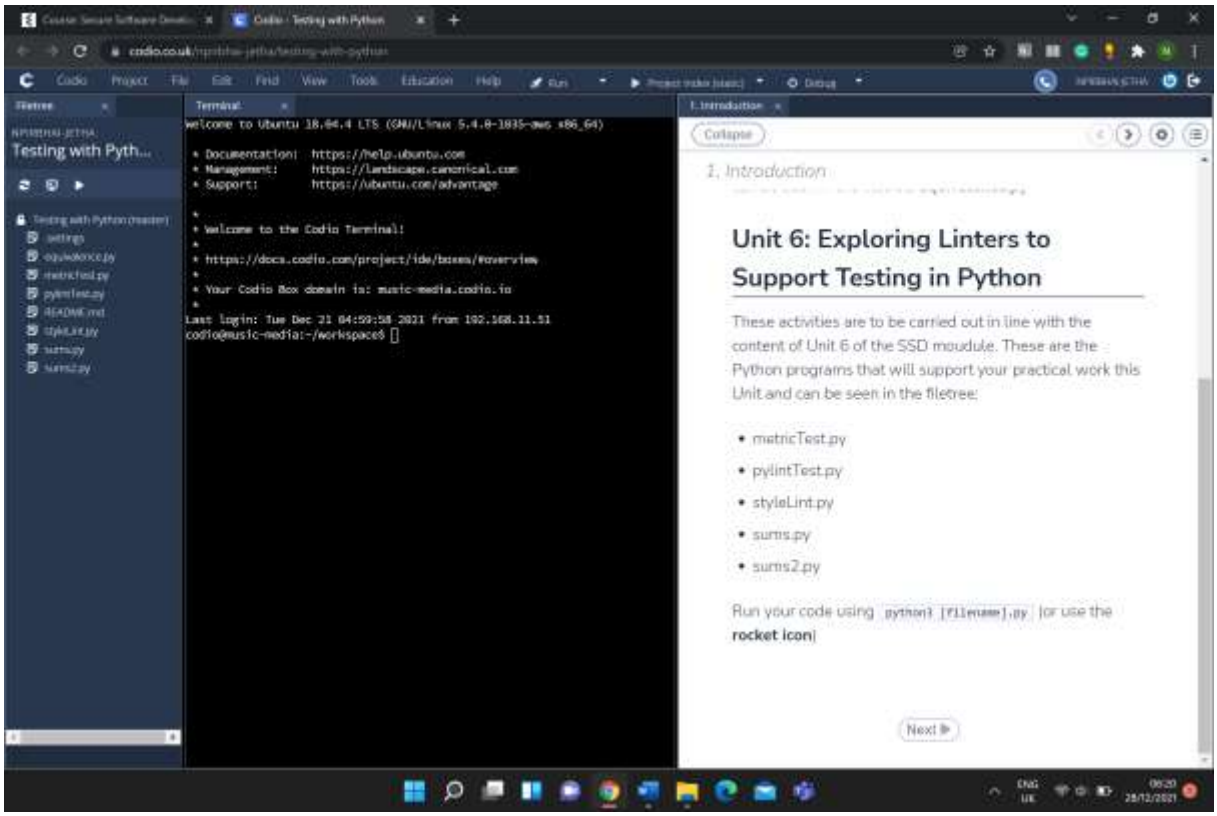
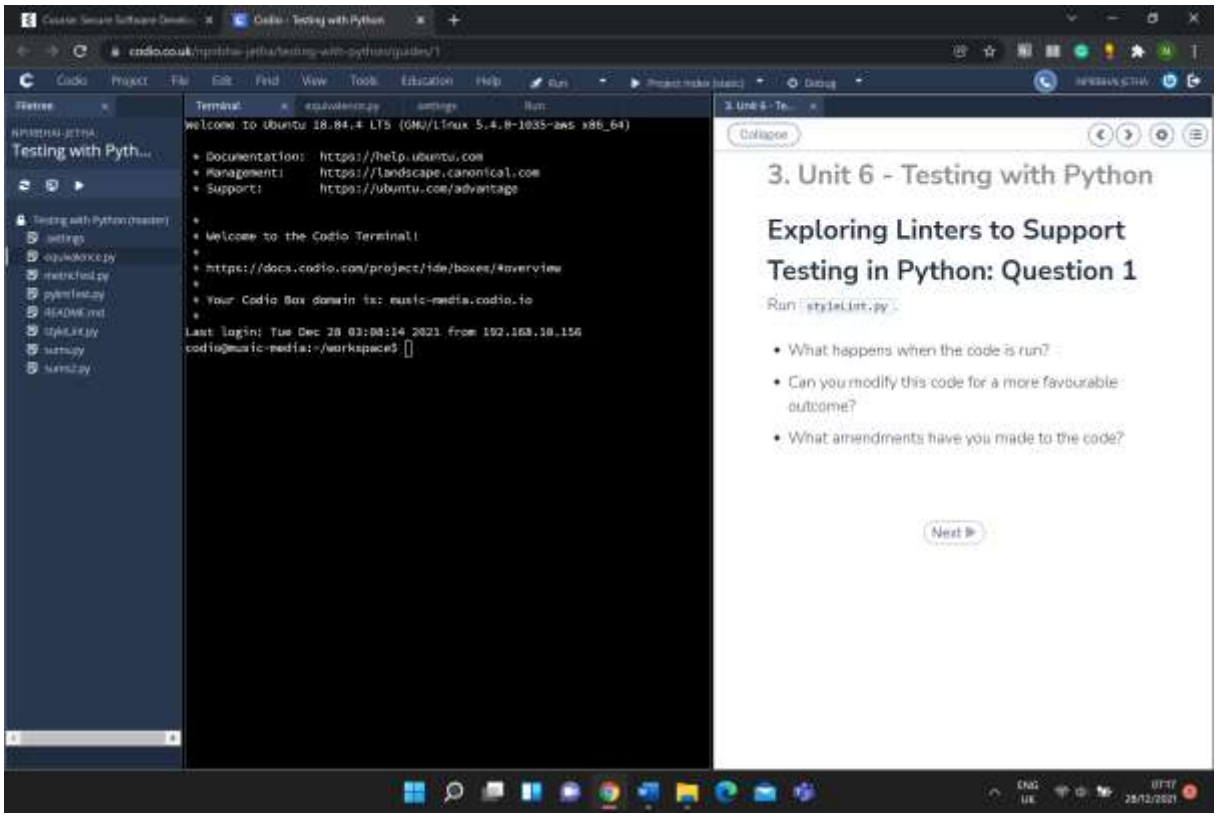
