Data Structures (2028C) – Fall2019– Lab 2

Topics covered: File I/O, Classes and multi-file programs

Lab due: ***11:55PM before the next Lab***

Objective:

The objective of this Lab is to create, read from and write to files, design and implement classes using correct class file separation.

Task 1: Read from files

1. Create a new project.
2. Write a program to read from and display the contents of the first 10 lines of a text file. It should prompt the user for the name of the file.
3. Include in the program logic to gracefully handle the case that the file doesn't exist
4. Take a screen shot of the output screen to include in your lab report. Include in the submission your test files that include at a minimum a file shorter than, exactly, and longer than 10 lines.

Task 2: Writing to files

1. Create a new project.
2. Modify the code from Lab 1, Task 2 to write the output to a file rather than the command line. The user should specify the name of the file.
   a. If the file doesn't exist, create it.
   b. If the file does exist, append to the end of it.
3. Take a screen shot of the output screen to include in your lab report. Include in the submission your result files from testing without a preexisting file and with appending to a file.

Task 3: Classes.

1. Create a new project.
2. Modify the code from Lab 1, Task 3 to use a class instead of a structure. Don't initialize the values.
   a. Place the Class declaration into a separate .h file.
   b. Place the Class definition into a separate .cpp file.
   c. Add members for description (string), and TaxExempt (bool)
3. Write the code to read the values for the items of the class from a text file. The text file should be human readable (the text file may look like the following repeated as many times as necessary although you may decide on a different file format as long as it is human readable)

   ID: 914
   Units: 842
   Price: 12.95
   Description: Blue widget used for powering devices.
   TaxExempt: False

4. Ensure you have tested this with no input file, an empty input file, an input file that has less than the number of items than defined in the array, and an input file with more than the number of items than defined in the array.
5. Include the input and output files you used in your lab report.

Lab Submission:

1. Write a lab report including the following information:
    a. A description of the objectives/concepts explored in this assignment including why you think they are important to this course and a career in CS and/or Engineering. Include screen shot(s) from Task 1 and 2.
    b. Why you selected the file access flag(s) you selected including what access flag(s) considered but didn't use for each of the Tasks.
    c. An explanation of why you designed the class and input file the way you did in Task 3.
    d. What you learned about testing in the 3 Tasks. Did you consider the test cases before you stated, during or after you completed coding?
2. Include all source code from all Tasks, input and output files, and any special instructions to compile and run those programs.
3. Package all files in a single zip folder and submit the file to ________@mail.uc.edu.

Lab Grading:

1. 20% - Lab attendance
2. 15% - Task 1 has been correctly implemented and meets all requirements.
3. 20% - Task 2 has been correctly modified and meets all requirements.
4. 25% - Task 3 has been correctly modified and meets all requirements.
5. 20% - Lab report contains all required information and is well written.

If program fails to compile, 0% will be given for that Task.