Southern New Hampshire University

CS 260 Up-to-Speed Activity Guidelines and Rubric

Overview: In this course, you will be responsible for developing programs in C++. When you are learning a new programming language, it is important for you to realize that, while there are differences in syntax and the standard/core libraries, the basic programming fundamentals are the same. Loops, variables, and operators function in the same manner, regardless of the language. If you keep that in mind, three of the most useful techniques to learning a new language are the following:

1. Having a reference for the language (zyBooks serves as this, especially the included reference sections)
2. Having well-written code to use as examples
3. Writing code in the language

The assignments in Modules One and Two provide both sample code and practice writing C++ code. You will build a simple console program that uses a menu to enable testing of the logic you will complete. The code in Lab1-3.cpp and Lab2-1.cpp is partially complete, with token placeholders such as:

- ?type?
- ?variable?
- ?retval? (return value)

Follow the specifics in the Lab1-3 and Lab2-2 Up-to-Speed Requirement Details documents for more comprehensive instructions for each of these assignments.

Prompt: The following **critical elements** should be addressed in your assignment submission:

- **Functioning Code**
    - Code must meet its specifications and behave as desired. To develop proper code, you should **produce fully functioning code** (with no errors) that aligns with accompanying annotations.
    - You should write your code in such a way that it actually executes, even if it does not produce the correct output. You will be given credit for partially correct output that can actually be viewed and seen to be partially correct.

- **Accurate Results / Output:** A properly generated result means that your script:
    - generates accurate output, and
    - produces results that are streamlined, efficient, and error-free.

- **Annotation / Documentation:** All code should also be well-commented. This is a practiced art that requires striking a balance between commenting everything, which adds a great deal of unneeded noise to the code, and commenting nothing. Well-annotated code requires you to:

    - explain the purpose of lines or sections of your code detailing the approach and method the programmer took to achieve a specific task in the code, and
    - document any section of code that is producing errors or incorrect results.

- **Style and Structure:** One lesson to be learned in this course is how to write code that is clearly readable and formatted in an organized manner. To achieve this, you should:
    - develop **logically organized code** that can be modified and maintained, and
    - utilize proper **syntax**, style, and language conventions/best practices.

Rubric

Guidelines for Submission: For each exercise, include the C++ (.cpp) file, including a header comment containing a title (name, course, date, project number).

Critical Elements	Proficient (100%)	Needs Improvement (70%)	Not Evident (0%)	Value
Functioning Code	Program is fully functioning and includes code to meet all specifications	Program is not fully functioning or does not include all of the specifications of the given problem	Program does not run, or significant details of the specifications are violated or omitted	30
Accurate Results / Output	Data results are accurate in regard to the given problem	Program produces incorrect results for the given problem	Program does not produce results for the given problem	20
Annotation / Documentation	Code annotations fully explain the code and facilitate navigation of the script code	Comments provide little assistance with understanding the code, and code annotations do not fully explain the code or do not facilitate navigation of your script code	Code is not fully or logically annotated	20
Style and Structure: Logically Organized Code	The code is logically organized	The code contains portions that are not logically organized	Code is poorly organized or very difficult to read	15
Style and Structure: Syntax	Code follows proper syntax and conventions	Code contains variations from established syntax and conventions	Code contains significant variations from established syntax and conventions	15
			Total	**100%**