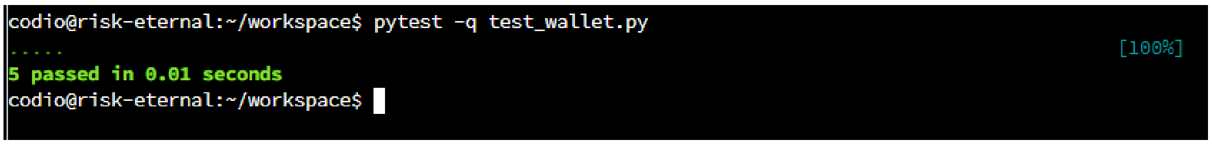
UNIT 6 CODIO ACTIVITY

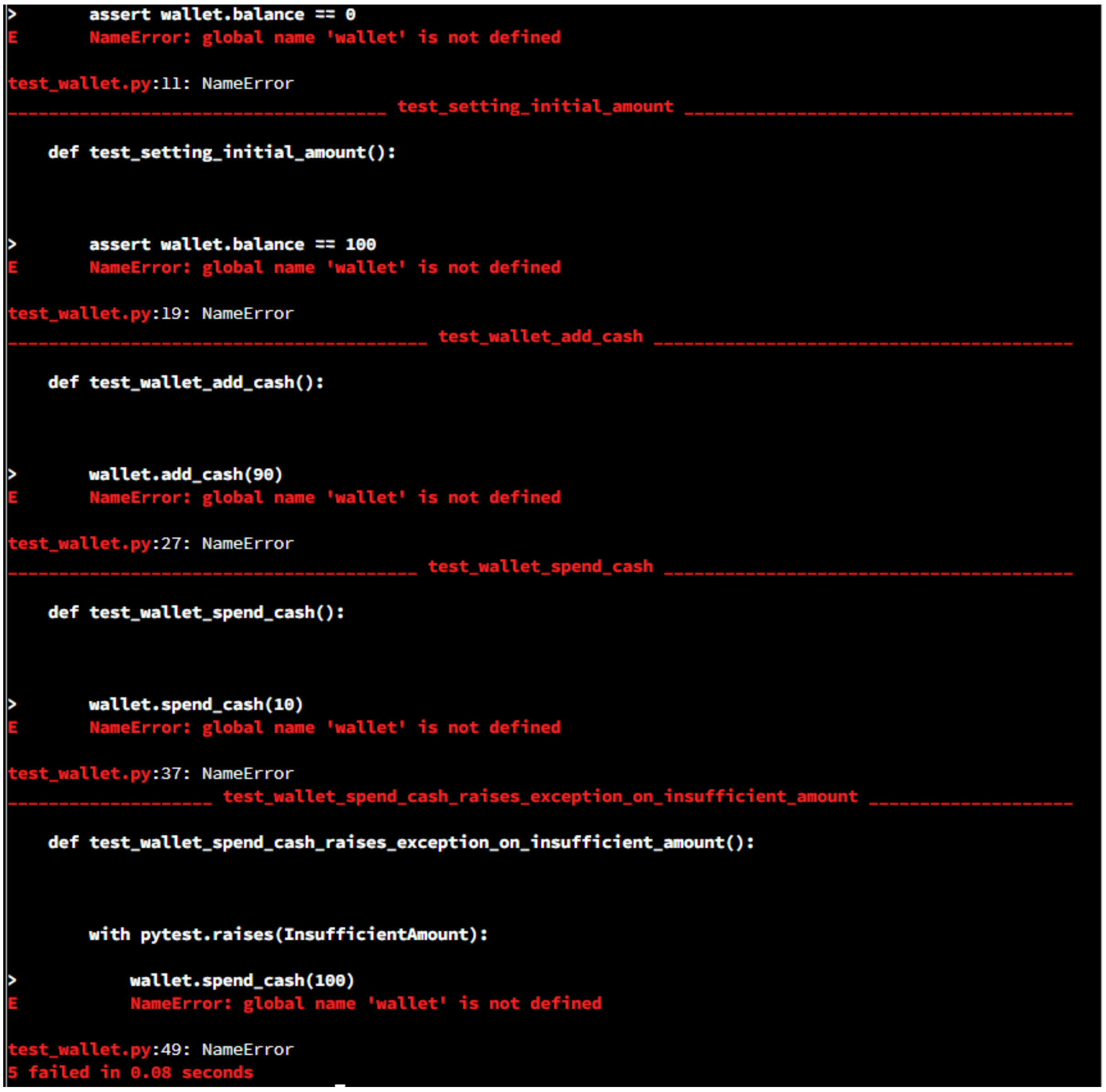
PYTEST



```
codio@risk-eternal:~/workspace$ pytest -q test_wallet.py
.....                                                                    [100%]
5 passed in 0.01 seconds
codio@risk-eternal:~/workspace$ 
```

