

CSCE 121 Assignment #7

due November 10

Objectives: *This assignment gives you some experience with designing and writing C++ programs using pointers, arrays, strings and linked lists.*

All work should be pre-graded in the labs and turned in to eCampus by the deadline. For programs, you need to turn in only the source code (not object code or executable code). Your code will be tested using a g++ compiler. Therefore, you are welcome to develop your program in Visual Studio or Xcode, but make sure your code also compiles using g++ and runs on a Linux machine.

1. (10 points) Create a text file, called README, in which you provide:

- Your First Name, Last Name, UIN, Section Number, User Name, E-mail address
- State the Aggie Honor statement:

I certify that I have listed all the sources that I used to develop the solutions and code to the submitted work.

On my honor as an Aggie, I have neither given nor received any unauthorized help on this academic work.

Your Name

Date

- List any resources used such as webpages (provide URL). Do not mention the textbook and discussions with the Instructor, TA, or Peer Teachers.
- List any known problems with the assignment you are turning in. For example, if you know your code does not run correctly, state that. This should be a short explanation.
- Provide a list of problems.
- Provide a short description or pseudocode.
- Write how you tested your program(s) for correctness and how you used exceptions for the error handling.
- Submit to eCampus an electronic version of the README file along with your C++ source code for the problems listed below and a hard copy to your TA.

2. (30 points) Complete the drills

(a) 1 through 7, p. 515. The code for this problems should be in a file called:

LastName__UIN__Drills-num_p515.cpp

where num is the number of a drill. The order of instructions should follow the order of the drill problems, comment each instruction.

The code for the drills above should be completed during the lab time, and presented to your TA or PTs before its submission to eCampus.

3. Complete the exercises from the textbook:

(a) (10 points) Exercise 3, p. 624

(b) (10 points) Exercise 4, p. 624

Important: The files should be named:

LastName__UIN__Ex_3.cpp,

LastName__UIN__Ex_4.cpp

4. (30 points) Implement a class `Singly_linked_list` that can hold objects of type `Student_struct` (see quiz 7). The public part of this class should contain parameter and non-parameter constructors, and the functions: `insert_first`, `insert_after`, `remove_first`, `remove_after`, `print_list`. Declare the overloaded output `operator<<` in the header file but not inside of the definition of the class `Singly_linked_list`. Read in data from a user file.
5. (10 points) Use the inheritance property to implement a class `Sorted_Singly_linked_list` with the same functionality as the class `Singly_linked_list` from the previous item but the list must be sorted with respect to the students IDs. Declare the overloaded output `operator<<` in the header file but not inside of the definition of the class `Sorted_Singly_linked_list`. Read in data from a user file.

Important: The file for both the classes should be named:

LastName__UIN__Linked_lists.zip. where LastName is your last name and UIN is your UIN number.

Submit all these files together with the drills as a zip/tar file to eCampus for grading.