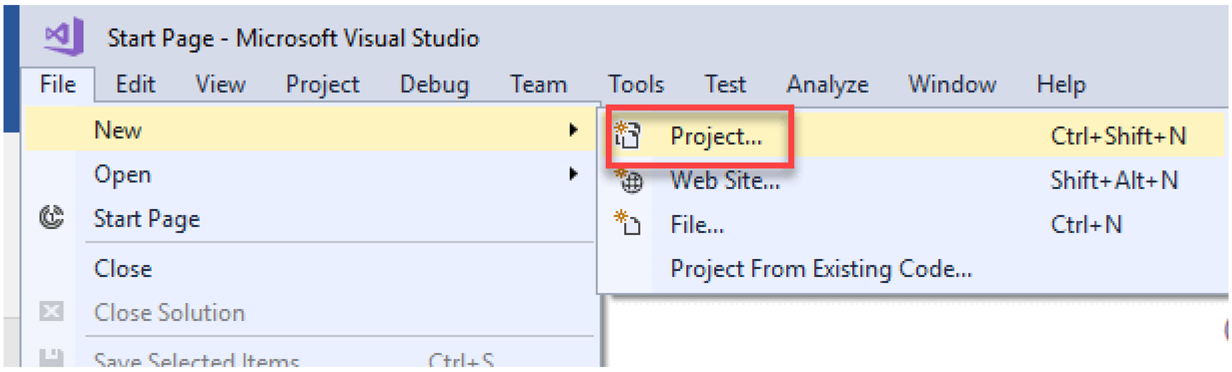
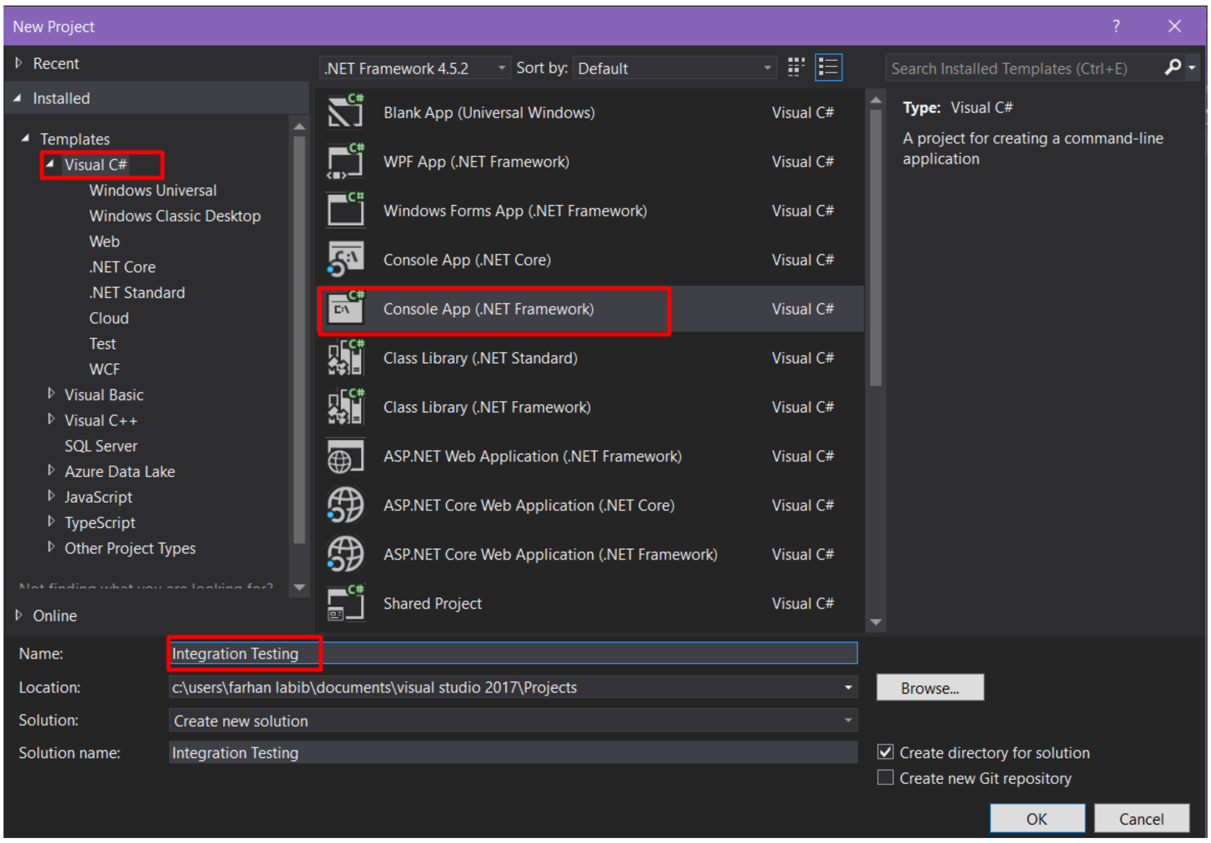
Create a new project in Visual Studio:

Step 1) In the File Menu, Click New > Project



Step 2) In the next screen,

1. Select the option 'Visual C#'
2. Click on Console App (.Net Framework)
3. Enter the name as "Desired Name"
4. Click OK



New Project
Recent
Installed
Templates
Visual C#
Windows Universal
Windows Classic Desktop
Web
.NET Core
.NET Standard
Cloud
Test
WCF
Visual Basic
Visual C++
SQL Server
Azure Data Lake
JavaScript
TypeScript
Other Project Types
Online
.NET Framework 4.5.2
Sort by: Default
Search Installed Templates (Ctrl+E)
Blank App (Universal Windows) Visual C#
WPF App (.NET Framework) Visual C#
Windows Forms App (.NET Framework) Visual C#
Console App (.NET Core) Visual C#
Console App (.NET Framework) Visual C#
Class Library (.NET Standard) Visual C#
Class Library (.NET Framework) Visual C#
ASP.NET Web Application (.NET Framework) Visual C#
ASP.NET Core Web Application (.NET Core) Visual C#
ASP.NET Core Web Application (.NET Framework) Visual C#
Shared Project Visual C#
Browse...
OK
Cancel