Fig 1: Screenshot showing successful pytest run on test_wallet.py

After amending the codes



```
>       assert wallet.balance == 0
E       NameError: global name 'wallet' is not defined

test_wallet.py:11: NameError
____________________________________ test_setting_initial_amount ____________________________________

    def test_setting_initial_amount():


>       assert wallet.balance == 100
E       NameError: global name 'wallet' is not defined

test_wallet.py:19: NameError
________________________________________ test_wallet_add_cash ________________________________________

    def test_wallet_add_cash():


>       wallet.add_cash(90)
E       NameError: global name 'wallet' is not defined

test_wallet.py:27: NameError
_______________________________________ test_wallet_spend_cash _______________________________________

    def test_wallet_spend_cash():


>       wallet.spend_cash(10)
E       NameError: global name 'wallet' is not defined

test_wallet.py:37: NameError
__________________ test_wallet_spend_cash_raises_exception_on_insufficient_amount ___________________

    def test_wallet_spend_cash_raises_exception_on_insufficient_amount():


        with pytest.raises(InsufficientAmount):

>           wallet.spend_cash(100)
E           NameError: global name 'wallet' is not defined

test_wallet.py:49: NameError
5 failed in 0.08 seconds
```

Fig 2: Failed pytest run due to NameError

I caused NameErrors in the test_wallet.py file by removing the test keyword in the function names. Ideally test functions should be named as shown below (starting with "test")

```python
def test_wallet_add_cash():

    wallet = Wallet(10)

    wallet.add_cash(90)

    assert wallet.balance == 100


def test_wallet_spend_cash():

    wallet = Wallet(20)

    wallet.spend_cash(10)

    assert wallet.balance == 10
```

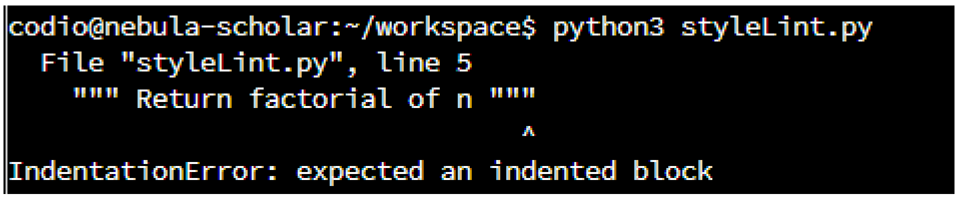
Unit 6 - Testing with Python

Exploring Linters to Support Testing in Python

Question 1

Run styleLint.py.

What happens when the code is run? Indentation error, the code was not properly indented



```
codio@nebula-scholar:~/workspace$ python3 styleLint.py
  File "styleLint.py", line 5
    """ Return factorial of n """
                                ^
IndentationError: expected an indented block
```

Can you modify this code for a more favourable outcome? Yes

```
# CODE SOURCE: SOFTWARE ARCHITECTURE WITH PYTHON

def factorial(n):
""" Return factorial of n """
if n == 0:
return 1
else:
return n*factorial(n-1)
```

