Programming Language Learning Series
Mastery of Python Language
(Currency Converter)

In this project, you will design your own class that will interact with the Google Finance currency calculator API to perform conversions among various currencies.

Implement the following tasks:

1. Querying Calculator API: The library *urllib* is a way to create a connection to a web server and download data from the web server as if it were a file. The particular function in the library is *urlopen*, which we can use to create the connection. Like any file, you open it, read the contents, then close it. Here is an example:

```
import urllib
web_obj=
urllib.urlopen("https://www.google.com/finance/converter?a=100&from=USD&to=SEK
")
results_str = web_obj.read()
web_obj.close()
```

The result is a single string that is the full content of the web query you provided. It is best to use the *read()* function (which returns the entire contents as a single string) as you will find that iteration or other functions such as *readlines()* get somewhat confused on the end of line character. Make sure that the API is working as expected by end of this task.

2. Parsing: You need to parse the result string to find the information required. There are over 350 lines to the string and the one line you are interested in is about 8 lines from the bottom. For example, for the first sample above that converts 100 USD to SEK the line of interest is listed below and all you want is the converted value: 665.1200

```
<div id=currency_converter_result>100 USD =
                                    <span class=bld>665.1200 SEK</span>
```

Make sure you examine the invalid case (the last example above) to see what is returned. Make sure you parse the exact data you want by end of this task.

3. Class Design: Create a class called Currency. This class is to be instantiated with an amount and currency type (three character currency code), and is used to fetch information from the web. The class details:

a. __init__: The constructor takes the following arguments

a. An amount. Default: 0
b. A currency code. Default: USD

Programming Language Learning Series
Mastery of Python Language
(Currency Converter)

Note: If the currency code provided is invalid, set the currency code to an empty string and the amount to 0.

b. convert_to: method takes a single argument, a currency code, with no default. It returns a new Currency object with the new currency code and the converted amount. This method is where you will use *urllib* functions to get data from the web to convert this instance's currency to the currency specified in the parameter.

c. The following operations must be implemented:

a.__repr__: Return the amount and type as a string

b.__add__and __radd__: Return the type Currency and must be able to handle curr1+curr2, curr1+5, curr1+2.71

c.__sub__and __rsub__: Return the type Currency and must be able to handle curr1-curr2, curr1-5, curr1-2.71

d.__gt__: Return the type Currency and must be able to handle curr1>curr2

e.__str__ *method:* Return the amount and type as a string

4. Your class only needs to handle currency codes USD, EUR, SEK, CAD, CNY, and GBP

5. Unit Testing: Unit test each class you have written sothat you can be more confident on the project outcome.

6. Main Method: Write the code to integrate subtasks.

A sample test cases look like this:

```
curr=Currency(7.50,'USD')
print(curr)
curr2=Currency(2,'EUR')
print(curr2)
new_curr=curr2.convert_to('USD')
print(new_curr)
sum_curr=curr+curr2
print(sum_curr)
```

Programming Language Learning Series
Mastery of Python Language
(Currency Converter)

```
sum_curr2=curr+5.5

print(sum_curr2)
```