Programming Fundamentals Using Python

2018

Problem Set 14

Most recent updated: July 18, 2018

Objectives

1. Decorator

Note: Solve the programming problems listed using your favorite text editor. Make sure you save your programs in files with suitably chosen names, **and try as much as possible to write your code with good style (see the style guide for python code)**. In each problem find out a way to test the correctness of your program. After writing each program, test it, debug it if the program is incorrect, correct it, and repeat this process until you have a fully working program. Show your working program to one of the cohort instructors.

Problems: Cohort sessions

1. *Higher order function* Write a function called `add_powers(n)` that returns a function that takes in a list. The function `add_powers(n)` sums all the elements to the power of n. For example,

```
>>> series1 = [0, 1, 2, 3, 4, 5]
>>> series2 = [2, 4, 8, 16, 32]
>>> power_reducer = add_powers(2)
>>> power_reducer(series1)
55
>>> power_reducer(series2)
1364
>>> power_reducer = add_powers(3)
>>> power_reducer(series1)
225
>>> power_reducer(series2)
37448
```

2. *Decorator* Write a decorator function called `logit(f)` which takes any function and modifies it behaviour in such a way that it prints out the function address and lists all its arguments.

3. *Decorator property* Write a decorator function which does what `@property` do to a method. Recall that you can create a property by calling the following function:

   ```
   property_name = property(f_getter, f_setter)
   ```

   Name your decorator `myproperty` and test it on your `Fraction` class.

End of Problem Set 14.