Modified version

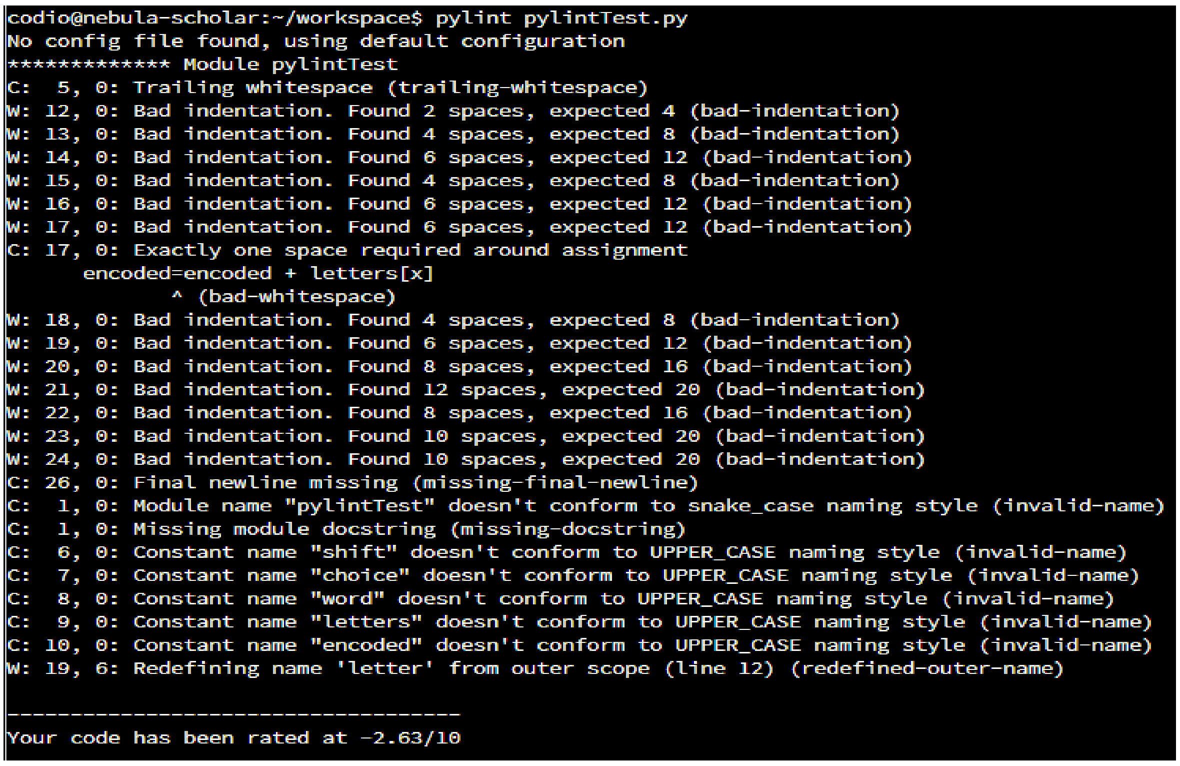
```
# CODE SOURCE: SOFTWARE ARCHITECTURE WITH PYTHON

def factorial(n):
  """ Return factorial of n """
  if n == 0:
    return 1
  else:
    return n*factorial(n-1)
```

What amendments have you made to the code? **I indented the code in line with python style**

Question 2:Run pylint on pylintTest.py

```
# SOURCE OF CODE: https://docs.pylint.org/en/1.6.0/tutorial.html

import string

shift = 3
choice = raw_input("would you like to encode or decode?")
word = (raw_input("Please enter text"))
letters = string.ascii_letters + string.punctuation + string.digits
encoded = ''
if choice == "encode":
  for letter in word:
    if letter == ' ':
      encoded = encoded + ' '
    else:
      x = letters.index(letter) + shift
      encoded=encoded + letters[x]
    if choice == "decode":
      for letter in word:
        if letter == ' ':
            encoded = encoded + ' '
        else:
          x = letters.index(letter) - shift
          encoded = encoded + letters[x]

print encoded
```

