

hw6 Problem2&3&4

November 20, 2018

1 Assignment 7 Part 1

1.0.1 MACS 30000, Dr. Evans

1.0.2 Zeyu Xu

Due Monday, Nov. 26 at 11:30 AM

1.0.3 1. Unit Testing in Python

Problem 1

Here is my code to test the function, the followings are respectively `smallest_factor` function (saved as `Problem1_a.py`), test cases (saved as `test_Problem1_a.py`) and check code.

```
In [1]: def smallest_factor(n):  
        """Return the smallest prime factor of the positive integer n."""  
        if n == 1: return 1  
        for i in range(2, int(n**.5)):  
            if n % i == 0: return i  
        return n
```

```
In [2]: import Problem1_a  
        def test_case():  
            assert Problem1_a.smallest_factor(9) == 3, "failed on 9"
```

```
In [3]: ! py.test
```

```
===== test session starts =====
```

```
platform win32 -- Python 3.6.4, pytest-4.0.0, py-1.7.0, pluggy-0.8.0
```

```
rootdir: C:\Users\ccb\Desktop\Perspective\persp-analysis_A18\Assignments\A7\Problem 1, inifile
```

```
plugins: cov-2.6.0
```

```
collected 3 items
```

```
test_Problem1_a.py F [ 33%]
```

```
test_Problem1_b.py . [ 66%]
```

```
test_Problem1_c.py . [100%]
```

```
===== FAILURES =====
```

```

----- test_case -----

def test_case():
>     assert Problem1_a.smallest_factor(9) == 3, "failed on 9"
E     AssertionError: failed on 9
E     assert 9 == 3
E     + where 9 = <function smallest_factor at 0x000001D091E15C80>(9)
E     + where <function smallest_factor at 0x000001D091E15C80> = Problem1_a.smallest_factor

test_Problem1_a.py:3: AssertionError
===== 1 failed, 2 passed in 0.10 seconds =====

```

Here is my correction

```

In [4]: def smallest_factor(n):
        """Return the smallest prime factor of the positive integer n."""
        if n == 1: return 1
        for i in range(2, int(n**.5)+1):
            if n % i == 0: return i
        return n

```

Problem 2

Here is my code to check my coverage on smallest_factor function

```

In [5]: ! py.test --cov

```

```

===== test session starts =====
platform win32 -- Python 3.6.4, pytest-4.0.0, py-1.7.0, pluggy-0.8.0
rootdir: C:\Users\ccb\Desktop\Perspective\persp-analysis_A18\Assignments\A7\Problem 1, inifile
plugins: cov-2.6.0
collected 3 items

test_Problem1_a.py F                                [ 33%]
test_Problem1_b.py .                                [ 66%]
test_Problem1_c.py .                                [100%]

```

```

===== FAILURES =====
----- test_case -----

def test_case():
>     assert Problem1_a.smallest_factor(9) == 3, "failed on 9"
E     AssertionError: failed on 9
E     assert 9 == 3
E     + where 9 = <function smallest_factor at 0x000001EE0AB727B8>(9)
E     + where <function smallest_factor at 0x000001EE0AB727B8> = Problem1_a.smallest_factor

```

```
test_Problem1_a.py:3: AssertionError
```

```
----- coverage: platform win32, python 3.6.4-final-0 -----
```

| Name | Stmts | Miss | Cover |
|--------------------|-------|------|-------|
| Problem1_a.py | 5 | 0 | 100% |
| Problem1_b.py | 10 | 0 | 100% |
| Problem1_c.py | 14 | 0 | 100% |
| test_Problem1_a.py | 3 | 0 | 100% |
| test_Problem1_b.py | 7 | 0 | 100% |
| test_Problem1_c.py | 9 | 0 | 100% |
| TOTAL | 48 | 0 | 100% |

```
===== 1 failed, 2 passed in 0.09 seconds =====
```

Here is my code to test month_length function, the followings are respectively month_length function (saved as Problem1_b.py), test cases (saved as test_Problem1_b.py) and check code.

```
In [6]: def month_length(month, leap_year=False):
        """Return the number of days in the given month."""
        if month in {"September", "April", "June", "November"}:
            return 30
        elif month in {"January", "March", "May", "July", "August", "October", "December"}:
            return 31
        if month == "February":
            if not leap_year:
                return 28
            else:
                return 29
        else:
            return None
```

```
In [7]: import Problem1_b
        def test_cases():
            assert Problem1_b.month_length("September") == 30, "False Function"
            assert Problem1_b.month_length("March") == 31, "False Function"
            assert Problem1_b.month_length("February") == 28, "False Function"
            assert Problem1_b.month_length("February", leap_year = True) == 29, "False Function"
            assert Problem1_b.month_length(3) == None, "False Function"
```

```
In [8]: ! py.test --cov
```

```
===== test session starts =====
```

```
platform win32 -- Python 3.6.4, pytest-4.0.0, py-1.7.0, pluggy-0.8.0
```

```
rootdir: C:\Users\ccb\Desktop\Perspective\persp-analysis_A18\Assignments\A7\Problem 1, inifile
```

```
plugins: cov-2.6.0
```

