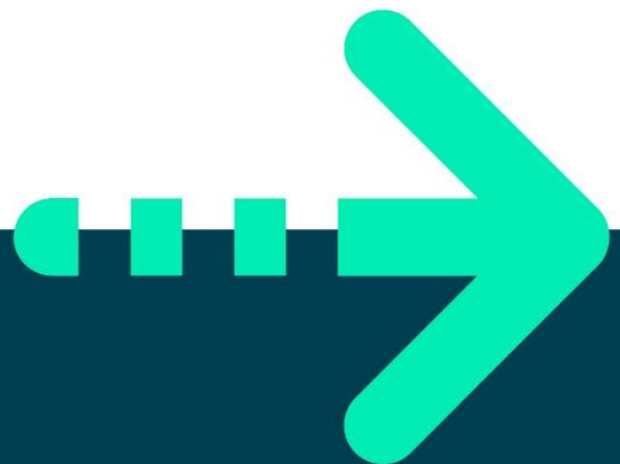




LAB 08, INTRODUCTION TO TESTING





Lab 08, Introduction to Testing Tools

Objective

The objectives of this practical session is to get you started with unit testing.

Overview

In this lab, we are going to use unit tests to test a method which checks if userID and password parameters are valid by following a set of rules.

Working with Security class

1. Create a new Console Application called **Lab13**.
2. Add a new public class to this package called Security
3. Add the following code to the Security class

```
public class Security {
    public string ErrorMessage { get; set; }

    public bool Logon(string userId, string password) {
        if (userId == null || userId.Trim() == "") {
            ErrorMessage = "UserID may not be null or empty";
            return false;
        }
        else if (password == null || password.Trim() == "") {
            ErrorMessage = "Password may not be null or empty";
            return false;
        }
        else if (!validatePassword(password)) {
            ErrorMessage = "Min 8 chars with an uppercase and a number";
            return false;
        }
        else {
            ErrorMessage = "";
            return true;
        }
    }

    private bool validatePassword(string password) {
        bool hasUpper = false, hasDigit = false;
        if (password.Length < 8)
            return false;

        foreach (char c in password) {
            if (char.IsUpper(c)) {
                hasUpper = true;
                continue; // continue loop as no char can be upper and digit
            }
            if (char.IsDigit(c))
                hasDigit = true;
        }
        return hasUpper && hasDigit;
    }
}
```



```
}
```

You will now test the above *Logon* method against a set of rules which are

userId must not be null or spaces.

password must at least be 8 characters long.

password must at least contain a digit and an upper case character.

4. First you will create a unit test class in an MS-Test project.
Right click on the Logon() method code and select "*Create Unit Tests*"

Accept the defaults in the next dialog

Replace the test case in this class with the following:

```
[TestClass()]
public class SecurityTests
{
    [TestMethod()]
    public void testLogonEmptyUserId()
    {
        string userId = "";
        string password = "Freddy99";
        Security s = new Security();
        bool actual = s.Logon(userId, password);
        bool expected = false;
        Assert.AreEqual(expected, actual);
    }
}
```

5. Select the **Tests -> Run -> All Tests** menus.
6. Copy and paste the above method and create additional tests.
7. What tests would you devise? Review the code for other criteria.
The method names must describe what the test is about.

You will not be testing the error messages.

8. Click the run button to run your tests.

**** End ****