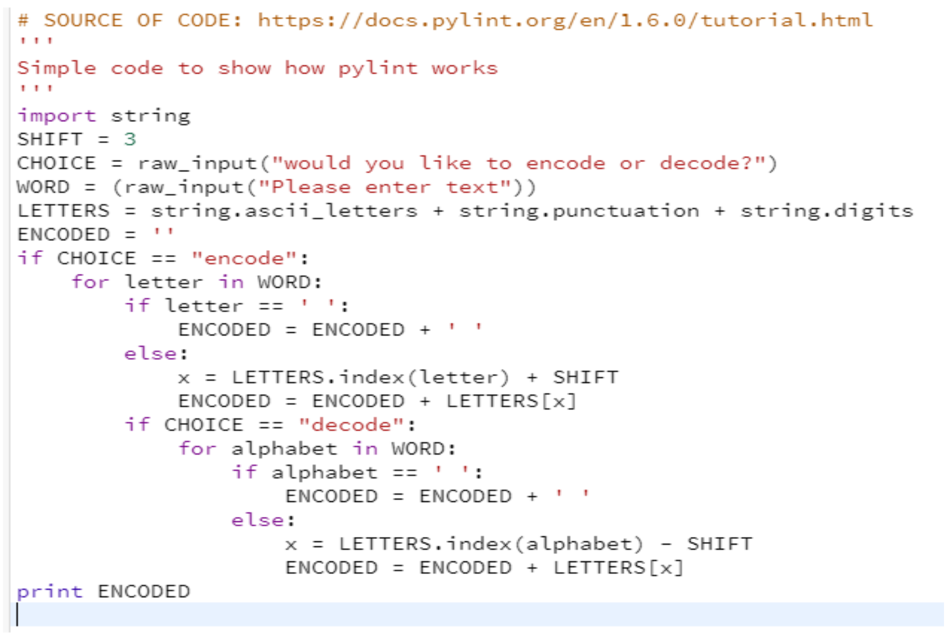
- **Review each of the code errors returned.**



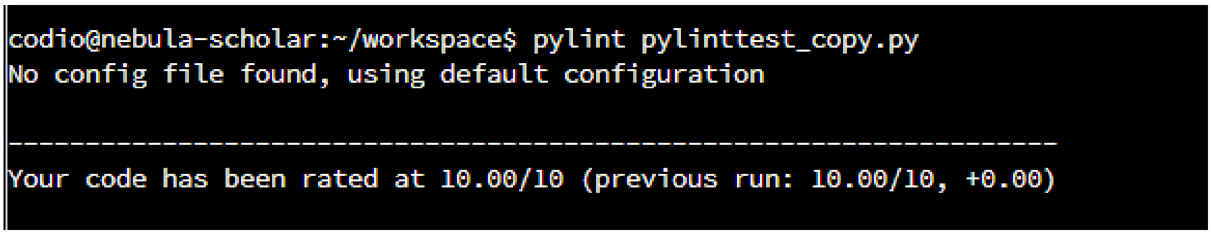
```
codio@nebula-scholar:~/workspace$ pylint pylintTest.py
No config file found, using default configuration
************* Module pylintTest
C:  5, 0: Trailing whitespace (trailing-whitespace)
W: 12, 0: Bad indentation. Found 2 spaces, expected 4 (bad-indentation)
W: 13, 0: Bad indentation. Found 4 spaces, expected 8 (bad-indentation)
W: 14, 0: Bad indentation. Found 6 spaces, expected 12 (bad-indentation)
W: 15, 0: Bad indentation. Found 4 spaces, expected 8 (bad-indentation)
W: 16, 0: Bad indentation. Found 6 spaces, expected 12 (bad-indentation)
W: 17, 0: Bad indentation. Found 6 spaces, expected 12 (bad-indentation)
C: 17, 0: Exactly one space required around assignment
      encoded=encoded + letters[x]
             ^ (bad-whitespace)
W: 18, 0: Bad indentation. Found 4 spaces, expected 8 (bad-indentation)
W: 19, 0: Bad indentation. Found 6 spaces, expected 12 (bad-indentation)
W: 20, 0: Bad indentation. Found 8 spaces, expected 16 (bad-indentation)
W: 21, 0: Bad indentation. Found 12 spaces, expected 20 (bad-indentation)
W: 22, 0: Bad indentation. Found 8 spaces, expected 16 (bad-indentation)
W: 23, 0: Bad indentation. Found 10 spaces, expected 20 (bad-indentation)
W: 24, 0: Bad indentation. Found 10 spaces, expected 20 (bad-indentation)
C: 26, 0: Final newline missing (missing-final-newline)
C:  1, 0: Module name "pylintTest" doesn't conform to snake_case naming style (invalid-name)
C:  1, 0: Missing module docstring (missing-docstring)
C:  6, 0: Constant name "shift" doesn't conform to UPPER_CASE naming style (invalid-name)
C:  7, 0: Constant name "choice" doesn't conform to UPPER_CASE naming style (invalid-name)
C:  8, 0: Constant name "word" doesn't conform to UPPER_CASE naming style (invalid-name)
C:  9, 0: Constant name "letters" doesn't conform to UPPER_CASE naming style (invalid-name)
C: 10, 0: Constant name "encoded" doesn't conform to UPPER_CASE naming style (invalid-name)
W: 19, 6: Redefining name 'letter' from outer scope (line 12) (redefined-outer-name)

------------------------------------
Your code has been rated at -2.63/10
```

