Exercise 1.6

Objectives:

- Defining modules
- Using the import statement

Files Created: None

Note: For this exercise involving modules, it is critically important to make sure you are running Python in a proper environment. You may need to check the value of `sys.path` if you can't get import statements to work. Ask for assistance if everything seems broken.

Before starting this exercise, first restart your Python interpreter session. If using IDLE, click on the shell window and look for a menu option "Shell > Restart Shell". You should get a message like this:

```
>>> ##################== RESTART ##################==
>>>
```

If you are using Unix, simply exit Python and restart the interpreter.

(a) Using the import statement

In previous exercises, you wrote two programs `pcost.py` and `stock.py`. Use the `import` statement to load these programs and use their functionality:

```
>>> import pcost
44671.15
>>> pcost.portfolio_cost('Data/portfolio2.dat')
19908.75
>>> from stock import Stock
>>> s = Stock('GOOG', 100, 490.10)
>>> s.name
'GOOG'
>>> s.cost()
49010.0
>>>
```

If you can't get the above statements to work, you might have placed your programs in a funny directory. Make sure you are running Python in the same directory as your files or that the directory is included on `sys.path`.

(b) Main Module

In your `pcost.py` program, the last statement called a function and printed out the result. Modify the program so that this step only occurs if the program is run as the main program. Now, try running the program two ways:

First, run the program as main:

```
bash % python3 pcost.py
44671.25
bash %
```

Next, run the program as a library import. You should not see any output.

```
>>> import pcost
>>>
```

[Solution | Index | Exercise 1.5 | Exercise 2.1]

>>> Advanced Python Mastery

... A course by dabeaz

... Copyright 2007-2023