Module 1 Coding Assessment

The assessment helps you validate your understanding of the module one objectives:

- Variables and data types
- Conditional and iteration logic
- Object-Oriented Programming
- File I/O and Collections
- Unit Testing

It also gives you practice with coding assessments you may encounter during the job interview process.

Overview

This is a time-boxed, individual coding assessment. You have one hour to complete the assessment. You are not expected to complete all the features of the problem in the time given. Complete as much as you can.

You may use any resource available excepts human beings. This is an individual effort.

Whatever you submit should not have compile or run-time errors. Features that are completed should run successfully to receive credit. Any unit tests you write should be error-free and run successfully.

If you are nearing the end of the assessment and you have compile/run-time errors, consider commenting out or removing the code that causes the errors.

At the end of the assessment, you should push your final project to your repository.

If you finish before the end of the hour, please be respectful of those still working. Please leave the classroom if you would like to talk with others that have completed the assessment.

Instructions

1. Pull from your repository like you do every day. The assessment is under the `module-1` folder of your repository.
2. Open the project in the `00-Assessment` folder.
3. You should see the typical folder structure for a project with a skeleton application program (`main()` or `Main()`) method.
4. Any files you create should be placed in the correct folders within the project.
5. Be sure to push your project when you are done or at the end of the assessment.
6. Your instructor will tell you when to begin and give you a "10-minute warning" before the end of the time-box.

Scoring Guidelines

0 - Nothing has been attempted, or the project includes compile errors in the code or tests. The project may compile, but there is less work than required for a 1.

1 - The class, attributes, and constructor are created without error, and the main()/Main() methods instantiate at least one object.

2 - All requirements for a 1 have been satisfied. Also, the additional methods have been implemented, including toString()/ToString(). Methods that are used in main()/Main() have been tested.

3 - All requirements for a 2 have been satisfied. Methods have been unit tested and File I/O code to read in the .csv is implemented.