- **Can you correct each of the errors identified by pylint? Before correcting the code errors, save the pylintTest.py file with a new name (it will be needed again in the next question).**

Corrected Version



```
# SOURCE OF CODE: https://docs.pylint.org/en/1.6.0/tutorial.html
'''
Simple code to show how pylint works
'''
import string
SHIFT = 3
CHOICE = raw_input("would you like to encode or decode?")
WORD = (raw_input("Please enter text"))
LETTERS = string.ascii_letters + string.punctuation + string.digits
ENCODED = ''
if CHOICE == "encode":
    for letter in WORD:
        if letter == ' ':
            ENCODED = ENCODED + ' '
        else:
            x = LETTERS.index(letter) + SHIFT
            ENCODED = ENCODED + LETTERS[x]
        if CHOICE == "decode":
            for alphabet in WORD:
                if alphabet == ' ':
                    ENCODED = ENCODED + ' '
                else:
                    x = LETTERS.index(alphabet) - SHIFT
                    ENCODED = ENCODED + LETTERS[x]
print ENCODED
```



```
codio@nebula-scholar:~/workspace$ pylint pylinttest_copy.py
No config file found, using default configuration

--------------------------------------------------------------------
Your code has been rated at 10.00/10 (previous run: 10.00/10, +0.00)
```

