



Competency

In this project, you will demonstrate your mastery of the following competency:

- Write professional, well-documented, and readable code

Scenario

As a recent graduate of SNHU, you've taken the time to consider the type of work you really want to do and therefore have focused your job search on junior developer positions. In searching the major job boards, you discover Chada Tech.



Chada Tech is a software engineering company that specializes in custom software design and development for both domestic and international clients. You read more about the company and decide this place would be a great fit for you. You are also happy to find that there is an opening for a junior developer and apply for the position immediately. The skills that you have acquired at SNHU have prepared you for this role. Consequently, you are selected for an interview!

As part of the Chada Tech interview process, you are asked to complete a coding skills test to “audition” for the role. This is a typical interviewing strategy that you may encounter in the field. Companies utilize this technique to ensure candidates can in fact do what they claim they can do. Therefore, this exercise will be based on a problem similar to what you will face in the junior developer role, and you will need to come up with a solution. This is your chance to really show them what you can do!

The Chada Tech interviewing team has provided you with the following problem statement and a document containing a set of functional requirements:

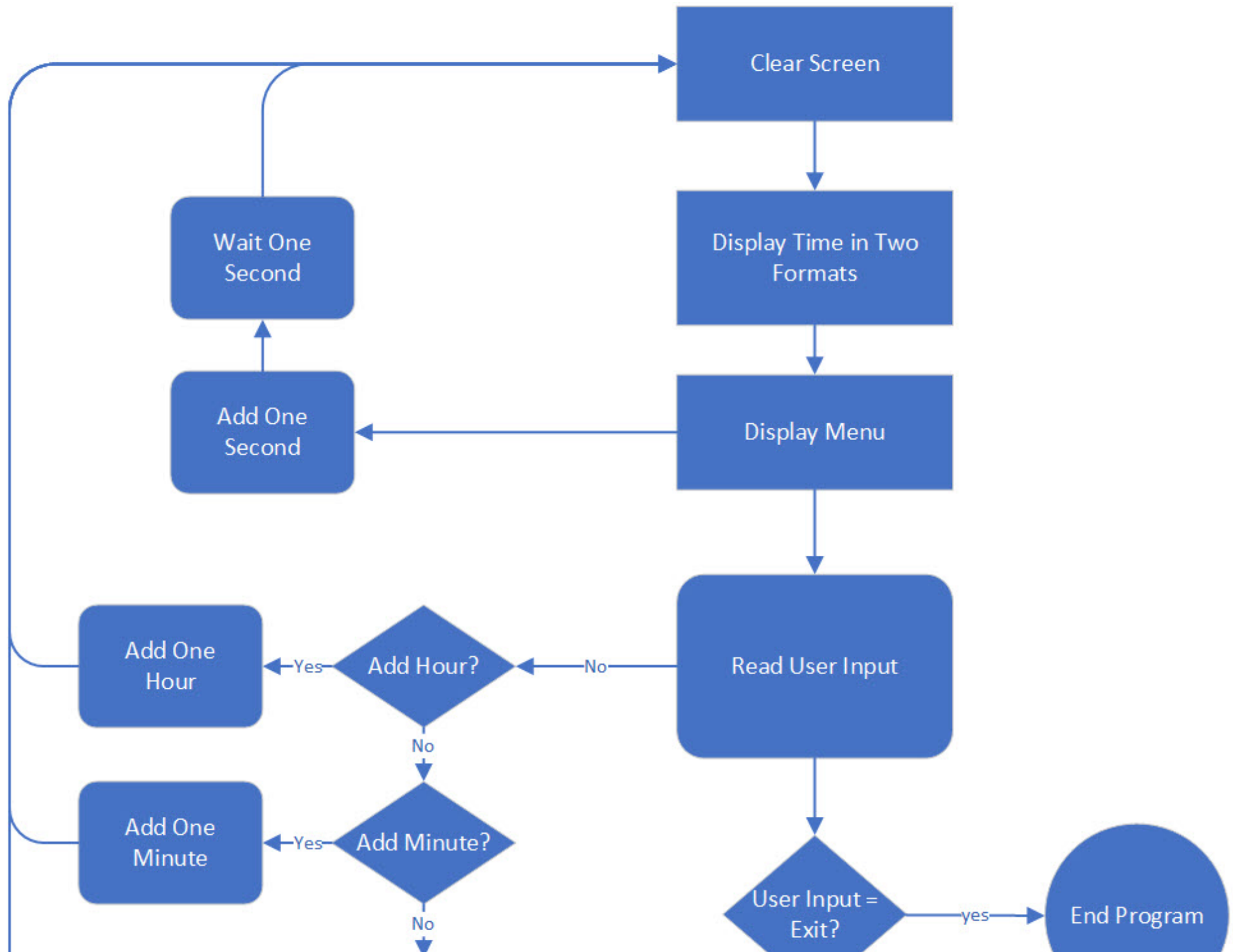
Chada Tech has domestic and international clients. To meet international standard [ISO 8601](#), Chada Tech wants their clients to be able to view a 12- and a 24-hour clock on their website rather than just the standard 12-hour clock.