Type: Visual C#

A project for creating a command-line application

Name: Integration Testing

Location: c:\users\farhan labib\documents\visual studio 2017\Projects

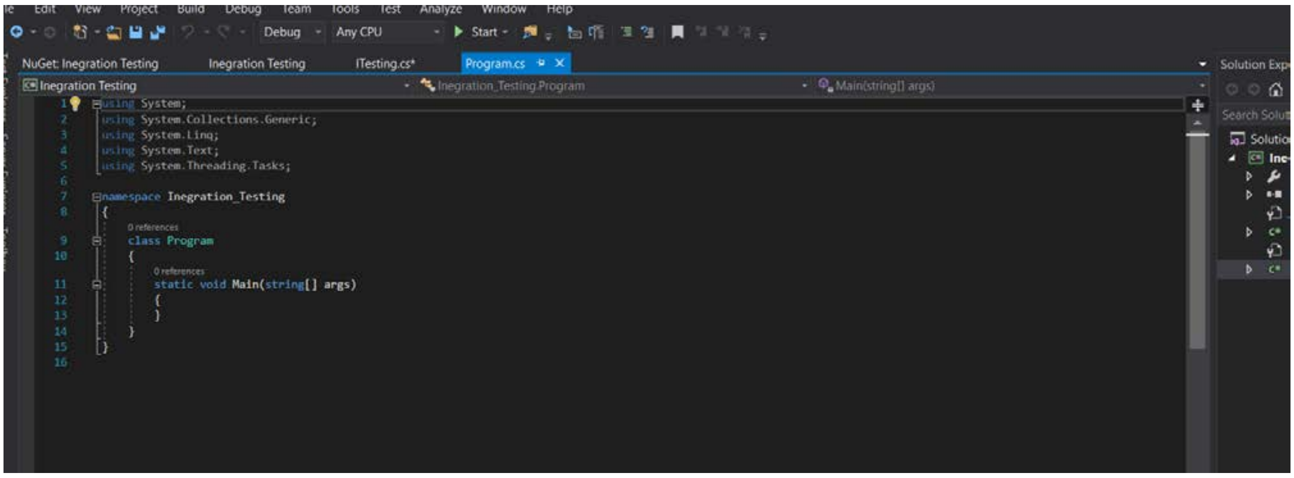
Solution: Create new solution

Solution name: Integration Testing

☑ Create directory for solution

☐ Create new Git repository

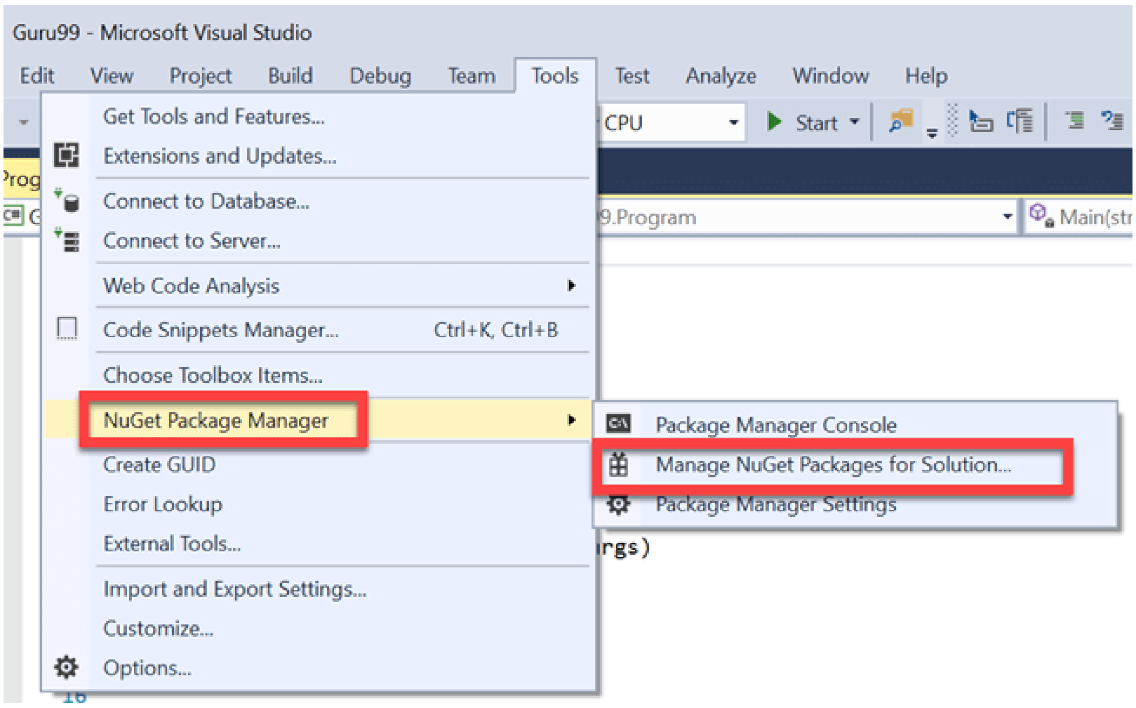
Step 3) The below screen will be displayed once the project is successfully created.



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Inegration_Testing
{
    0 references
    class Program
    {
        0 references
        static void Main(string[] args)
        {
        }
    }
}
```

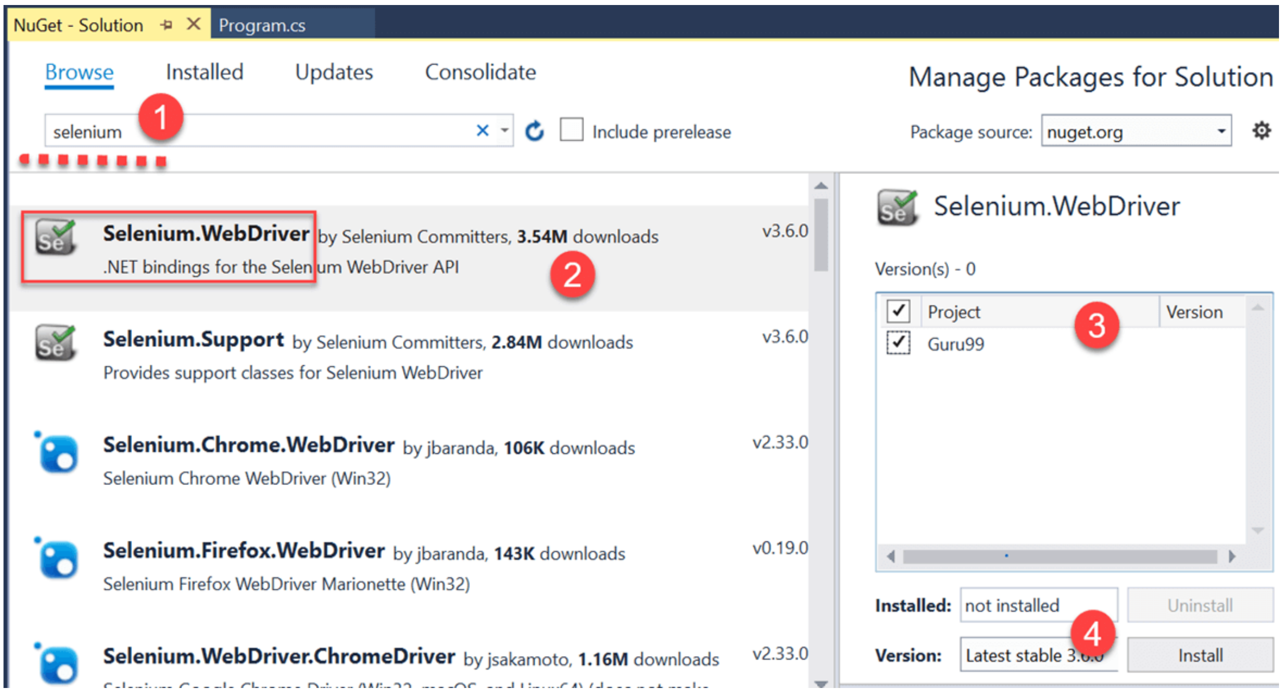
Set up Visual Studio with Selenium WebDriver:

Step 1) Navigate to Tools -> NuGet Package Manager -> Manage NuGet Packages for Solution

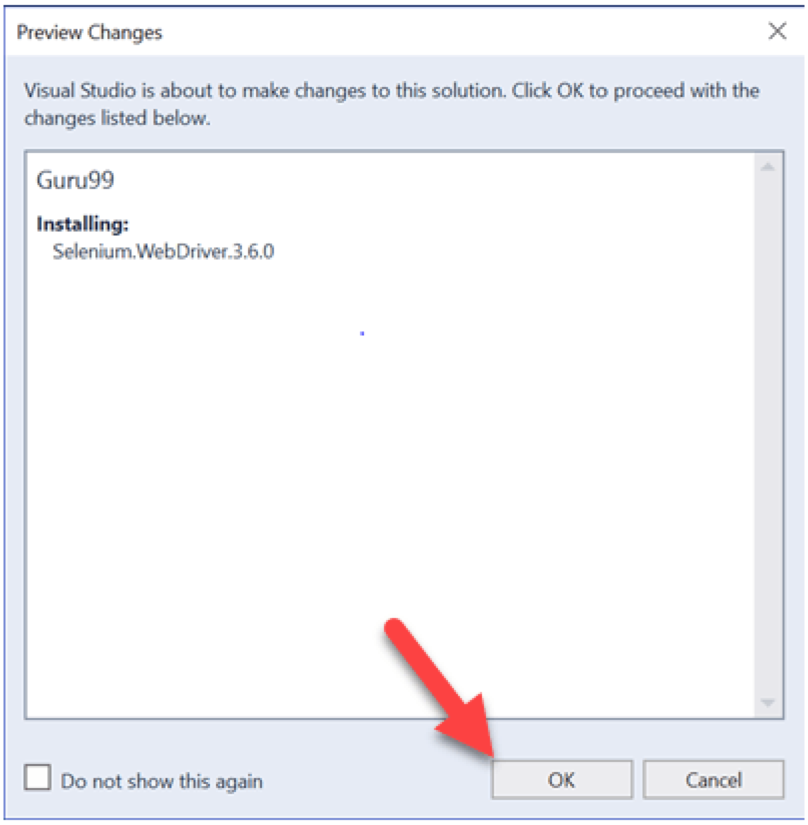

Guru99 - Microsoft Visual Studio
Edit View Project Build Debug Team Tools Test Analyze Window Help
Get Tools and Features...
Extensions and Updates...
Connect to Database...
Connect to Server...
Web Code Analysis
Code Snippets Manager... Ctrl+K, Ctrl+B
Choose Toolbox Items...
NuGet Package Manager
Create GUID
Error Lookup
External Tools...
Import and Export Settings...
Customize...
Options...
Package Manager Console
Manage NuGet Packages for Solution...
Package Manager Settings

Step 2) In the next screen

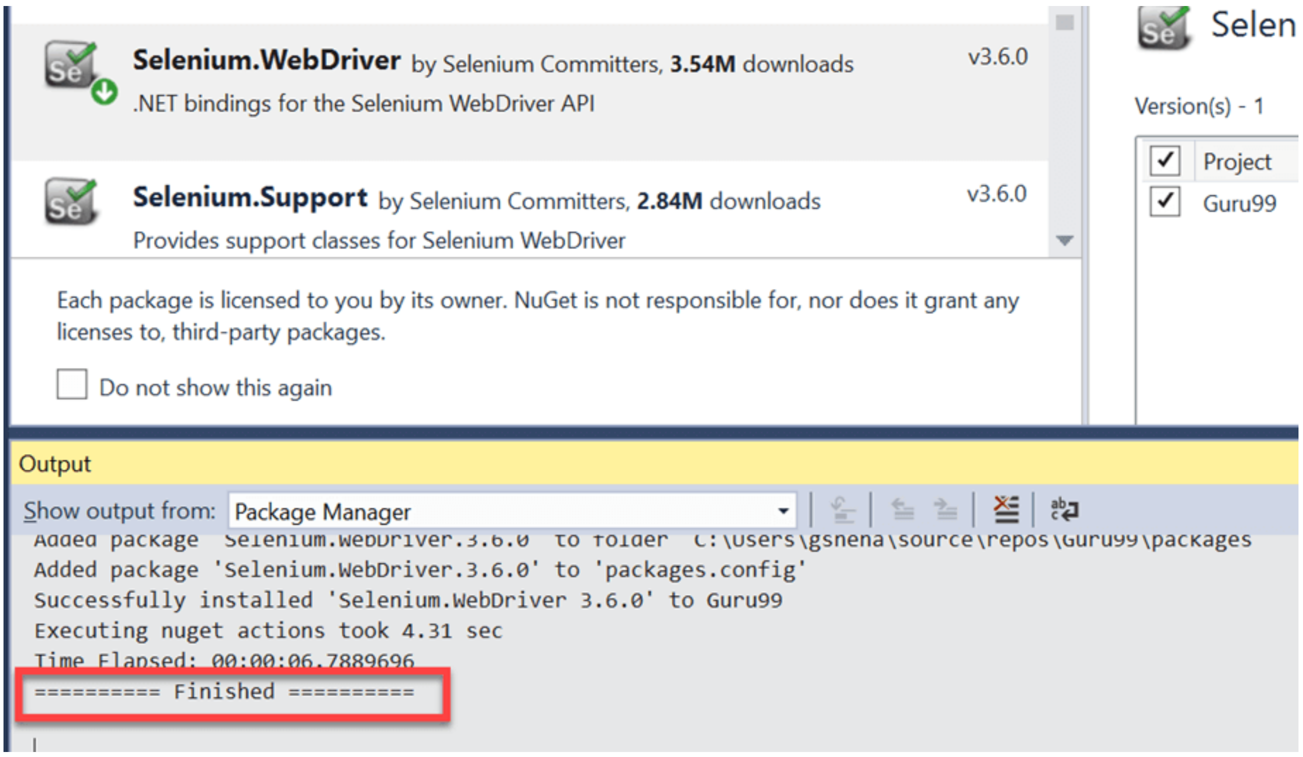
1. Search for Selenium on the resultant screen
2. Select the first search result