collected 3 items

```
test_Problem1_a.py F [ 33%]
test_Problem1_b.py . [ 66%]
test_Problem1_c.py . [100%]
```

```
===== FAILURES =====
----- test_case -----
```

```
def test_case():
>     assert Problem1_a.smallest_factor(9) == 3, "failed on 9"
E     AssertionError: failed on 9
E     assert 9 == 3
E     + where 9 = <function smallest_factor at 0x000001FF5CBA07B8>(9)
E     + where <function smallest_factor at 0x000001FF5CBA07B8> = Problem1_a.smallest_factor
```

test_Problem1_a.py:3: AssertionError

----- coverage: platform win32, python 3.6.4-final-0 -----

| Name | Stmts | Miss | Cover |
|--------------------|-------|------|-------|
| Problem1_a.py | 5 | 0 | 100% |
| Problem1_b.py | 10 | 0 | 100% |
| Problem1_c.py | 14 | 0 | 100% |
| test_Problem1_a.py | 3 | 0 | 100% |
| test_Problem1_b.py | 7 | 0 | 100% |
| test_Problem1_c.py | 9 | 0 | 100% |
| TOTAL | 48 | 0 | 100% |

===== 1 failed, 2 passed in 0.09 seconds =====

Problem 3

Here is my code to test operator function and check coverage, the followings are respectively operator function (saved as Problem1_c.py), test cases (saved as test_Problem1_c.py) and check code.

```
In [9]: def operate(a, b, oper):
        """Apply an arithmetic operation to a and b."""
        if type(oper) is not str:
            raise TypeError("oper must be a string")
        elif oper == '+':
            return a + b
        elif oper == '-':
            return a - b
        elif oper == '*':
```

```

        return a * b
    elif oper == '/':
        if b == 0:
            raise ZeroDivisionError("division by zero is undefined")
        return a / b
    raise ValueError("oper must be one of '+', '/', '-', or '*'")

```

```

In [10]: import Problem1_c, pytest
def test_cases():
    assert Problem1_c.operate(1,2,"+") == 3, "False Function"
    assert Problem1_c.operate(1,2,"-") == -1, "False Function"
    assert Problem1_c.operate(1,2,"*") == 2, "False Function"
    assert Problem1_c.operate(1,2,"/") == 0.5, "False Function"
    pytest.raises(ZeroDivisionError, Problem1_c.operate, a=1, b=0, oper='/')
    pytest.raises(TypeError, Problem1_c.operate, a=1, b=1, oper=1)
    pytest.raises(ValueError, Problem1_c.operate, a=1, b=1, oper="1")

```

```

In [11]: ! py.test --cov

```

```

===== test session starts =====

```

```

platform win32 -- Python 3.6.4, pytest-4.0.0, py-1.7.0, pluggy-0.8.0

```

```

rootdir: C:\Users\ccb\Desktop\Perspective\persp-analysis_A18\Assignments\A7\Problem 1, inifile

```

```

plugins: cov-2.6.0

```

```

collected 3 items

```

```

test_Problem1_a.py F [ 33%]
test_Problem1_b.py . [ 66%]
test_Problem1_c.py . [100%]

```

```

===== FAILURES =====

```

```

----- test_case -----

```

```

def test_case():
>     assert Problem1_a.smallest_factor(9) == 3, "failed on 9"
E     AssertionError: failed on 9
E     assert 9 == 3
E     + where 9 = <function smallest_factor at 0x000001CDD91827B8>(9)
E     + where <function smallest_factor at 0x000001CDD91827B8> = Problem1_a.smallest_factor

```

```

test_Problem1_a.py:3: AssertionError

```

```

----- coverage: platform win32, python 3.6.4-final-0 -----

```

| Name | Stmts | Miss | Cover |
|--------------------|-------|------|-------|
| Problem1_a.py | 5 | 0 | 100% |
| Problem1_b.py | 10 | 0 | 100% |
| Problem1_c.py | 14 | 0 | 100% |
| test_Problem1_a.py | 3 | 0 | 100% |

| | | | |
|--------------------|----|---|------|
| test_Problem1_b.py | 7 | 0 | 100% |
| test_Problem1_c.py | 9 | 0 | 100% |
| ----- | | | |
| TOTAL | 48 | 0 | 100% |

===== 1 failed, 2 passed in 0.09 seconds =====