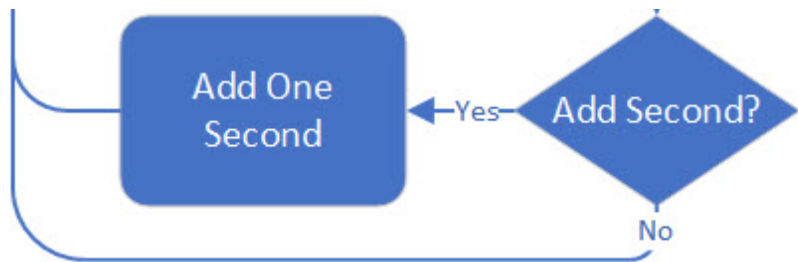
To stand out as a top candidate, you will need to apply program logic by creating two clocks that display simultaneously and allow for user input using secure and efficient C++ code.

Directions

Develop an object-oriented programming (OOP) application to create two clocks that display simultaneously and allow for user input using secure and efficient C++ code.

1. Review the Chada Tech Clocks Functional Requirements, located in the Supporting Materials section.
2. Review the following flowchart, as it is a graphical representation of the logic you will use in relation to your application's sequence of functions. A text version of this flowchart is available: [Project One Flowchart Text Version](#).





3. Your code should be properly modularized and written in functions so that your main() has the least amount of code.
4. Before you begin coding, remember that you must demonstrate industry standard best practices in all your code to ensure clarity, consistency, and efficiency. This includes:
 - a. Inserting in-line comments to denote your changes and to briefly describe the functionality of the code
 - b. Using appropriate variable, parameter, and other naming conventions throughout your code
5. When your solution is finished, zip your project including all components (CPP, H, and any other files used).
6. Upload your zipped project file to the project submission area.

What to Submit

To complete this project, you must submit the following:

Clocks Code

Submit your zipped project files. Be sure to include a file header with your name.

Supporting Materials

The following resource(s) may help support your work on the project:

Reading: [Chada Tech Clocks Functional Requirements](#)

Chada Tech provided you with this document, which includes all of the functional requirements for the clocks you will create. Your code must meet all of these functional requirements.

Reading: [Visual Studio Export Tutorial](#)

This guide will walk you through how to download all of your work from Visual Studio as a ZIP folder.

Project One Rubric

Criteria	Exemplary (100%)	Proficient (85%)	Needs Improvement (55%)	Not Evident (0%)	Value
Follows Flowchart Diagram	N/A	Code follows flowchart diagram logic in relation to the application's sequence of functions (100%)	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include building code based on the diagram provided and its sequence of functions (55%)	Does not attempt criterion (0%)	10

Criteria	Exemplary (100%)	Proficient (85%)	Needs Improvement (55%)	Not Evident (0%)	Value
Code Modularization	N/A	Code effectively includes functions and minimizes main() (100%)	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include minimizing main() by writing more code functions (55%)	Does not attempt criterion (0%)	20
Code Execution: Clock Displays	N/A	Code produces correct results for displaying a 12- and a 24-hour clock, simultaneously, in the correct 12- and 24-hour formats (100%)	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include displaying clocks side-by-side, proper clock format, etc. (55%)	Does not attempt criterion (0%)	15
Code Execution: Menu Functionality	N/A	Code displays and enables correct menu function options in the user menu (100%)	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include displaying all functional criteria for the user menu (such as adding an hour, minute, or second to both clocks) (55%)	Does not attempt criterion (0%)	20
Code Execution: Responds to User Input	N/A	Clocks produce correct results based on user selections from the menu (100%)	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include proper clock manipulation based on user selection, etc. (55%)	Does not attempt criterion (0%)	20
Industry Standard Best Practices	Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner (100%)	Uses industry standard best practices such as in-line comments and appropriate naming conventions to enhance readability and maintainability (85%)	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include naming conventions or in-line comments (55%)	Does not attempt criterion (0%)	15

Criteria	Exemplary (100%)	Proficient (85%)	Needs Improvement (55%)	Not Evident (0%)	Value
Total:					100%