

Unit Testing 1 - Exercise 1 Solution

1.

```
def add_two_numbers(a, b):
    if (not isinstance(a, int) and \
        not isinstance(a, float)) or \
        (not isinstance(b, int) and \
         not isinstance(b, float)):
        return "Invalid input"
    return a + b

# scenario 1 - adds two whole numbers
def test_adds_two_whole_numbers():
    expected = 2
    actual = add_two_numbers(1, 1)
    assert expected == actual

test_adds_two_whole_numbers()

# scenario 2 - adds a positive whole number to a negative whole number
def test_adds_pos_neg_numbers():
    expected = 0
    actual = add_two_numbers(-100, 100)
    assert expected == actual

test_adds_pos_neg_numbers()

# scenario 3 - adds two floating point numbers
def test_adds_two_fp_numbers():
    expected = 0.9
    actual = add_two_numbers(0.5, 0.4)
    assert expected == actual

test_adds_two_fp_numbers()

# scenario 4 - adds a string to a whole number
def test_adds_string_to_number():
    expected = "Invalid Input"
    actual = add_two_numbers("test", 1)
    assert expected == actual

test_adds_string_to_number()

# scenario 5 - adds two strings
def test_adds_two_strings():
```

```
def test_adds_two_strings():
    expected = "Invalid Input"
    actual = add_two_numbers("test", "case")
    assert expected == actual

test_adds_two_strings()
```