**CSCI 1630 – C# Programming I**

**Student Preparation for Lab 12**

**Lab 12 Objective**: Lab 12 will build **a Windows Form Application** which will collect currency rates data from a RESTful service residing on the cloud. The form will implement a currency calculator to compute the amount of foreign currency which can be purchased with a given amount of US dollars. For example, how many Chilean Pesos will buy $US 10.00 (you guessed it, a lots of pesos).

Access to RESTful cloud services is commonly performed via the HTPP protocol. Cloud services maintain updated data. A service consumer app can retrieve data from the service (and depending on the service itself, apps are allowed to add, modify and delete data).

In general, RESTful services exchange data with apps (e.g., our Lab 12 app) in two main formats:

* JSON (JavaScript Object Notation) – the most prevalent data-exchange format
* XML (Extensible Markup Language)

Interestingly enough, the data exchange between the app and the RESTful service is programming language independent. The app maybe written in the C# programming language (or in Java, or Swift), and the RESTful service may be implemented in the Python programming language. The bridge (or interface) between the app and the service is the HTTP protocol and the format of the data.

Lab 12 will collect data in the JSON format.

For an OOP C# app the main challenges to interact with a RESTful service are:

1. Connect to the service and obtain the data
2. Convert JSON data into C# class objects

Fortunately there are C# built-in libraries and C# external, free public libraries (a.k.a., frameworks) which can be used by a C# app to interact with a RESTful service.

**Resources for Building the Lab 12 App**: In preparation for building the **Lab 12 app**, you will need to be acquainted with the following concepts:

1. [Introduction to JSON with C#](https://www.youtube.com/watch?v=lR-WVS__iug&nohtml5=False): This is a YouTube video which explains the JSON data format and importantly, how to add the Json.NET framework to a project
2. [What is NuGet and How to install its Packages into Visual Studio](http://davidsonsousa.net/en/post/what-is-nuget-and-how-to-install-its-packages-into-visual-studio-2010). (Although it refers to V.S. 2010) this link also explains how to add the Json.NET framework to a project
3. [Use C# to get JSON Data from the Web and Map it to .NET Class => Made Easy!](http://www.codeproject.com/Tips/397574/Use-Csharp-to-get-JSON-Data-from-the-Web-and-Map-i) This two-page document provides a **central code example we will use for building the Lab 12 app**. Your work consists of:
   1. Studying the code and understand what the code does
   2. Use the web to research any line of code in the document which you do not understand. (This is an example of the kind of research you will be doing in the “real-IT-world”, when you start working)
   3. The currency data comes from the [Open Exchange Rates (.org)](https://openexchangerates.org/) website. To get free access to the data you will need to create an account, by clicking the “sign up” button at the “.org” website, and click the “**Forever Free**” plan. (If the website asks for your e-mail, enter ***your CSCC e-mail address***). Upon successful signup, somewhere in a safe recoverable document save:
      1. Your login ID
      2. Your password, and
      3. Your **APP\_ID** (a long identifier) which you will need to retrieve currency data from the “.org” website.
      4. **IMPORTANT NOTE**: You will also need your **APP\_ID** when we actually implement the Lab 12 app.
   4. To demonstrate your understanding of the conversion of JSON data into objects of a C# class, ***use the code listed in resource 3*** to **write a console application** which simply displays the currency data obtained from the [Open Exchange Rates (.org)](https://openexchangerates.org/) website; that is, your console window should display values similar to the ones shown on page one, in **resource item 3**. **In your console application you will have to use**:
      1. Your Open Exchange Rates **APP\_ID** to retrieve the currency rates in JSON data format, and
      2. The Json.NET framework
      3. **IMPORTANT**: The **property identifiers** (and their order) defined in class **CurrencyRates** (in **resource item 3**) must match the **string-name identifiers** in the JSON payload retrieved from the “.org” website; **otherwise, the deserialization of the JSON payload will fail**. (For example, class **CurrencyRates** has the property **Rates**, which matches the string **“rates”** in the JSON data).

**You should complete the activities indicated on this paper one week before the topic *Consumption of Network Services* – see the table of activities at the end of the course syllabus.**