Exercise

TDD - Moq

march 2019

The Moq Koans

This is an exercise that contains a set of Koans for learning how to mock objects with Moq. The Koans are written in C# and use NUnit.

What are Koans?

A koan is a Zen Buddhism story or statement that focuses not on a rational answer but more of an intuitive one. These lessons intend the answerer to look beyond the obvious for a deeper smarter more meaningful answer.
The **goal** of a **koan** is **not the answer**, **but thinking about the question**.

How do I use the project?

Download the "MoqKoans.zip" file, extract it and open the MoqKoans.sln file with Visual Studio.
To use the Koans, start by opening the file "1_Interfaces.cs".
Go through each Unit Test in order.

1. Run the test to see that it is currently failing.
   (Run a single test method at a time, not all the tests in the project)

2. Anywhere you see underscores (___), replace it with appropriate code.
   In each test method, unless otherwise noted, you typically only need to replace the underscores to make the test pass.
   Think about what the answer should be, and why.
   Sometimes the test failure message will provide some hint as to what the answer is.

3. Rerun the test to see if it passes.
   This process of verifying the test is failing, then fixing it follows the "red-green-refactor" approach to test-driven development. In this case we aren't worrying about refactoring, but it is still good to get in the "red-green" flow of testing.

As you finish all the unit tests in that file, move to the next .cs file in the project.

Keep in mind that the goes is to learn how Moq works, not just to make all the tests in the project pass.
You should think about why a certain replacement would make the test pass, not just blindly fill in replacements based off the test failure messages.
For example, if you run a test method and the failure message says "Expected 5 but was ___", then don't just replace ___ with 5 and move on. Think about why 5 is the appropriate replacement, and preferably think about it before you look at the test failure message.