CIS 3325

Fall 2018

Assignment # 1

Due: Beginning of Class on Tuesday, September 18, 2018

50 points

**Assignment Requirements:**

### You will develop a simple windows application called CurrencyConverter. The app allows a user to enter the amount in US dollars and have that amount converted to selected foreign currency such as Japanese Yen, Chinese Yuan, Australian Dollar, Euro, and British Pound.

**App Process Workflow:**

1. At start up, the app prompts the user to enter an amount in US $ to convert and select one of five currencies to convert to.
2. User selects the currency type to convert to from a ListBox control.
3. User then clicks on the Convert button to convert the entered US dollar amount into the selected currency amount. The converted amount is displayed in a label control and should be properly formatted to show thousands comma separators and two decimal places.
4. Selecting a different currency to convert to from the Listbox and then clicking the Convert button should display the converted amount in the newly selected currency.
5. Clicking the “Clear” button clears the US amount textbox; deselects the selected value in the ListBox; clears the label showing converted value; and moves the cursor to the US amount textbox.
6. Clicking the “Exit” button, closes the form and exists application.

**UI Layout Requirements:**

1. The UI will comprise of the following UI elements and these elements will appear in the order described. The UI elements will be aligned and centered horizontally.
   1. Label displaying app name “Currency Converter by YourName”. Replace YourName with your first and last names.
   2. Label and a TextBox to prompt the user to enter the US $ amount to convert.
   3. Label to prompt user to select a currency type to convert to. This will be associated with a ListBox Control.
   4. A ListBox to display a list of currencies to select from. Only one currency may be selected at one time. Please refer to your CIS 2324 text to refresh your memory about how to work with this control.
   5. Label to display converted value. Associate another label with this label to inform users what will be displayed there.
   6. Three Buttons, one labeled “Convert”, second labeled “Clear”, and third one “Exit.”
   7. These labels, ListBox, Textbox and buttons should be properly aligned and spaced vertically for a visually appealing UI.
2. Tapping the “Clear” button resets the UI to its start up state – with no values and an unselected Listbox.
3. Ensure that your UI is well laid out. All controls, including the form, should be appropriately named and follow best practices of using lower camel case to name controls. The single form should be displayed in the center of the screen (see form’s startup position property).

**Processing Requirements:**

1. Your app MUST use MVC design paradigm. This is a requirement. A user-defined class named CurrencyConverterClass must handle the currency conversion process.
2. Please use the following conversion rates for US $1.00 obtained from <https://www.exchange-rates.org> on September 7, 2018.

|  |  |  |
| --- | --- | --- |
| Currency Selected | Currency Symbol | Exchange Rate for US $1.00 |
| Australian Dollar | AU$ | 1.40760 |
| Chinese Yuan | CNY | 6.8421 |
| Euro | € | 0.86483 |
| Japanese Yen | JPY | 111.08276 |
| UK Pound | British**£** | 0.77385 |

1. The Convert button should invoke an action methods that:
   1. Instantiates a CurrencyConverterClass;
   2. Access the public methods of the object to:
      1. Set the instance properties for US dollar amount to be converted and currency selected to convert to;
      2. Invoke the method ConvertCurrency, which would evaluate the selected currency to convert to determine the conversion rate to use and then compute the converted value by multiplying the US $ amount specified with the conversion rate and storing the result in the instance property of the CurrencyConverterClass.
      3. Invoke the getConvertedValue to retrieve and return the converted value in the selected currency to the form for display to the user.
   3. Construct the display string as specified in step 3 of the App Process Workflow (HINT: refer to CIS 2324 text to figure out how to format strings).
   4. Your app must declare a CurrencyConverterClass that is designed to have the following structure. The declaration and implementation should be done in its own C# file named CurrencyConverterClass.CS.

|  |  |  |
| --- | --- | --- |
| CurrencyConversionClass | | |
|  | | |
| Instance Property Name | Property and Data Type | Initial Value |
| Private Instance properties: |  |  |
| USAmount | VAR, Double | 0.0 |
| ConvertedValue | VAR, Double | 0.0 |
| currencyToConvertTo | VAR, String | “” |
| AUDollar | Constant, Double | 1.40760 |
| CNY | Constant, Double | 6.8421 |
| Euro | Constant, Double | 0.86483 |
| JPY | Constant, Double | 111.08276 |
| BritishPound | Constant, Double | 0.77385 |
| Public Instance Methods: | | |
| SetValues(US $ Amount: Double, SelectedCurrency: String) | | |
| ConvertCurrency() – no arguments used and no values returned | | |
| getConvertedValue() – This method returns the converted currency value as Double data type. | | |

**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. **Following submission requirements must be adhered to:**
3. The single-form windows project should be named – “YourName-Assignment1” e.g. “John Doe-Assignment1”
4. Save the windows project on a USB flash drive in a folder labeled “Your name” Replace “your name” with your first and Last Name. e.g. “John Doe”.
5. 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 artifacts created by VS 2017 for your assignment.
6. Submit the USB drive with your completed and running app as well as the following printouts arranged per sequence below:
   1. A cover page with your name, project name, and submission date.
   2. Printout of CurrencyConverterClass file. (Model)
   3. Printout of the form’s code behind file (Controller) file.
   4. Two Screenshots of your running app showing (View):
      * 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. **Please ask me how to do this in class** **before the due deadline**.

**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.**

Note:

You will have rely on many UI-related topics and basic programming structures you covered in CIS 2324. This will be a good refresher on the material you covered in CIS 2324.

Please plan ahead if you need to use the department provided windows PC 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.