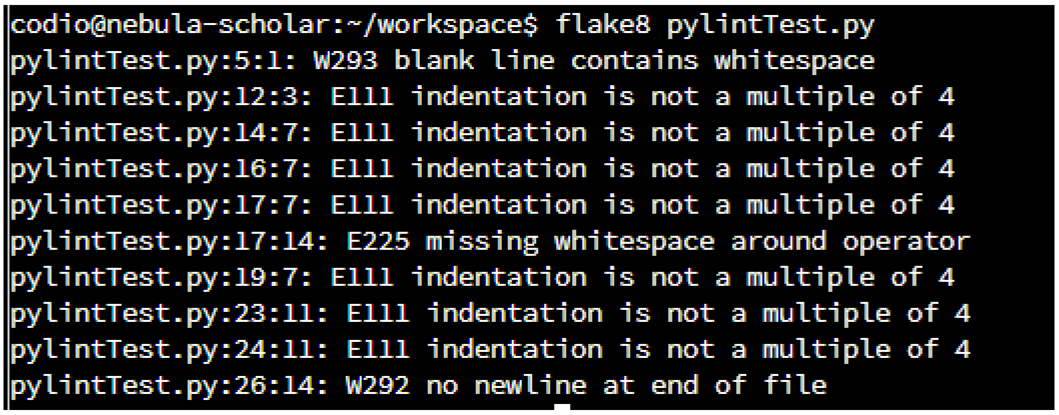
Question 3

Ensure flake8 is in your virtual box -

pip install flake8

Run flake8 on pylintTest.py

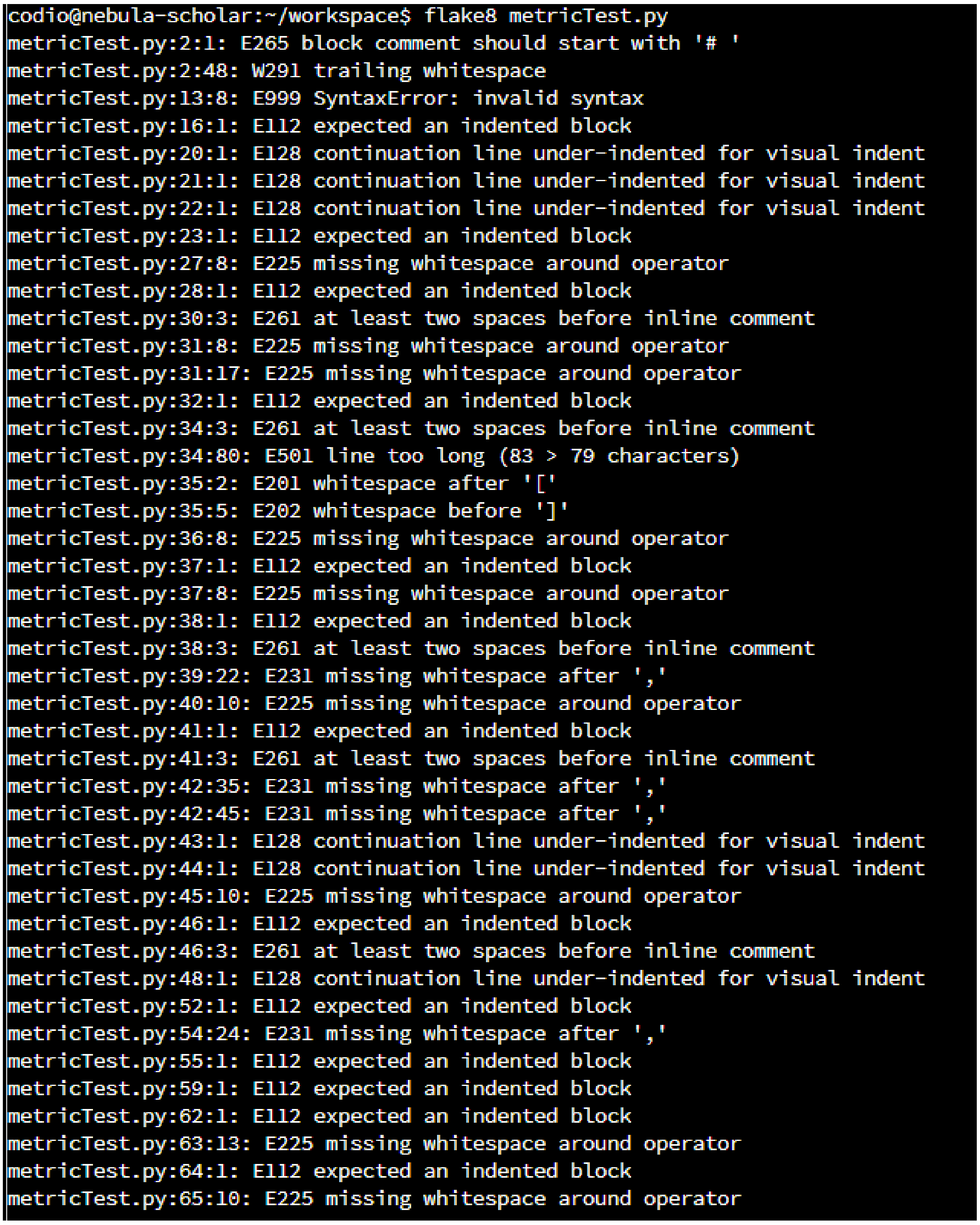
- **Review the errors returned. In what way does this error message differ from the error message returned by pylint?**



```
codio@nebula-scholar:~/workspace$ flake8 pylintTest.py
pylintTest.py:5:1: W293 blank line contains whitespace
pylintTest.py:12:3: E111 indentation is not a multiple of 4
pylintTest.py:14:7: E111 indentation is not a multiple of 4
pylintTest.py:16:7: E111 indentation is not a multiple of 4
pylintTest.py:17:7: E111 indentation is not a multiple of 4
pylintTest.py:17:14: E225 missing whitespace around operator
pylintTest.py:19:7: E111 indentation is not a multiple of 4
pylintTest.py:23:11: E111 indentation is not a multiple of 4
pylintTest.py:24:11: E111 indentation is not a multiple of 4
pylintTest.py:26:14: W292 no newline at end of file
```