3. Check the project checkbox
4. Click on 'Install'


NuGet - Solution
Program.cs
Browse
Installed
Updates
Consolidate
Manage Packages for Solution
selenium
1
Include prerelease
Package source: nuget.org
Selenium.WebDriver by Selenium Committers, 3.54M downloads
.NET bindings for the Selenium WebDriver API
v3.6.0
2
Selenium.Support by Selenium Committers, 2.84M downloads
Provides support classes for Selenium WebDriver
v3.6.0
Selenium.Chrome.WebDriver by jbaranda, 106K downloads
Selenium Chrome WebDriver (Win32)
v2.33.0
Selenium.Firefox.WebDriver by jbaranda, 143K downloads
Selenium Firefox WebDriver Marionette (Win32)
v0.19.0
Selenium.WebDriver.ChromeDriver by jsakamoto, 1.16M downloads
v2.33.0
Selenium.WebDriver
Version(s) - 0
Project
Version
3
Guru99
Installed: not installed
Uninstall
4
Version: Latest stable 3.6.0
Install

Step 3) Click on 'OK' button in the pop-up screen


Preview Changes
Visual Studio is about to make changes to this solution. Click OK to proceed with the changes listed below.
Guru99
Installing:
Selenium.WebDriver.3.6.0
Do not show this again
OK
Cancel

Step 4) The below message will be displayed once the package is successfully installed.



Selenium.WebDriver by Selenium Committers, **3.54M** downloads v3.6.0
.NET bindings for the Selenium WebDriver API

Selenium.Support by Selenium Committers, **2.84M** downloads v3.6.0
Provides support classes for Selenium WebDriver

Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any licenses to, third-party packages.

☐ Do not show this again

