

## Project #5 – URL Downloading Console Application

**DUE DATE: 3/31/20 – End of Day**

### Description:

Your IT department needs an application to assist them with the daily download of specific website data. They would like you to create a console application they can configure as required to download any number of URLs.

**Skills Used:** use inheritance, polymorphism, interfaces and abstract classes, utilize unit testing, recognize and apply appropriate data structures

### User Experience:

1. The user opens up a command prompt and navigates to the location of the executable on disk
2. The user types the name of the executable and the application downloads the URLs into the app.config file
3. URLs downloaded successfully are saved to a file with the configured 'name' attribute in the app.config
4. URLs that fail to download simply write a message to the console using Console.WriteLine indicating the error
5. Subsequent runs of the application will save over any previously downloaded files

### Project Instructions

1. Clone or download the repository for this project to your local machine. If you have troubles retrieving the project from the repository URL, please contact the instructor for assistance.

**GITHUB PROJECT URL:** <https://github.com/gottjl01/cs203-project5-cs>

**GITHUB REPOSITORY URL:** <https://github.com/gottjl01/cs203-project5-cs.git>

2. You will need an internet connection to restore the nuget packages in this project.
3. You must write your own unit tests for this project. You will need to write a minimum of 4 unit tests covering anything you feel is appropriate. The unit tests you write must be able to pass and actually test some aspect of the application, such as loading valid or missing configuration sections. The skeleton of the unit testing classes are provided, but not implemented. You may reference previous projects for examples on how to configure and define your own unit tests.
4. The **PFW.CSIST203.Project5** code files in this solution must be implemented as follows:
  - a. DownloadConfiguration.cs: This configuration class must have a single System.Configuration.ConfigurationProperty ReadOnly property named "Urls" that uses the Items("Urls") dictionary object for the ConfigurationSection class
  - b. Url.cs: This class must have ConfigurationProperty ReadOnly Properties added for 'name' and 'url', which use the internal Items dictionary of the base ConfigurationElement class
  - c. Urls.cs: The CreateNewElement() method must create a new Url object and the GetElementKey must cast the element to a Url object type and return the 'name' property
  - d. DownloadConfigurationTests.cs: Add test(s) to this class to verify loading the configuration sections provided below do not cause the application to throw an exception
  - e. DownloaderTests.cs: Add test(s) to this class to verify that the application does not crash when trying to download the target URLs and is saves the resulting data from the URL into a physical file on disk with the name specified in the 'name' attribute for the given entry.
  - f. No other files in this project require modification, nor should they be modified.

5. The following configuration sections represent the tests that will be performed with your application:

- a. `<PFW.CSIST203.Project5.Config.DownloadConfiguration>`  
    `<Urls>`  
        `<Url name="bad-1.html" url="http://asdfbogus23322.com/whatever" />`  
        `<Url name="bad-2.html" url="https://jakkww33399x0.net/data" />`  
    `</Urls>`  
    `</PFW.CSIST203.Project5.Config.DownloadConfiguration>`
- b. `<PFW.CSIST203.Project5.Config.DownloadConfiguration>`  
    `<Urls>`  
        `<Url name="search-1.html" url="https://google.com" />`  
        `<Url name="search-2.html" url="https://bing.com" />`  
        `<Url name="search-3.html" url="https://yahoo.com" />`  
    `</Urls>`  
    `</PFW.CSIST203.Project5.Config.DownloadConfiguration>`

## Code Documentation

Code comments are meant to provide a brief explanation in areas where the code is less self-explanatory. I ask that you use common sense and critically think about the places a reader of your code might need some guidance. In this project, I have provided many comments in the areas that are pre-implemented that you have been instructed not to modify. However, any method, field or property that you create or modify must have comments added using the standard IDE behavior in the Visual Studio IDE by typing three slashes (///) above a method, property or field. This will cause the IDE to automatically create the commenting structure that you will need to fill in. All parameters and return information must be completed.

## Submission

Zip your assignment, including all source, project and solution file(s) and submit the archive through Blackboard by the due date. Late projects will be accepted with a one-week grace period, but no later barring extenuating circumstances, which must be communicated and approved by the professor.

## Grading

Your grade is determined by the following rubric:

Scoring Rubric	
Assignment Task	Points
Application Downloads URLs and saves them with correct name	20
Unit Tests are defined, implemented and succeed	20
Class, Method and Property Documentation	10
<b>TOTAL</b>	<b>50</b>

## Extra Credit

You may earn additional credit on this project by implementing the same project using VB.NET. You can find the online github repos of the VB.NET version of these projects at the following URLs:

**GITHUB PROJECT URL:** <https://github.com/gottjl01/cs203-project5-vb>

**GITHUB REPOSITORY URL:** <https://github.com/gottjl01/cs203-project5-vb.git>

Please include the VB.NET solution with your normal project solution. It is best to place the two solutions in different folders and ZIP them up together for the Blackboard submission.

**NOTE:** The purpose of the extra credit is to encourage you to learn how to convert C# code into VB.NET code using the .NET Framework. DO NOT utilize any online or downloaded tools to automate this process.