ASOS Software Engineering Code Test

Time limit: 2 hours

This is a refactoring exercise based on a C# Visual Studio solution which currently has no unit tests and a number of code quality and testability problems.

# Requirements

We would like you to refactor the AddCustomer() method in the CustomerService class in order to make the class easier to read and maintain, and to be unit testable, applying good engineering principles such as SOLID, DRY, YAGNI, KISS and Tell-Don’t-Ask and a test-driven approach.

You must write at least one unit test, which covers the successful addition of a customer without relying on any external dependencies. You should also write any other tests that you think are required to cover the logic.

You can change anything in the App project (method signatures, constructors, etc.), apart from making the CustomerDataAccess class or AddCustomer()method non-static.

This package is used as a library by other applications within the organisation. You therefore must not change anything which will break the Harness project, which represents the existing integrations and backwards compatibility with these must be maintained. You must also not change the code in this project.

We know the validation logic isn't ideal but it is beyond the scope of this test to fix or update it.

# Your solution

We want to see how you break down the problem and we’re looking for simple, clean, readable code to demonstrate this. Please try to avoid over-engineering or gold plating and please do not go over the two-hour time limit.

Please note that all of your unit tests mustpass when we execute them – a submission with failing tests will not pass the review stage.

You can use whichever test and/or mocking frameworks you wish. We’re not expecting the refactor to be ‘finished’ as there are many things which could be improved given enough time but it should meet all of the requirements above.

We think that good code doesn’t need to be explained in comments, however you’re welcome to add additional narrative for the purposes of the code test to explain a choice of approach, or what you would do given more time.

Feel free to use the internet to look up anything you need. Once you are done, please reply to the email you received containing a zip file of your completed test (cleaned to remove binaries), with your name in the file name.