Tech-Elevator-C-Sharp-Dotnet-Artifacts

Curriculum Structure

The Tech Elevator C#/.NET curriculum is divided into four modules:

- **module 1** - Basic C#/.NET and OOP
- **module 2** - SQL Server and Database Access from C# using DAOs
- **module 3** - Front-End - HTTP - REST - Writing and consuming APIs
- **final-capstone** - a two week long project chosen from a list of options develeoped as an Agile team

Each module has a set of digital artifacts stored:

- **LMS** - Each topic has a required reading assignmeng in the LMS to give them a basic introduction/understanding of a topic. Students are expected to complete the reading and the associated tutorial before the night before a topic is taught. Details of the topic and clearing up any confusion from the tutorial is done in the instructor led code along training session.

- **topic folder** containing:

    - **README.md** - Overview of the topic
    - **lecture** folder - starter code for instructor led code along to teach the topic
    - **lecture-notes folder** - notes for the instructor containing topic details and examples of how it may be taught
    - **exercise folder** - assigned to students after the lecture to be completed within 2 days of assignment
    - **exercise-final folder** - a solution to the exercises given to the students 2 days after the assignment due date
    - **tutorial folder** - an assignment the student is expected to complete after doing the tutorial in the LMS and before the lecture on the topic
    - **tutorial-final folder** - a solution to the tutorial assignment which is given to the students before the start of the lecture on the topic. Some instructors used this as a review of what they learned in the tutorial.
    - Most folders contain a README describing how the folder is expected to be used. *

- **reources** folder - contains suppemental files and scripts used in the curriculum

Suggested use for Galvanize Team to "get up to speed" on C#

- If you are new to C#:

    - reading the *tutorial* for a topic in the LMS will acquaint you with how that topic is implemented in C#/.Net
    - Review the code in the *tutorial-final* folder to see how the basic features of a topic are implemented
    - Review the *lecture-notes* for a topic to get additional details on the topic
    - While reviewing the *lecture-notes*, also relate them to what is coded on the *lecture-final* code to reinforce the topic

- If you are refeshing your knowledge, perform the steps above for the topics you want to refresh