CIS 3325

Fall 2019

Assignment # 1

Due: Beginning of Class on Tuesday, September 17, 2019

50 points

**Assignment Requirements:**

### You will develop a single-form Windows application called CurrencyConverter. The app allows a user to enter an amount in US dollars and have that amount converted to selected foreign currency such as Japanese Yen (JPY), Chinese Yuan (CNY), Australian Dollar (AUD), Euro (EUR) and British Pound (GBP). Thus, this app is useful whenever shopping online at websites, which accept payment in multiple currencies.

### Reading Material: Murach (Ch. 12 - 362-379, 382-389; Ch. 4 – 100-124, 130-131; class notes and class demos)

**App Process Workflow:**

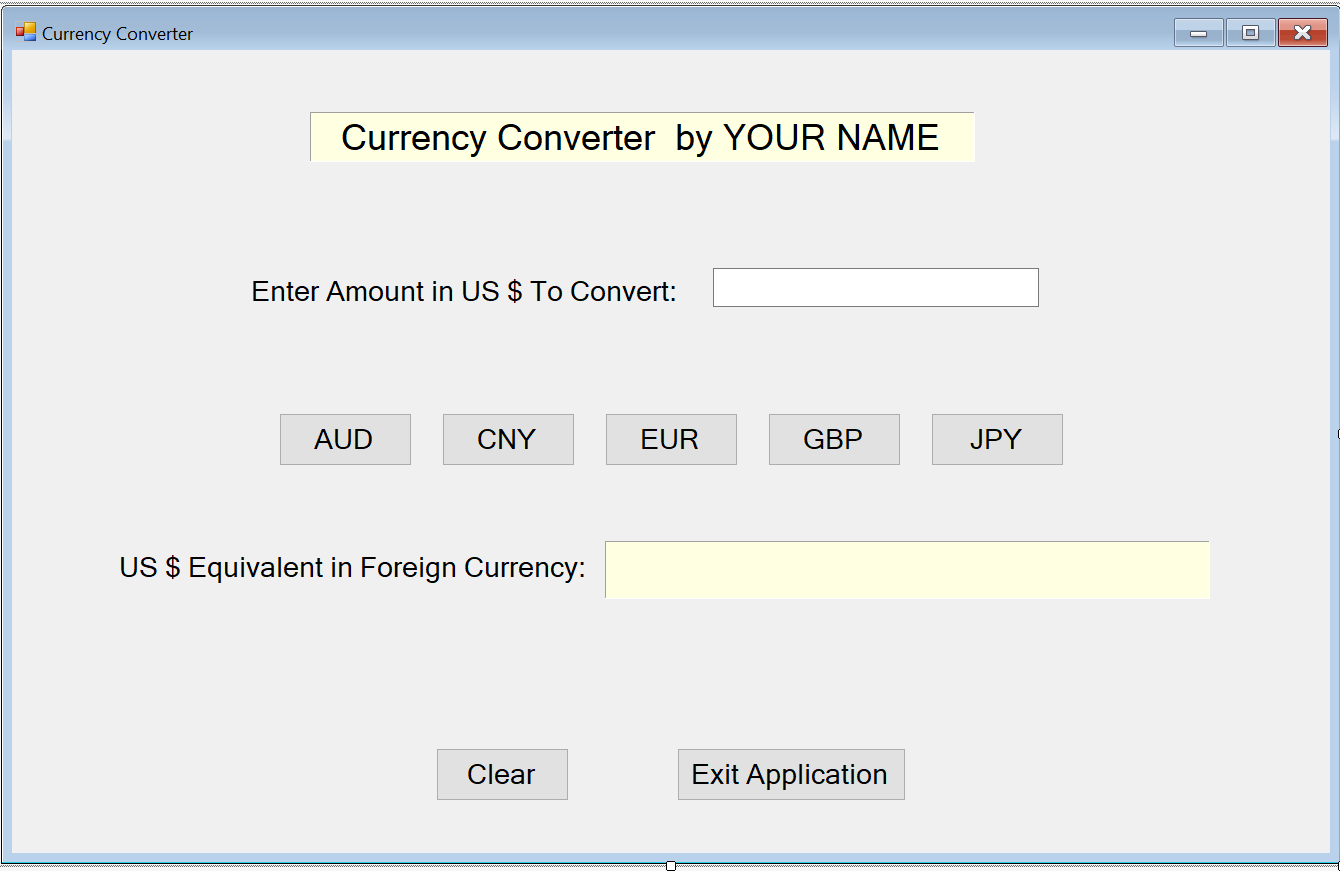
1. At start up, the app prompts the user to enter an amount in US $ to convert. The User enters a numeric value for the amount. This value must be of double data type.
2. User then selects one of five currencies to convert to by clicking on the appropriate command button. This invokes the calculate process to convert the entered $US amount into the selected foreign currency amount.
3. The converted amount is displayed in a label control as “US $XX is equal to (foreign Currency symbol) xxxx,” where XX is the $US value converted into xxxx in equivalent foreign currency amount. For example, if you are converting US $100 to British pounds, then the converted value will be displayed as

“US $100 is equal to GBP 40**”**

1. Clicking the “Clear” button clears the form of all values and places it in its startup position.
2. Clicking on the “Exit” button closes the form and then exits application.