Flake8 appears to focus on indentation and whitespace only while Pylint addresses variable case, variable scope, file naming style, missing docstrings in addition to indentation and whitespace

Run flake8 on metricTest.py.

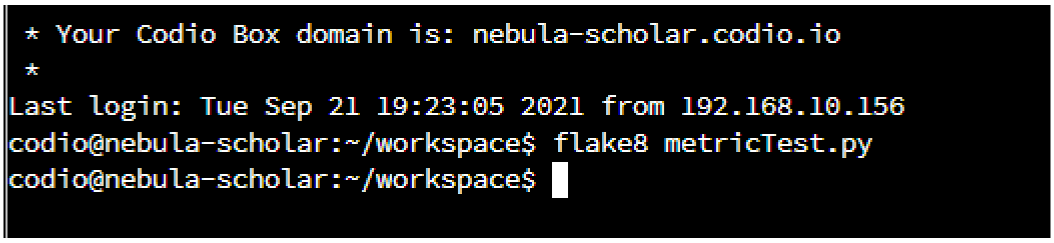


```
codio@nebula-scholar:~/workspace$ flake8 metricTest.py
metricTest.py:2:1: E265 block comment should start with '# '
metricTest.py:2:48: W291 trailing whitespace
metricTest.py:13:8: E999 SyntaxError: invalid syntax
metricTest.py:16:1: E112 expected an indented block
metricTest.py:20:1: E128 continuation line under-indented for visual indent
metricTest.py:21:1: E128 continuation line under-indented for visual indent
metricTest.py:22:1: E128 continuation line under-indented for visual indent
metricTest.py:23:1: E112 expected an indented block
metricTest.py:27:8: E225 missing whitespace around operator
metricTest.py:28:1: E112 expected an indented block
metricTest.py:30:3: E261 at least two spaces before inline comment
metricTest.py:31:8: E225 missing whitespace around operator
metricTest.py:31:17: E225 missing whitespace around operator
metricTest.py:32:1: E112 expected an indented block
metricTest.py:34:3: E261 at least two spaces before inline comment
metricTest.py:34:80: E501 line too long (83 > 79 characters)
metricTest.py:35:2: E201 whitespace after '['
metricTest.py:35:5: E202 whitespace before ']'
metricTest.py:36:8: E225 missing whitespace around operator
metricTest.py:37:1: E112 expected an indented block
metricTest.py:37:8: E225 missing whitespace around operator
metricTest.py:38:1: E112 expected an indented block
metricTest.py:38:3: E261 at least two spaces before inline comment
metricTest.py:39:22: E231 missing whitespace after ','
metricTest.py:40:10: E225 missing whitespace around operator
metricTest.py:41:1: E112 expected an indented block
metricTest.py:41:3: E261 at least two spaces before inline comment
metricTest.py:42:35: E231 missing whitespace after ','
metricTest.py:42:45: E231 missing whitespace after ','
metricTest.py:43:1: E128 continuation line under-indented for visual indent
metricTest.py:44:1: E128 continuation line under-indented for visual indent
metricTest.py:45:10: E225 missing whitespace around operator
metricTest.py:46:1: E112 expected an indented block
metricTest.py:46:3: E261 at least two spaces before inline comment
metricTest.py:48:1: E128 continuation line under-indented for visual indent
metricTest.py:52:1: E112 expected an indented block
metricTest.py:54:24: E231 missing whitespace after ','
metricTest.py:55:1: E112 expected an indented block
metricTest.py:59:1: E112 expected an indented block
metricTest.py:62:1: E112 expected an indented block
metricTest.py:63:13: E225 missing whitespace around operator
metricTest.py:64:1: E112 expected an indented block
metricTest.py:65:10: E225 missing whitespace around operator
```

