explaining how the final exam score will be modified.
Q 4 Generate a script file showing source, compilation and test.
Q 5 Create a testing table to report on the tests that were performed.

What to turn in (Stage 4): Answers to Questions 3, 4 and 5 into folder `FinalProjStage4`.