```
Added package 'Selenium.WebDriver.3.6.0' to folder 'C:\Users\gsneha\source\repos\Guru99\packages
Added package 'Selenium.WebDriver.3.6.0' to 'packages.config'
Successfully installed 'Selenium.WebDriver 3.6.0' to Guru99
Executing nuget actions took 4.31 sec
Time Elapsed: 00:00:06.7889696
========== Finished ==========
```

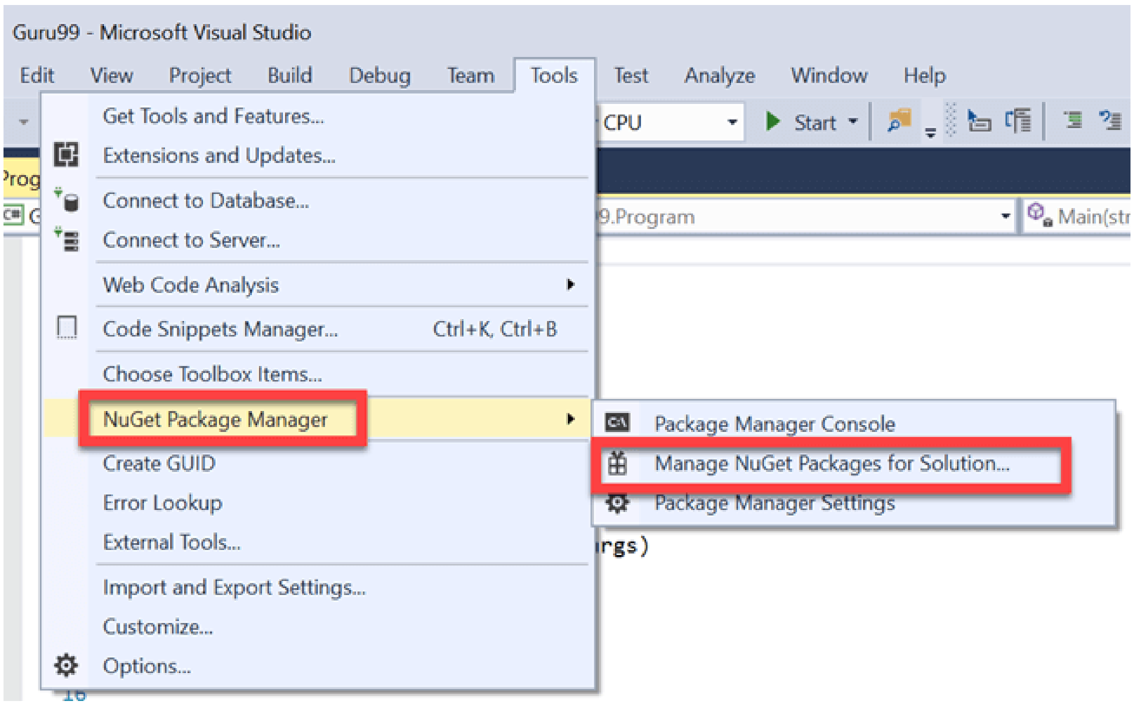
NUnit Framework: Overview

NUnit is the Unit Testing framework supported by Visual Studio and Selenium WebDriver. NUnit is the most widely used Unit Testing framework for .Net applications. NUnit presents the test results in a readable format and allows a tester to debug the automated tests.

We need to install NUnit Framework and NUnit Test Adapter onto Visual Studio inorder to use it.

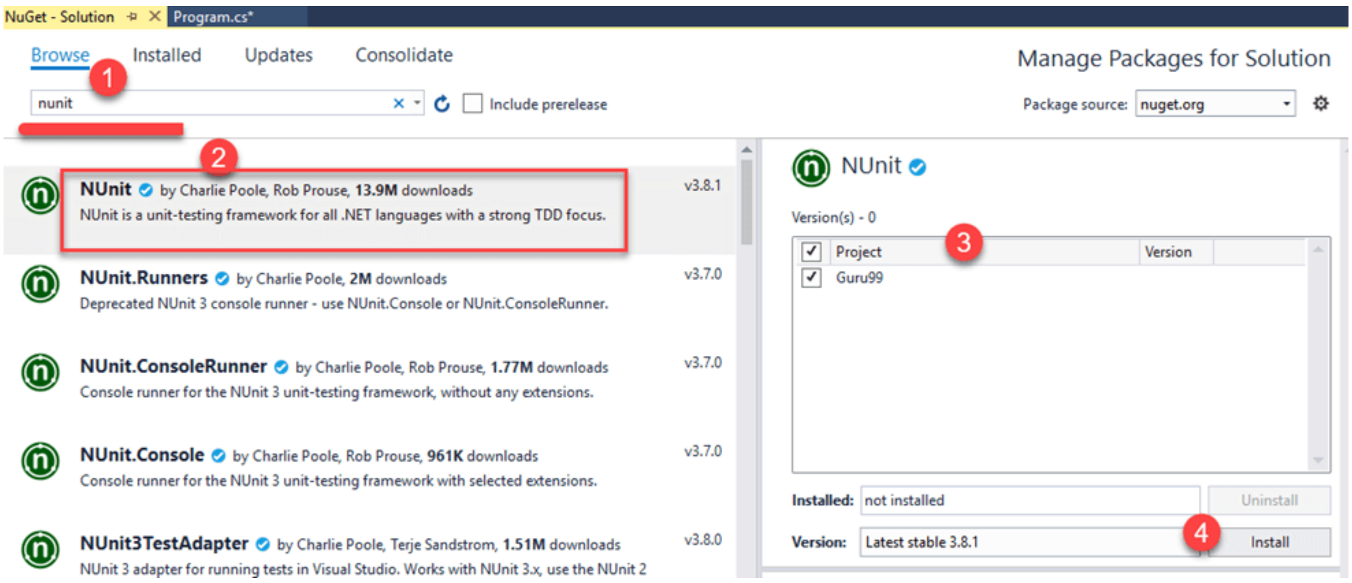
Steps to install NUnit Framework:

1. Navigate to Tools -> NuGet Package Manager -> Manage NuGet Packages for Solution


Guru99 - Microsoft Visual Studio
Edit View Project Build Debug Team Tools Test Analyze Window Help
Get Tools and Features...
Extensions and Updates...
Connect to Database...
Connect to Server...
Web Code Analysis
Code Snippets Manager... Ctrl+K, Ctrl+B
Choose Toolbox Items...
NuGet Package Manager
Create GUID
Error Lookup
External Tools...
Import and Export Settings...
Customize...
Options...
Package Manager Console
Manage NuGet Packages for Solution...
Package Manager Settings

Step 2) In the next window

1. Search for NUnit
2. Select the search result
3. Select Project
4. Click Install



NUnit by Charlie Poole, Rob Prouse, 13.9M downloads v3.8.1
NUnit is a unit-testing framework for all .NET languages with a strong TDD focus.

NUnit.Runners by Charlie Poole, 2M downloads v3.7.0
Deprecated NUnit 3 console runner - use NUnit.Console or NUnit.ConsoleRunner.

NUnit.ConsoleRunner by Charlie Poole, Rob Prouse, 1.77M downloads v3.7.0
Console runner for the NUnit 3 unit-testing framework, without any extensions.

NUnit.Console by Charlie Poole, Rob Prouse, 961K downloads v3.7.0
Console runner for the NUnit 3 unit-testing framework with selected extensions.

NUnit3TestAdapter by Charlie Poole, Terje Sandstrom, 1.51M downloads v3.8.0
NUnit 3 adapter for running tests in Visual Studio. Works with NUnit 3.x, use the NUnit 2

NUnit

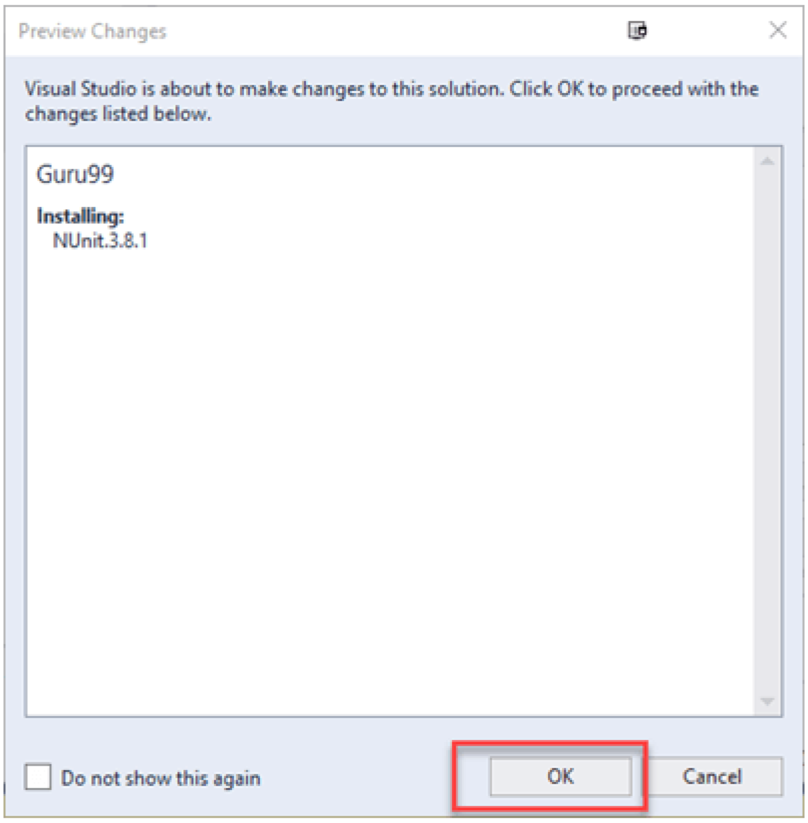
Version(s) - 0

	Project	Version
✓	Guru99	

Installed: not installed

Version: Latest stable 3.8.1

Step 3) The below popup will appear. Click on 'Ok' button.

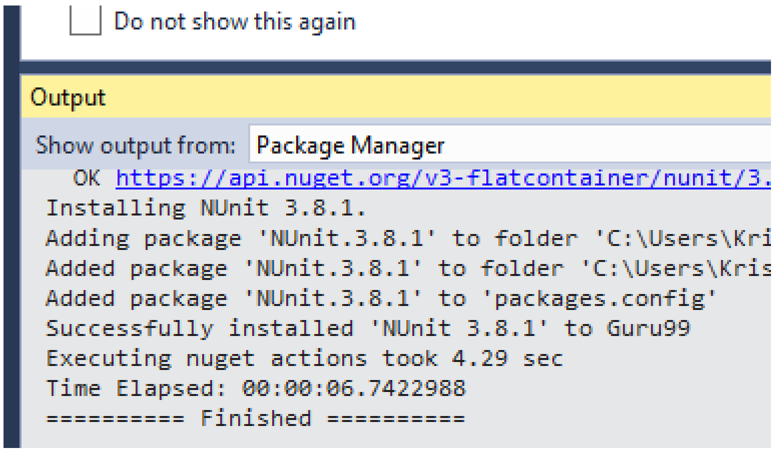


Visual Studio is about to make changes to this solution. Click OK to proceed with the changes listed below.

Guru99

Installing:
NUnit.3.8.1

Step 4) The below message will appear once the installation is complete.



Show output from: Package Manager

