SIT323 Practical Software Development, Trimester 2, 2018

Week 4 – Practical 3

Unit Testing

Introduction

This practical will help you design unit tests without writing any code, and implement unit tests using Visual Studio.

Task 1 – Unit Testing

A template to help design and report results of several unit tests can be found in the file called **Unit Test - Template.docx**. Examples of several unit tests can be found in the file called **Unit Test - Examples.docx**.

The ZIP file **Class 3 - Unit Testing.zip** contains C# code of three test methods, these were presented during lecture.

Your task is to use these three test methods and the unit test template document to produce a unit test report. Use the examples in **Unit Test - Examples.docx** to help your produce this report.

Task 2 – Unit Test Design

You can design a unit test by partially completing a template. During design, you need not complete the following fields: Test Method, Actual Data, Test Result and Test Comment. These four fields can be blank during design. A design can then be given to a programmer to write a test method to implement the unit test(s). This test method can then be run using Test Explorer in order to complete these four blank fields.

For example, the following is a design for testing the TryRange() method of the Validator class in the old program.

Test Scenario ID	**Test Description**				
1	Check a value against a range, inclusive.				
Test Method	**Method Tested**				
	Boolean Validator.TryRange(int n, int lowerLimit, int upperLimit)				
Test Case ID	**Parameters**	**Expected Data**	**Actual Data**	**Test Result**	**Test Comments**
1.1	n = 123 lowerLimit = 100 upperLimit = 200	expectedReturn = true			
1.2	n = 98 lowerLimit = 100 upperLimit = 200	expectedReturn = false			
1.3	n = 321 lowerLimit = 100 upperLimit = 200	expectedReturn = false			

Your task is to add six more test cases to the above example, these are required to test the range boundaries, one test for each value of n = 99, 100, 101, 199, 200, and 201.

Task 3 – Unit Test Implementation

Use a copy of the Old Program to:

- Implement a test method for the design of Task 2.
- Run your test method using Test Explorer.
- Complete the unit test design table by filling in the blank fields.

Task 4 – Unit Test Design

Your task is to create a unit test design for each the following methods of the old program.

- `Boolean Validator.IsHexColourCode(String hexColour)`
- `Boolean Validator.IsFilename(String name)`

Your tests should be based on several valid and several invalid values.

1. For a valid hexColour value, the method is expected to return a true value.
2. For an invalid hexColour value, the method is expected to return false. In this case, if the method actually returned a true value instead of false, then we can conclude that this method is incorrect.

Task 5 – Unit Test Implementation

Use a copy of the Old Program to:

- Implement a test method for the design of Task 4.
- Run your test method using Test Explorer.
- Complete the unit test design table by filling in the blank fields.