- **Can you correct each of the errors returned by flake8?**

  Yes

- **What amendments have you made to the code?**
  - A lot of indentation errors were corrected
  - Removed or inserted whitespaces where necessary
  - Reduced long lines by breaking them into 2 lines
  - Fixed one syntax error



```
 * Your Codio Box domain is: nebula-scholar.codio.io
 *
Last login: Tue Sep 21 19:23:05 2021 from 192.168.10.156
codio@nebula-scholar:~/workspace$ flake8 metricTest.py
codio@nebula-scholar:~/workspace$ 
```

Amended code

```python
# CODE SOURCE: SOFTWARE ARCHITECTURE WITH PYTHON
"""
Module metricTest.py
Metric example - Module which is used as a testbed for static checkers.
This is a mix of different functions and classes doing different things.
"""
import random


def fn(x, y):
    """ A function which performs a sum """
    return x + y


def find_optimal_route(start_time, expected_time, favorite_route='SBS1K',
                       favorite_option='bus'):
    d = (expected_time - start_time).total_seconds()/60.0
    if d <= 30:
        return 'car'
    # If d>30 but <45, first drive then take metro
    if d > 30 and d < 45:
        return ('car', 'metro')
    # If d>45 there are a combination of options
    if d > 45:
        if d < 60:
            # First volvo,then connecting bus
            return ('bus:335E', 'bus:connector')
        elif d > 80:
            # Might as well go by normal bus
            return random.choice(('bus:330', 'bus:331',
                                  ':'.join((favorite_option, favorite_route))))
        elif d > 90:
            # Relax and choose favorite route
            return ':'.join((favorite_option, favorite_route))


class C(object):
    """ A class which does almost nothing """
    def __init__(self, x, y):
        self.x = x
        self.y = y

    def f(self):
        pass
    def g(self, x, y):
        if self.x > x:
            return self.x + self.y
        elif x > self.x:
            return x + self.y


class D(C):
    """ D class """
    def __init__(self, x):
        self.x = x

    def f(self, x, y):
        if x > y:
            return x - y
        else:
            return x + y

    def g(self, y):
        if self.x > y:
            return self.x + y
        else:
            return y - self.x
```

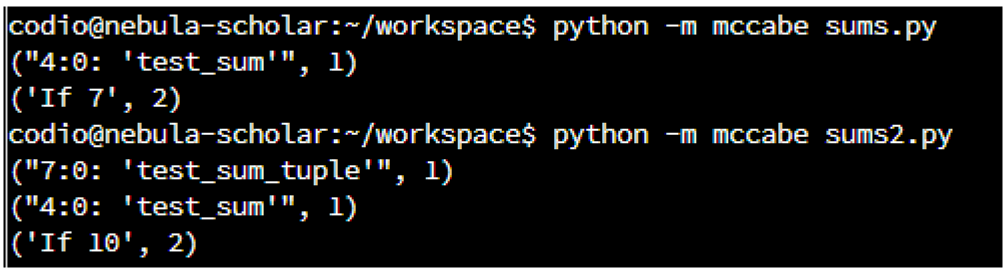
Python: Question 4

Ensure mccabe is in your virtual box -

pip install mccabe

Run mccabe on sums.py. What is the result?

Run mccabe on sums2.py. What is the result?



```
codio@nebula-scholar:~/workspace$ python -m mccabe sums.py
("4:0: 'test_sum'", 1)
('If 7', 2)
codio@nebula-scholar:~/workspace$ python -m mccabe sums2.py
("7:0: 'test_sum_tuple'", 1)
("4:0: 'test_sum'", 1)
('If 10', 2)
```

- **What are the contributors to the cyclomatic complexity in each piece of code?**

If statements contribute to the complexity of a code (Giampedraglia, 2020)

References

Giampedraglia (2020) Available from: https://www.asapdevelopers.com/python-code-complexity/ [Accessed 21 September 2021]