```
  OK https://api.nuget.org/v3-flatcontainer/nunit/3.
Installing NUnit 3.8.1.
Adding package 'NUnit.3.8.1' to folder 'C:\Users\Kri
Added package 'NUnit.3.8.1' to folder 'C:\Users\Kris
Added package 'NUnit.3.8.1' to 'packages.config'
Successfully installed 'NUnit 3.8.1' to Guru99
Executing nuget actions took 4.29 sec
Time Elapsed: 00:00:06.7422988
========== Finished ==========
```

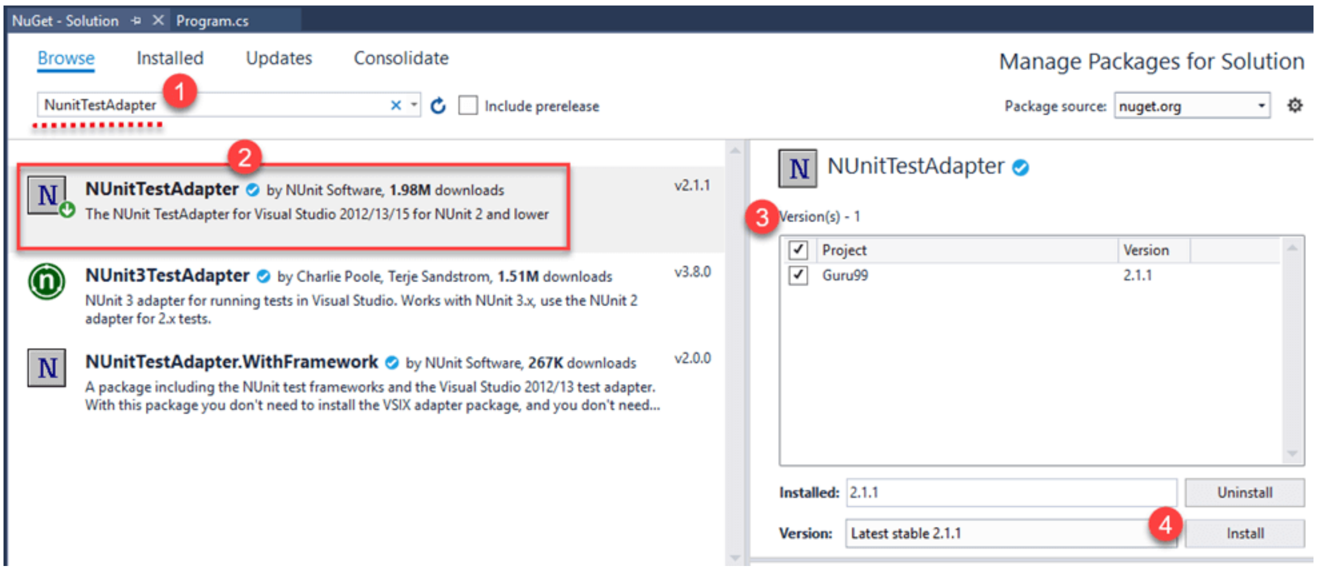
Steps to download NUnit Test Adapter

Please note that the below steps work only for 32-bit machines. For 64-bit machines, you need to download the 'NUnit3 Test Adapter' by following the same process as mentioned below.

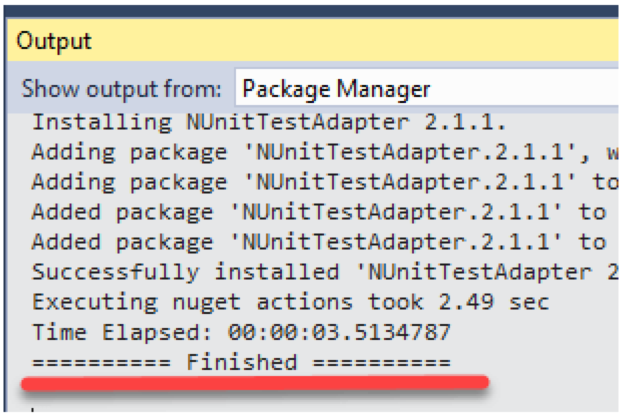
Step 1) Navigate to Tools ->NuGet Package Manager -> Manage NuGet Packages for Solution. In that screen,

1. Search NUnitTestAdapter
2. Click Search Result
3. Select Project

4. Click Install


NuGet - Solution
Program.cs
Browse
Installed
Updates
Consolidate
Manage Packages for Solution
NunitTestAdapter
1
Include prerelease
Package source: nuget.org
2
NUnitTestAdapter by NUnit Software, 1.98M downloads
The NUnit TestAdapter for Visual Studio 2012/13/15 for NUnit 2 and lower
v2.1.1
NUnit3TestAdapter by Charlie Poole, Terje Sandstrom, 1.51M downloads
NUnit 3 adapter for running tests in Visual Studio. Works with NUnit 3.x, use the NUnit 2 adapter for 2.x tests.
v3.8.0
NUnitTestAdapter.WithFramework by NUnit Software, 267K downloads
A package including the NUnit test frameworks and the Visual Studio 2012/13 test adapter. With this package you don't need to install the VSIX adapter package, and you don't need...
v2.0.0
NUnitTestAdapter
3
Version(s) - 1
Project
Version
Guru99
2.1.1
Installed: 2.1.1
Uninstall
Version: Latest stable 2.1.1
4
Install

Step 2) Click OK on the confirmation pop-up. Once install is done you will see the following message-



