```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Net;
5  using System.Net.Http;
6  using System.Text;
7  using System.Threading.Tasks;
8
9  namespace ConsoleApp2
10 {
11     public class WebsiteDataModel
12     {
13         public string WebsiteUrl { get; set; } = "";
14         public string WebsiteData { get; set; } = "";
15     }
16     internal class Bacon { }
17     class Program
18     {
19
20         static async Task Main(string[] args)
21         {
22             Console.WriteLine("DownloadWebsiteAsync_WithReturnValue >> ");
23             WebsiteDataModel task = await
                 DownloadWebsiteAsync_WithReturnValue();
24             Console.WriteLine(task.WebsiteUrl);
25
26             Console.WriteLine("DownloadWebsiteAsync_WithoutReturnValue >> ");
27             await DownloadWebsiteAsync_WithoutReturnValue();
28
29             Bacon obj = await FryBaconAsync(2);
30
31             Console.ReadLine();
32         }
33
34
35         private static async
             Task<WebsiteDataModel>DownloadWebsiteAsync_WithReturnValue()
36         {
37             WebsiteDataModel output = new WebsiteDataModel();
38             WebClient client = new WebClient();
39             output.WebsiteUrl = "https://www.yahoo.com";
40             output.WebsiteData = await client.DownloadStringTaskAsync
                 ("https://www.yahoo.com");
41             return output;
42         }
43
44
45         private static async Task DownloadWebsiteAsync_WithoutReturnValue()
46         {
47             WebsiteDataModel output = new WebsiteDataModel();
48             WebClient client = new WebClient();
49             output.WebsiteUrl = "https://www.yahoo.com";
50             output.WebsiteData = await client.DownloadStringTaskAsync
```

```
                ("https://www.yahoo.com");
        }


        private static async Task<Bacon> FryBaconAsync(int slices)
        {
            Console.WriteLine($"putting {slices} slices of bacon in the
              pan");
            Console.WriteLine("cooking first side of bacon...");
            await Task.Delay(1000);
            for (int slice = 0; slice < slices; slice++)
            {
                Console.WriteLine("flipping a slice of bacon");
            }
            Console.WriteLine("cooking the second side of bacon...");
            await Task.Delay(1000);
            Console.WriteLine("Put bacon on plate");

            return new Bacon();
        }


        public async Task<int> GetUrlContentLengthAsync()
        {
            var client = new HttpClient();
            Task<string> getStringTask =client.GetStringAsync("https://
              docs.microsoft.com/dotnet");
            DoIndependentWork();
            string contents = await getStringTask;
            return contents.Length;
        }

        void DoIndependentWork()
        {
            Console.WriteLine("Working...");
        }
    }


}

```