

Homework 7 [Extra Credit] – Functions and Modules

Due on Wednesday, December 07, 2016 11:59 AM (Noon)

Instructions: Write Python scripts (.py files) to solve the following problems. Output screenshots should also be submitted. Use variables where necessary and give meaningful names to variables. All script files should have a comment block at the top, and also provide comments alongside your code. Upload a .pdf/.docx file containing your solutions along with your .py files.

Problem 1. Implement your own 'mymath' module.

[60 points]

In this exercise, you are going to create and use your own math module 'mymath', which supports six functions – add, subtract, multiply, divide, isPrime and factorial.

Your `mymath.py` file should have the definitions of the following six functions:

- (i) `add()` –
Formal parameters: num1, num2
Return value: Sum of num1 and num2
- (ii) `subtract()` –
Formal parameters: num1, num2
Return value: Difference of num1 and num2
- (iii) `multiply()` –
Formal parameters: num1, num2
Return value: Product of num1 and num2
- (iv) `divide()` –
Formal parameters: num1, num2
Return value: Division of num1 and num2
- (v) `isPrime()` –
Formal parameter: num
Return value (Boolean): `True` if num is prime, `False` otherwise
[Your code should evaluate correctly for all non-negative integers]
- (vi) `factorial()` –
Formal parameter: num
Return value: Factorial value of num
[Your code should evaluate correctly for all non-negative integers]

Now that you have implemented your module, it is time to test if it works correctly. Create a new python script `testmymath.py`. In this script do the following:

- (i) Import the module you just created.
- (ii) Input two numbers from the user and save them in variables x and y.
- (iii) Call `add()` with x and y as arguments (actual parameters) and save the returned value in variable resAdd. Print the value of resAdd.
- (iv) Call `subtract()` with x and y as arguments and save the returned value in variable resSub. Print the value of resSub.

- (v) Call `multiply()` with `x` and `y` as arguments and save the returned value in variable `resMul`. Print the value of `resMul`.
- (vi) Call `divide()` with `x` and `y` as arguments and save the returned value in variable `resDiv`. Print the value of `resDiv`.
- (vii) Call `isPrime()` with `x` as argument and save the returned value in variable `resPri`. Print the value of `resPri`.
- (viii) Call `factorial()` with `y` as argument and save the returned value in variable `resFac`. Print the value of `resFac`.

Note that you need to use `module_name.function_name` format to call your functions. You should run the test script multiple times with different values to ensure that the functions are working properly.

Sample output of testmymath.py:

```
Enter 1st number: 11
Enter 2nd number: 6
```

```
Sum = 17
Difference = 5
Product = 66
Division = 1.8333
```

```
11 is Prime
6 Factorial = 720
```

Files to be submitted:

1. Module: `mymath.py`
2. Test script: `testmymath.py`
3. Output screenshots

Problem 2. Write a python script `fibonacci.py` to compute the n^{th} number in the Fibonacci sequence (1, 1, 2, 3, 5, 8, 13, 21, ...), where n is any positive integer. You have to do the following in your script: **[40 points]**

- (i) Define a function called `fibonacci()` that has one formal parameter n , and returns the n^{th} number in the Fibonacci sequence.
- (ii) Write code that ask the user to enter a number and save it in variable `num`.
- (iii) Call `fibonacci()` with `num` as argument and save the result in variable `f`. Print the value of `f`.

Sample output I of fibonacci.py:

Enter a number: 5

Fibonacci = 5

Sample output II of fibonacci.py:

Enter a number: 2

Fibonacci = 1

Files to be submitted:

1. Script: `fibonacci.py`
2. Output screenshots