```
Installing NUnitTestAdapter 2.1.1.
Adding package 'NUnitTestAdapter.2.1.1', w
Adding package 'NUnitTestAdapter.2.1.1' to
Added package 'NUnitTestAdapter.2.1.1' to
Added package 'NUnitTestAdapter.2.1.1' to
Successfully installed 'NUnitTestAdapter 2
Executing nuget actions took 2.49 sec
Time Elapsed: 00:00:03.5134787
========== Finished ==========
```

Selenium and NUnit framework:

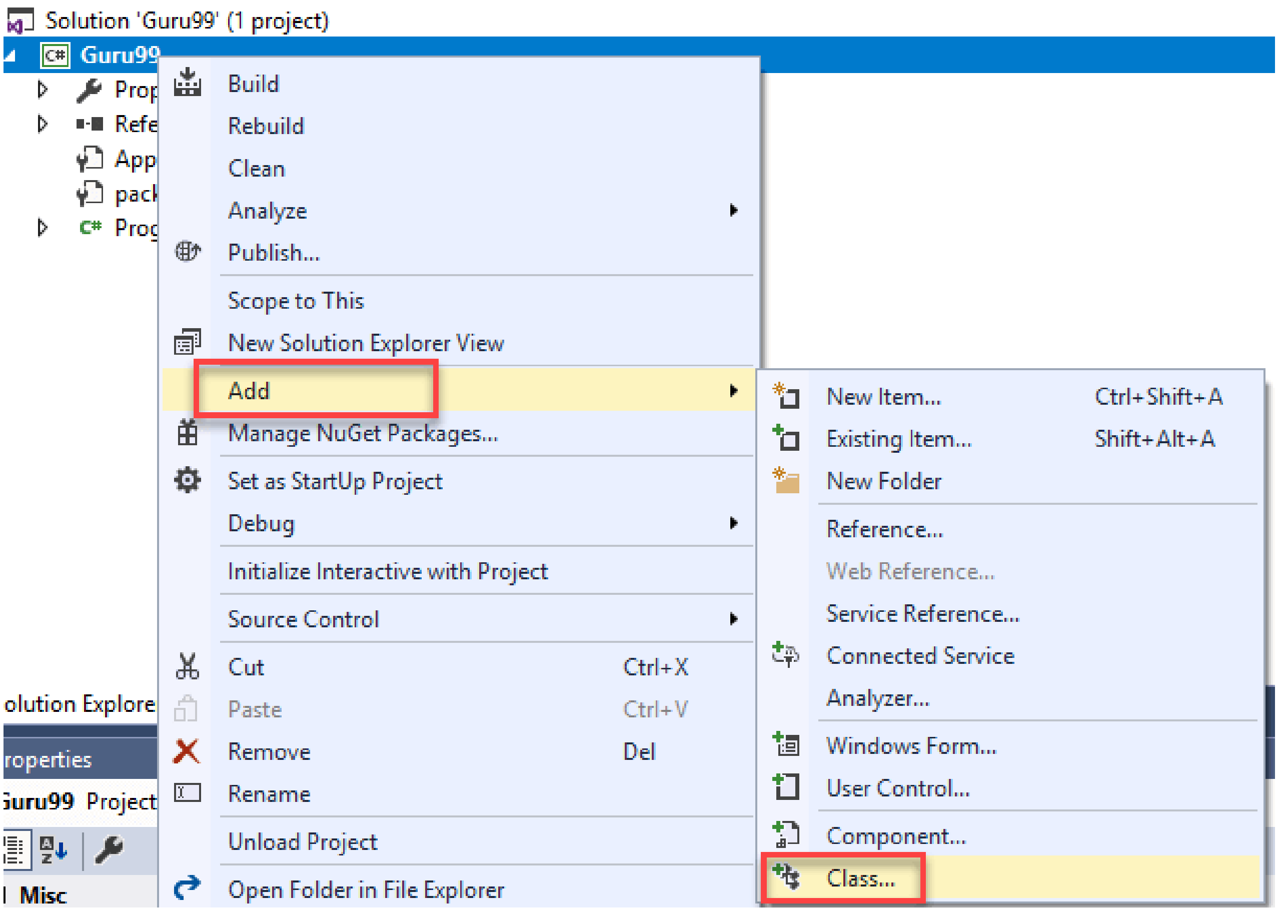
Integration of selenium with NUnit framework allows a tester to differentiate between various test classes. NUnit also allows testers to use annotations such as SetUp, Test, and TearDown to perform actions before and after running the test.

NUnit framework can be integrated with Selenium by creating a NUnit test class and running the test class using NUnit framework.

The below are the steps needed to create and run a test class using NUnit framework.

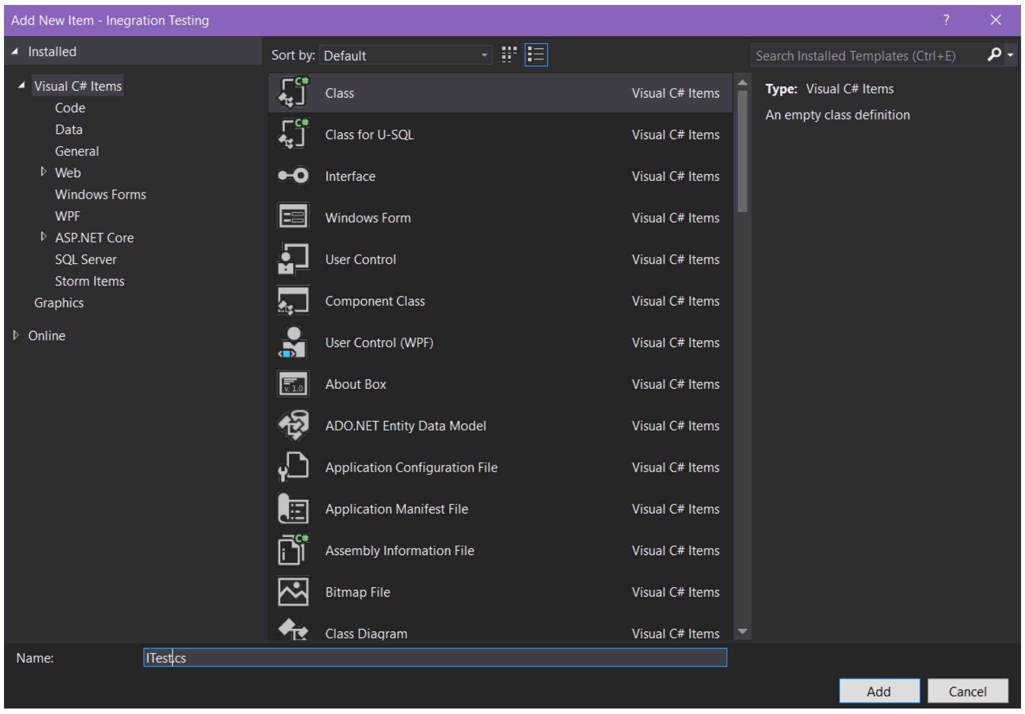
Steps to create a NUnit Test class in Selenium:

Step 1) In the Solution Explorer, Right click on project > Add > Class



Step 2) Class creation window will appear.

1. Provide a name to the class
2. Click on Add button


Add New Item - Inegration Testing

Installed

Visual C# Items: Code, Data, General, Web, Windows Forms, WPF, ASP.NET Core, SQL Server, Storm Items

Graphics

Online

Sort by: Default

Search Installed Templates (Ctrl+E)

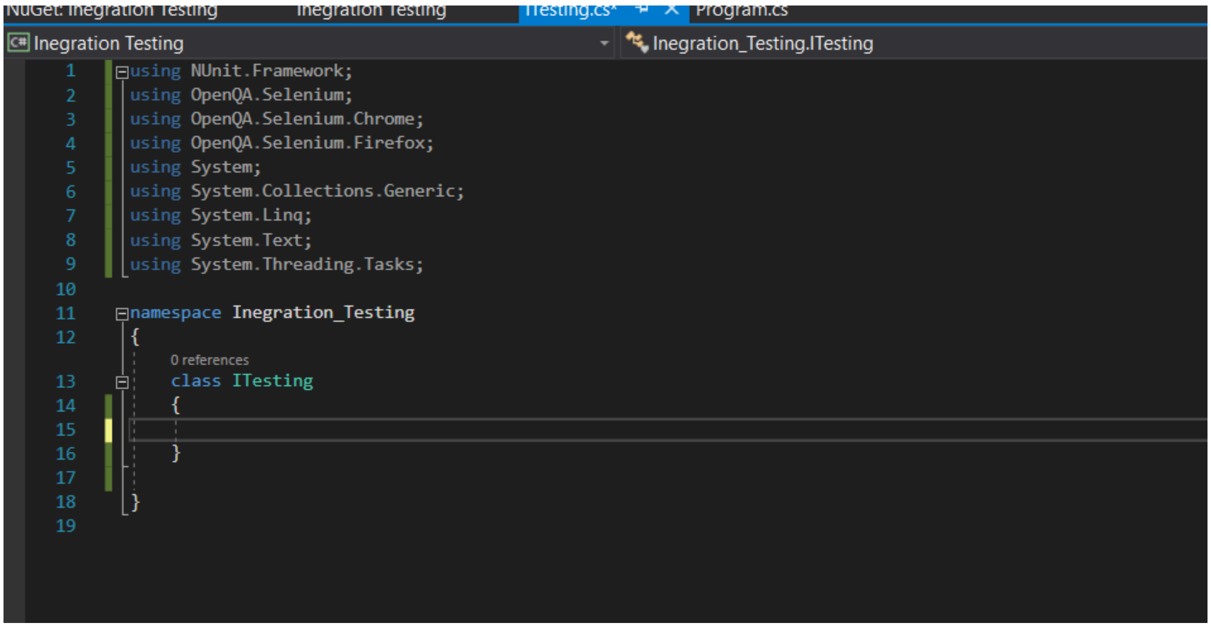
Class	Visual C# Items
Class for U-SQL	Visual C# Items
Interface	Visual C# Items
Windows Form	Visual C# Items
User Control	Visual C# Items
Component Class	Visual C# Items
User Control (WPF)	Visual C# Items
About Box	Visual C# Items
ADO.NET Entity Data Model	Visual C# Items
Application Configuration File	Visual C# Items
Application Manifest File	Visual C# Items
Assembly Information File	Visual C# Items
Bitmap File	Visual C# Items
Class Diagram	Visual C# Items

Type: Visual C# Items

An empty class definition

Name: ITest.cs

Step 3) The below screen will appear.



