PROG 2070: Winter 2025

Programming - Software Quality Assurance

Assignment #2 – Unit Testing

[75 Marks – worth 10%]

Due Date: Details in eConestoga

This assignment is to be done in groups of three. Your group are to do your own work and do not share with other groups. Sharing work is an Academic Offense and is subject to a penalty. Be aware that all source code and other documentation is automatically checked by eConestoga upon submission.

*As well, the use of generative AI (such as CoPilot or ChatGPT) is **NOT** permitted for code generation. Using these tools to generate code for this assignment is an Academic Offence.*

For this assignment, you are going to develop and test a C# (.NET Framework) class in Visual Studio. You can set up your project as a console app, but do not need to test the console app as part of the project. You will need to create a class file and assess the quality of it by developing and executing unit tests on the class you have created.

The class you are creating would be part of an eCommerce application. The class itself will be for the Products of the application. The class will have a Product Constructor which will have the following attributes:

ProdID
ProdName
ItemPrice
StockAmount

These attributes should hold the value reflected in their name. The class should also include Methods to Increase the stock and decrease the stock. You will use Visual Studio to create you class and will use Git to track your changes.

Once you have created your project, use Git Bash to initialize a Git repository within the Project folder you just created. Do your first initial commit of your project folder in Git (remembering to include a specific commit message).

Return to Visual Studio and begin creating your Product class. The class should be a separate file. The Product constructor will hold the attributes of the product and the increase and decrease methods will increase and decrease the Stock count. You will need to be able to get and set the attributes.

When you have completed work on the Product class, return to Git Bash. Stage and commit your entire solution and label the commit as containing the completed Product class.

For this assignment, you only need to work with your Product class. This class should be separate from the Program class. Note, the Program class does not need any functional code.

Return to your project and begin work on creating unit tests for your Product class. Your project should look similar to the following:

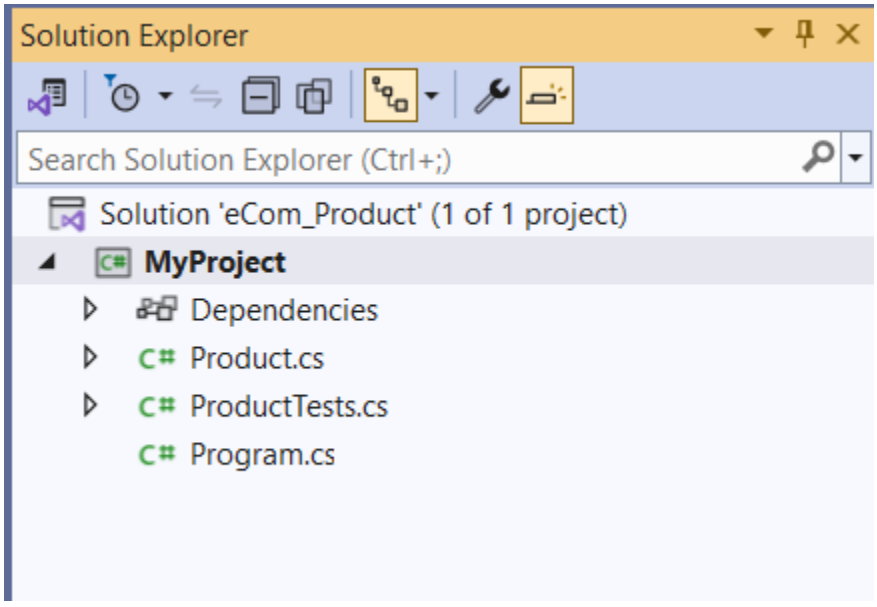

Figure 1

You have your Main Project, which contains a placeholder class for your application (Program.cs in Figure 1) and the Product class (Product.cs in Figure 1). You will have a second project in the same solution containing the test cases. Remember the naming conventions for the Test Project, the Test Class, and the Test Methods.

Develop three (3) unit test cases for each of the attributes and three (3) each for stock increase methods for a total of 18 unit tests.

Each team member will be responsible for writing 6 tests. As well, each team member needs to include a comment block to describe what their tests do and why they chose those tests.

For your tests, assume the following maximum and minimum values:

Product ID: 5 - 50000
Price: $5 - $5000
Stock: 5 - 500000

Each of your tests MUST use the Assert.That() assert method using the appropriate expected and actual variables.

Ensure that your tests are properly named as per the convention shown in the course slides and that they are created using the proper A-A-A format along with the appropriate spacing. You should also use Boundary Value Analysis when selecting your test data.

When you have completed work on the unit tests, return to Git Bash. Stage and commit your entire project and label the commit as containing the completed unit tests.

Provide a single screenshot of all 18 Unit cases being run in the VS Code Test Explorer – do not submit 18 separate screenshots of each individual unit test. All of your unit tests should pass and be based on correct inputs and expected results. Provide a screenshot or output from a `git log` command showing your three commits. Feel free to run more commits as it makes sense for your project and work schedule. But at minimum, there should be the three commits as described in the assignment.

Submissions

The format for submitting the assignment is as follows.

An eConestoga submission containing:

1. A single MS Word document named:

Group#_Student1FirstName_ Student2FirstName_Assignment2.docx

The Word document should contain (in order):

a. Assignment Title Page with your names, student IDs, "Assignment #2" in the title and date;
b. Screenshot showing the results of your unit tests being run in Test Explorer, and a screenshot/output of your git repository log.
c. Copy of Product class source code (The Product.cs in example Figure 1 screenshot)
d. Copy of Unit Test class source code (The ProductTest.cs in example Figure 1 screenshot)

When providing a "copy" of your code in the document, please do not screenshot it, but rather copy the code from Visual Studio and paste it into your submission document.

2. A single compressed (.zip format) archive file containing:
   a. **the entire Project folder** of your source code. (Do not include the .docx file in the .zip file)
   b. Your Git repository

Please submit the above to the **"Assignment #2"** assignment folder in eConestoga

Late penalties will apply for any late submissions.
A -10% penalty will be applied for any files that are zipped up or not using the correct naming format.
As this is a technical report, proper spelling and grammar will also be required and marks may be lost for reports that have poor spelling and / or grammar.
A more detailed marking scheme is shown in the Rubric associated with the Assignment Folder in eConestoga.