Lab 16 - Classes and operator overloading

Submission:

This lab has to be demoed on week 13, together with all other labs published between weeks 8 and 13. You will have to explain your code, so make sure it is well documented and you understand it.

Objective:

To practice, understand and design Python classes. To use operator overloading in a real problem.

Problem:

Extend the class Currency in the file `Lab16skeleton.py` in Week 11 folder. 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 are:

`__init__`: The constructor takes the following arguments (with defaults indicated):

a. An amount. Default: 1
b. A currency code. If the currency code provided is invalid, set the currency code to "" (an empty string) and the amount to 0. Default: EUR

`convert_to`: This 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 partially implemented. It uses the urllib.request library for opening URLs. You can see more about this library here https://docs.python.org/3/library/urllib.request.html

The `api.exchangeratesapi.io` website is used to perform conversions. The URL https://api.exchangeratesapi.io/latest?base=EUR&symbols=USD returns the a JSON string:

```
'{"rates":{"USD":1.1058},"base":"EUR","date":"2019-11-22"}'
```

While the URL https://api.exchangeratesapi.io/latest?base=EUR&symbols=GBP returns the JSON string :

```
'{"rates":{"GBP":0.8598},"base":"EUR","date":"2019-11-22"}'
```

Python offers a built-in JSON package which can be used to easily manipulate JSON data. You can see a nice introduction here https://www.w3schools.com/python/python_json.asp
For example, it is possible to access the fields of the JSON string above in the following way:

```
import json
exchange_info =
json.loads('{"rates":{"GBP":0.8598},"base":"EUR","date":"2019-11-22"}')

print(exchange_info["rates"]["GBP"]) # 0.8598
print(exchange_info["base"]) # EUR
print(exchange_info["date"]) # 2019-11-22
```

You should extend this method to extract the exchange rate from a JSON string of this type and return a new Currency object with the corrected converted amount.

The following operations must be implemented. For each method it should be able to handle operands of type Currency, float, or int (for example, your `__add__` method should be able to handle `curr1 + curr2`, or handle `curr1 + 5`, or handle `curr1 + 2.71`).

a. `__str__`: Return the amount and type as a string
b. `__add__` and `__radd__`
c. `__sub__` and `__rsub__`
d. `__gt__`
e. `__repr__`

Include a main program that tests all your methods.