```
using NUnit.Framework;
using OpenQA.Selenium;
using OpenQA.Selenium.Chrome;
using OpenQA.Selenium.Firefox;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Inegration_Testing
{
    class ITesting
    {

    }

}
```

Step 4) Add the following code to the created class. Please note that you need to specify the location of 'chromdriver.exe' file during chrome driver initialization.

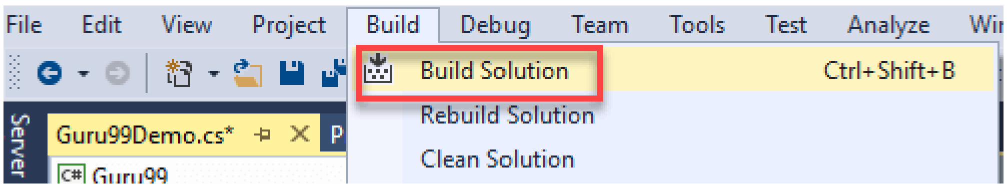
```
using NUnit.Framework;
using OpenQA.Selenium;
using OpenQA.Selenium.Chrome;
using OpenQA.Selenium.Firefox;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Inegration_Testing
{
    class ITesting
    {
        IWebDriver driver;

        [SetUp]
        public void startBrowser()
        {
            driver = new ChromeDriver();
        }

        [Test]
        public void test()
        {
            driver.Url = "http://result.ewubd.edu";
            IWebElement element = driver.FindElement(By.Name("txtstudid"));
            element.SendKeys("Your Id");

            IWebElement password = driver.FindElement(By.Name("txtpass"));
            password.SendKeys("Your Password");
            driver.FindElement(By.Id("studlogin")).Click();

            String at = driver.Title;
            String et = "East West University";

            if (at == et)
            {
                Console.WriteLine("Test Successful");
                IWebElement element2 = 
driver.FindElement(By.XPath("/html/body/table/tbody/tr[4]/td/div/div/div[2]/a"));
                element2.Click();
            }
            else
            {
                Console.WriteLine("Unsuccessful");
            }

        }


        [TearDown]
        public void closeBrowser()
        {
            driver.Close();
        }
    }

}
```

[How to find Xpath: https://www.youtube.com/watch?v=tV7ziQCIZAg]

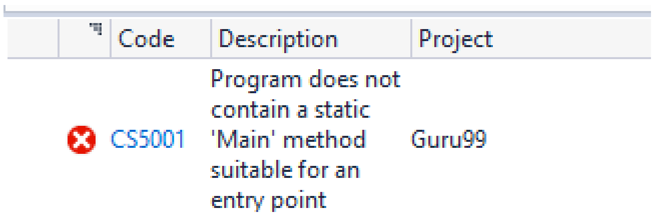
##Code Explanation:

Here we try to login to our university's Result site and download the PDF file. The explanation of these commands used in this code is given below.

Step 4) Click on 'Build' -> 'Build Solution'

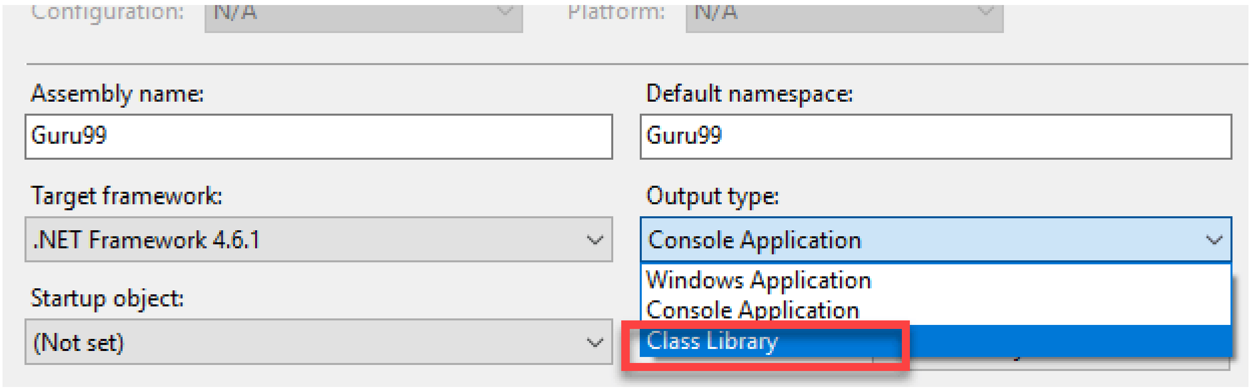


NOTE: You may get an error like "Does not contain a static 'main' method suitable for an entry point" when you build



Code	Description	Project
CS5001	Program does not contain a static 'Main' method suitable for an entry point	Guru99

To resolve this Got to Project > Properties and change Output Type to "Class Library." The default is "Console Application."



Configuration: N/A Platform: N/A

Assembly name: Guru99

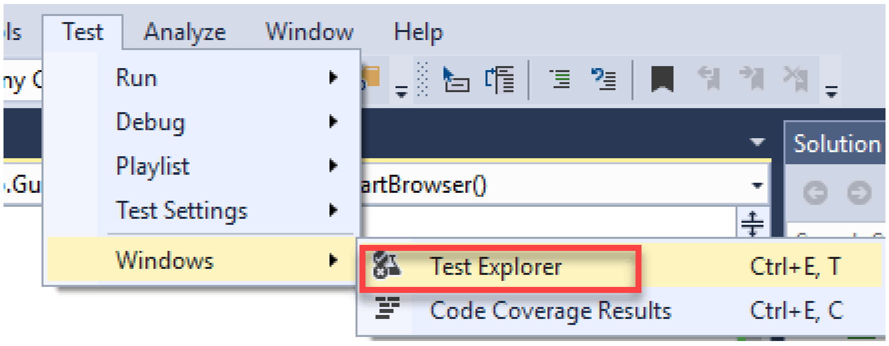
Default namespace: Guru99

Target framework: .NET Framework 4.6.1

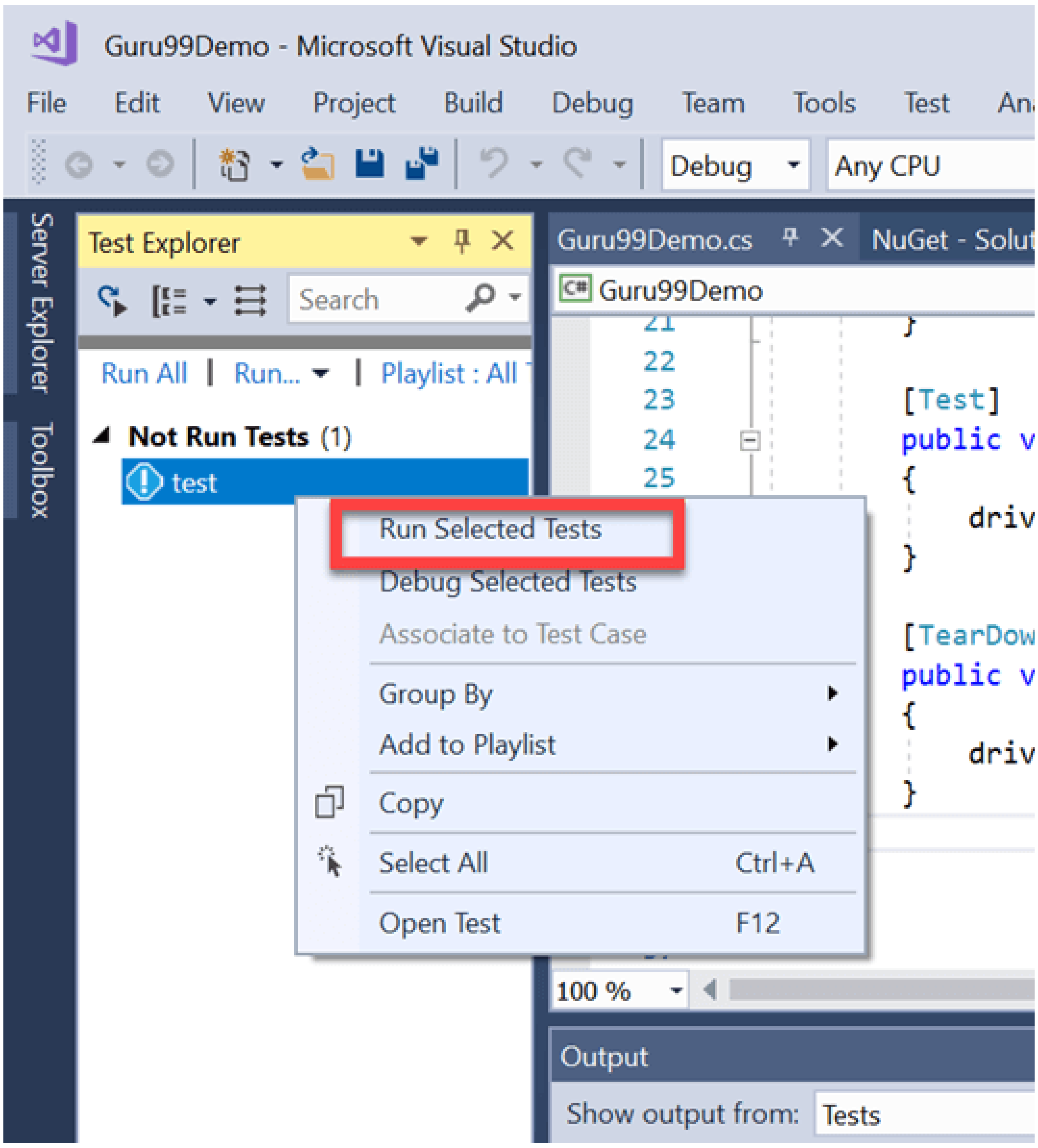
Output type: Console Application

Startup object: (Not set)

Step 5) Once the build is successful, we need to open the Test Explorer window. Click on Test -> Windows -> Test Explorer

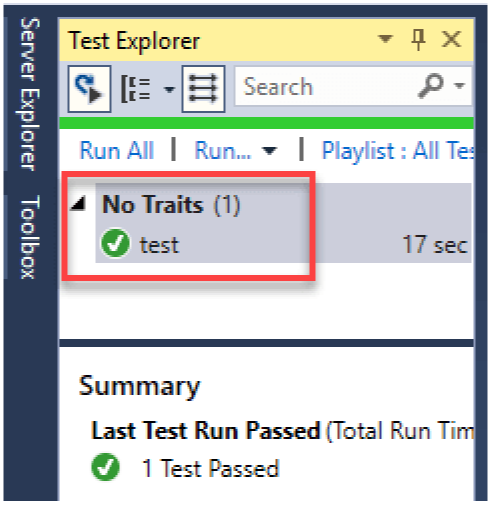


Step 6) Test Explorer window opens with the list of available tests. Right-click on Test Explorer and select Run Selected Tests



