09 Practical

Basics

1. Write a loop that will add up all the even numbers from 1 to 100 inclusive. Your result should be 2550.
2. Write a function that will take two strings as input, and return the longest string, if they are the same length it will return the first one.

File Input and Output

3. Write code that will write the following to File1.csv.

```
Student, Mark
Foo, 10
Bar, 50
Alan, 20
Tim, 30
```

4. Write code that will calculate the average of the marks column, the minimum mark and the maximum mark and add it to a new file along with the rest of the content.
I suggest you do this in steps, figure out the average then the rest.

Input (File1.csv)	Desired Output (FIle2.csv)

`Student, Mark`
`Foo, 10`
`Bar, 50`
`Alan, 20`
`Tim, 30`
	`Student, Mark`
	`Foo, 10`
	`Bar, 50`
	`Alan, 20`
	`Tim, 30`
	`Avg, 27.5`
	`Max, 50`
	`Min, 10`

Classes

5. Create a new class called **`SimpleShopping`**, the class will need a name and will also have a dictionary as a property. As below:

```
class SimpleShopping:
   def __init__(self, name):
       self.name = name
       self.items = dict()
```

6. Add two new methods to the shopping class “add” and “remove”, add will add a new item to the shopping list, while remove will remove it. Usage example:

```
shop = SimpleShopping("New")  # shop.items: {}
shop.add("Apple")        # shop.items: {'Apple': 1}
shop.add("Banana")       # shop.items: {'Apple': 1, 'Banana': 1}
shop.add("Apple")        # shop.items: {'Apple': 1, 'Banana': 1}
shop.remove("Apple")     # shop.items: {'Banana': 1}
```

7. Currently our shopping list assumes we only ever have or need one of everything. Create a subclass of our **`SimpleShopping`** class called **`SmartShopping`** that will keep track of quantities properly. Usage example:

```
class SmartShopping(SimpleShopping):
   # Your code here

shop = SmartShopping("New")   # shop.items: {}
shop.add("Apple")        # shop.items: {'Apple': 1}
shop.add("Banana")       # shop.items: {'Apple': 1, 'Banana': 1}
shop.add("Apple")        # shop.items: {'Apple': 2, 'Banana': 1}
shop.remove("Apple")     # shop.items: {'Apple': 1, 'Banana': 1}
shop.remove("Apple")     # shop.items: {'Banana': 1}
```

Testing

8. Write appropriate doc tests for the function you wrote in Q2.

Reference

DocTests

Recall that a doctest looks like this:

```
import doctest

def add(x, y):
   """
   Add two numbers
   >>> add(5, 2)
   7
   >>> add(2, 2)
   4
   >>> add(9, 12)
   21
   """
   return x + y

doctest.testmod(verbose=True) # Run the tests
```

Unit Tests

Recall that a unit test is written like so:

```
import unittest

class TestAdd(unittest.TestCase):
   def test_add(self):
       self.assertEqual(add(3, 4), 7)
       self.assertTrue(add(3,4) == add(6,1))

unittest.main()  # Run the tests
```

File IO

Normal Reader

```python
with open("list.txt", "r") as shop:
    read = csv.reader(shop)
    for row in read:
        print(row)
```

Normal Writer

```python
with open("list.txt", "w") as shop:
    c = csv.writer(shop)
    c.writerow(['Item', 'Quantity'])
    c.writerow(['Apple', '2'])
    c.writerow(['Orange', '5'])
```

Dictionary Reader

```python
with open("list.txt", "r") as shop:
    read = csv.DictReader(shop)
    for row in read:
        print(row)
```

Dict Writer

```python
with open("list.txt", "w") as shop:
    c = csv.DictWriter(shop, fieldnames=['Item', 'Quantity'])
    c.writerow({'Item': 'Apple', 'Quantity': '2'})
```