**UI Layout Requirements:**

1. The UI will comprise of a single form. This UI is illustrated in Figure 1.
2. Ensure that your UI is well laid out and displayed in the center of the screen.
3. All naming conventions discussed in class must be adhered to and strictly followed.



**Processing Requirements:**

1. Your app **MUST use MVC design paradigm**. This is a requirement. A user-defined class named CurrencyConverterClass must act as the model and handle the currency conversion process.
2. Your app must define a CurrencyConverterClass that is designed to have the following structure. The declaration and implementation should be done in its own file named CurrencyConverterClass.cs in its own appropriately named class library project.
3. The CurrencyConverterClass specifications are provided in the class diagram presented below:

**CurrencyConverterClass**

Class



Fields



AUD : double



CNY : double



EUR : double



foreignCurrencyValue : double



GBP : double



JPY : double



usAmount : double



Properties



ForeignCurrencyValue { get; } : double



USAmount { set; } : double



Methods



CurrencyConversion(string currSymbol) : void



CurrencyConverter()

More details are provided below. Following conversion rates for US $1.00 were obtained from <https://www.exchange-rates.org> on September 5, 2019

|  |  |  |
| --- | --- | --- |
| Currency Selected | Currency Symbol | Exchange Rate for US $1.00 |
| Australian Dollar | AUD | 1.46740 |
| Chinese Yuan | CNY | 7.1490 |
| Euro | EUR | 0.90620 |
| Japanese Yen | JPY | 106.90979 |
| UK Pound | GBP | 0.811 |

|  |  |  |
| --- | --- | --- |
| CurrencyConverterClass | | |
|  | | |
| Instance Property Name | Property and Data Type | Initial Value |
| Private Fields/Properties: |  |  |
| usAmount | VAR, Double |  |
| foreignCurrencyValue | VAR, Double |  |
| Private Constants for storing exchange rates | | |
| AUD | Constant, Double | 1.46740 |
| CNY | Constant, Double | 7.1490 |
| EUR | Constant, Double | 0.90620 |
| JPY | Constant, Double | 106.90979 |
| GBP | Constant, Double | 0.811 |
| Public Process Methods: | | |
| CurrencyConversion (currSymbol: String) | | |
| CurrencyConverterClass() 🡨 default constructor | | |

1. Once the user taps on a command button, your app should invoke an action to initiate the conversion process for the selected foreign currency
2. The conversion process will:
   1. Instantiates a CurrencyConverterClass class;
   2. Access the public methods of the object to:
      1. Set the property for US dollar amount to be converted;
      2. Invoke the CurrencyConversion method, passing in the currency symbol; The CurrencyConversion method will evaluate the foreign currency selected, determine the exchange rate and compute the amount in foreign currency.
      3. Invoke the getter method for the foreignCurrencyValue property to retrieve the value in selected currency.
   3. Construct the display string as specified in step 4 of the App Process Workflow (HINT: use string concatenation as you learned in CIS 2324 and from Gaddis textbook).

**Submission Requirements:**

1. The assignment is to be completed individually by each student. This is a requirement. **Collaborative submissions (or work resulting from students working together) will be treated as academic dishonesty and handled accordingly – pay attention to this. I am serious about unsanctioned collaboration on projects and assignments.**
2. **The following submission requirements must be followed:**
3. The solution must be named: Assignment1-YourName. e.g. “Assignment1 - JohnDoe”.
4. The form project should be named: Assignment1UserInterface.
5. The Class Library project should be named: Assignment1ClassLibrary
6. Save the entire solution on a USB flash drive in a folder labeled “YourName” Replace “YourName” with your first and Last Name. e.g. “JohnDoe”.
7. This folder must be the only one on your USB drive and must be available at the root level of the drive. This folder should contain the all files created by Visual Studio for your assignment.
8. Submit the USB drive with your completed and **FULLY Working** app as well as the following printouts arranged per sequence below:
   1. A cover page with your name, project name, course number and section, and submission date.
   2. Printout (not a screenshot) of the form’s code file.
   3. Printout (not a screenshot) of CurrencyConverterClass.CS file
   4. Two Screenshots of your running app showing:
      * 1. user entering the amount to convert , and,
        2. the result of converting US $250 into Euros.
   5. **Staple** all printouts in the above order.
   6. Attach your USB drive to the printouts package with a paper clip. Hook the paper clip through the staple.

**Academic Dishonesty:**

1. **This is an individual assignment. Each individual is to do his/her own work.**
2. **Any collusion or sharing of work among individuals will be considered an academic dishonesty and will be handled in accordance with the Texas State’s Honor Code.**
3. **This instructor is serious about enforcing this policy to the fullest extent possible.**

Please plan ahead if you need to use the department provided computers in MCOY 336 to complete the assignments in this course. The lab hours vary and may not be available on the day you need to work on the assignments. It is highly recommended that you start work on the assignment the day it is made available on TRACS. It also helps if you plan out your app on paper first so that you don’t waste time once you get to the lab.