```
[Test]
public v
{
    driv
}

[TearDow
public v
{
    driv
}
```

Step 7) Selenium must open the browser with the specified URL and Login to the website and download the PDF then close the browser. Test case status will be changed to 'Pass' on the Test Explorer window.



Selenium WebDriver Commands in C#:

C# uses the interface 'IWebDriver' for browser interactions. The following are the category of commands available in C#.

1. Browser commands
2. Web Element commands
3. Dropdown commands

Browser commands: [1]

The following are the list of browser commands available in C#.

Command Name	**Description**	**Syntax**
Url Command	This command is used to open a specified URL in the browser.	`driver.Url = "https://www.guru99.com"`
Title Command	This command is used to retrieve the page title of the web page that is currently open	`String title = driver.Title`
PageSource Command	This command is used to retrieve the source code of web page that is currently open.	`String pageSource = driver.PageSource`
Close Command	This command is used to close the recently opened browser instance.	`driver.Close();`
Quit Command	This command is used to close all open browser instances	`driver.Quit();`
Back Command	This command is used to navigate to the previous page of browser history.	`driver.Navigate().Back();`

Forward Command	This command is used to navigate to the next page of browser history.	`driver.Navigate().Forward()`
Refresh Command	This command is used to perform browser refresh.	`driver.Navigate().Refresh()`

Webelement Commands:

A Webelement represents all the elements on a web page. They are represented by HTML tags. Each of the buttons, textboxes, links, images, tables, and frames fall under Webelements. Operations on web elements can be triggered using the IWebelement interface. To interact with a Webelement, we need to find the element on the webpage and then perform operations on it. Tools like Firebug and Firepath can be used to identify the Xpath of Webelement.

The following are the list of Webelement commands available in C#.

Command Name	**Description**	**Syntax**
Click command	This command is used to click on a Webelement. For the element to be clickable, the element must be visible on the webpage. This command is used for checkbox and radio button operations as well.	`IWebelement element = driver.FindElement(By.xpath("xpath of Webelement"));  element.Click();`

Clear command	This command is specifically used for clearing the existing contents of textboxes.	`IWebelement element = driver.FindElement(By.xpath("xpath of Webelement"));`
		`element.Clear();`

SendKeys command	This command is used to input a value onto text boxes. The value to be entered must be passed as a parameter to	`IWebelement element = driver.FindElement(By.xpath("xpath of Webelement"));`
		`element.SendKeys("guru99");`

Displayed command	This command is used to identify if a specific element is displayed on the webpage. This command returns a Boolean value; true or false depending on the visibility of web element.	`IWebelement element = driver.FindElement(By.xpath("xpath of Webelement"));`
		`Boolean status = element.Displayed;`

Enabled command	This command is used to identify if a particular web element is enabled on the web page. This command	`IWebelement element = driver.FindElement(By.xpath("xpath of Webelement"));`
		`Boolean status = element.Enabled;`

	returns a Boolean value; true or false as a result.	

Selected command	This command is used to identify if a particular web element is selected. This command is used for checkboxes,radio buttons, and select operations.	`IWebelement element = driver.FindElement(By.xpath("xpath of Webelement"));`
		`Boolean status = element.Selected;`

Submit command:	This command is similar to click command, The difference lies in whether the HTML form has a button with the type Submit. While the click command clicks on any button, submit command clicks on the only the buttons with type submit.	`IWebelement element = driver.FindElement(By.xpath("xpath of Webelement"));  element.submit();`

Text command	This command returns the innertext of a Webelement. This command returns a string value as a result.	`IWebelement element = driver.FindElement(By.xpath("xpath of Webelement"));`
		`String text=element.Text;`

TagName command	This command returns the HTML tag of a web element. It returns a string value as the result.	`IWebelement element = driver.FindElement(By.xpath("xpath of Webelement"));  String tagName = element.TagName;`

GetCSSValue Command:	This method is used to return the color of a web element on the form of a rgba string (Red,Green,Blue, and Alpha).	`IWebelement element = driver.FindElement(By.xpath("xpath of Webelement"));`
		`String color = element.getCSSValue;`
		
		Output- If the color of element is red, output would be rgba(255,0,0,1)

Dropdown Commands:

Dropdown operations in C# can be achieved using the SelectElement class.

The following are the various dropdown operations available in C#.

Command Name	Description	Syntax

SelectByText Command	This command selects an option of a dropdown based on the text of the option.	`IWebelement element = driver.FindElement(By.xpath("xpath of Webelement"));`
		`SelectElement select = new SeectElement(element);`
		`select.SelectByText("Guru99");`

SelectByIndex Command	This command is used to select an option based on its index. Index of dropdown starts at 0.	`IWebelement element = driver.FindElement(By.xpath("xpath of Webelement"));` `SelectElement select = new SelectElement(element);` `select.SelectByIndex("4");`
SelectByValue Command	This command is used to select an option based on its option value.	`IWebelement element = driver.FindElement(By.xpath("xpath of Webelement"));` `SelectElement select = new SelectElement(element);` `select.SelectByValue("Guru99");`
Options Command	This command is used to retrieve the list of options displayed in a dropdown.	`IWebelement element = driver.FindElement(By.xpath("xpath of Webelement"));` `SelectElement select = new SelectElement(element);` `List<IWebelement> options = select. Options;` `int size = options.Count;` `for(int i=0;i<options.size();i++)` `{` `String value = size.elementAt(i).Text;` `Console.writeLine(value);` `}` The above code prints all the options onto console within a dropdown.
IsMultiple command	This command is used to identify if a dropdown is a multi select dropdown; A multi select dropdown enables user to select more than one option in a dropdown at a time. This command returns a Boolean value.	`IWebelement element = driver.FindElement(By.xpath("xpath of Webelement"));` `SelectElement select = new SelectElement(element);` `Boolean status = select.IsMultiple();`
DeSelectAll command	This command is used in multi select dropdowns. It clears the options that have already been selected.	`IWebelement element = driver.FindElement(By.xpath("xpath of Webelement"));` `SelectElement select = new SelectElement(element);` `select.DeSelectAll();`
DeSelectByIndex command	This command deselects an already selected value using its index.	`IWebelement element = driver.FindElement(By.xpath("xpath of Webelement"));` `SelectElement select = new SelectElement(element);` `select.DeSelectByIndex("4");`
DeSelectByValue command	This command deselects an already selected value using its value.	`IWebelement element = driver.FindElement(By.xpath("xpath of Webelement"));` `SelectElement select = new SelectElement(element);` `select.DeSelectByValue("Guru99");`

DeSelectByText command	This command deselects an already selected value using its text.	`IWebelement element = driver.FindElement(By.xpath("xpath of Webelement"));`
		`SelectElement select = new SelectElement(element);`
		`select.DeSelectByText("Guru99");`

Reference:

1. Guru99.com