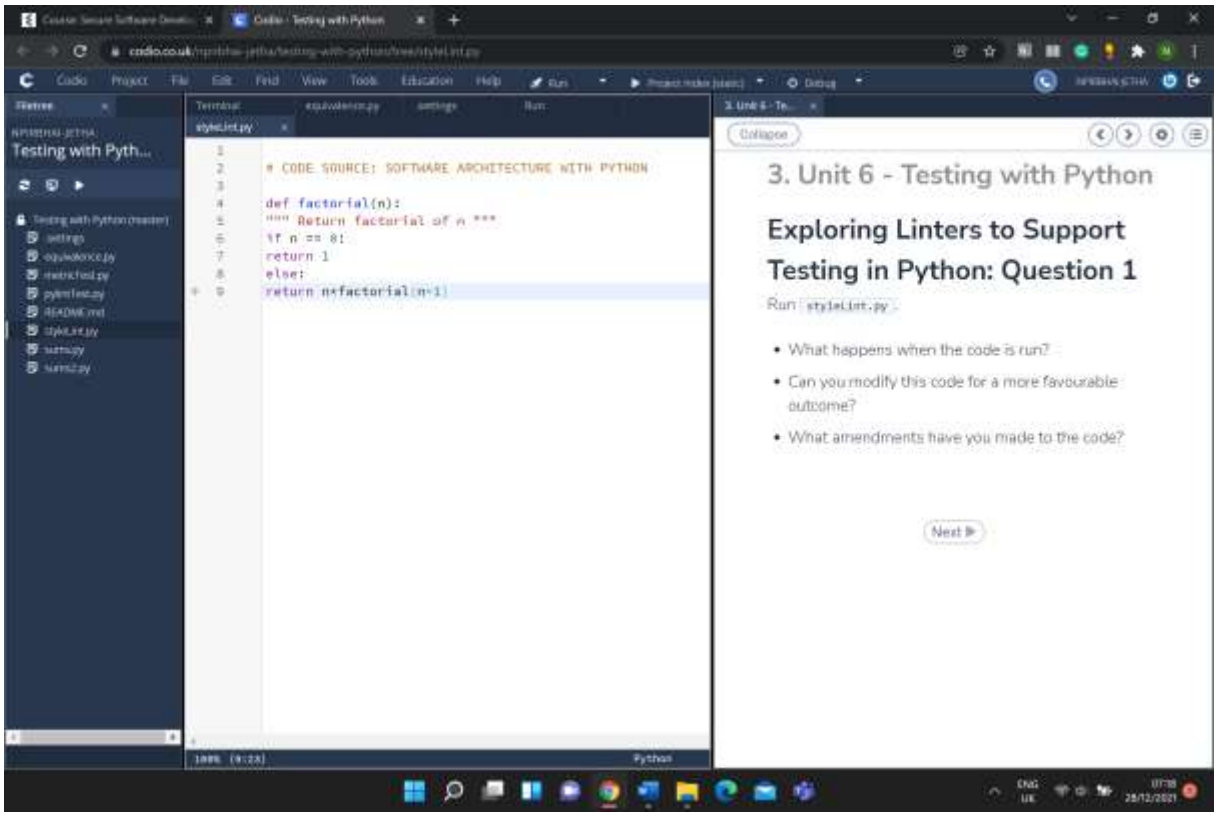
UNIT 6: EXPLORING LINTERS TO SUPPORT TESTING IN PYTHON

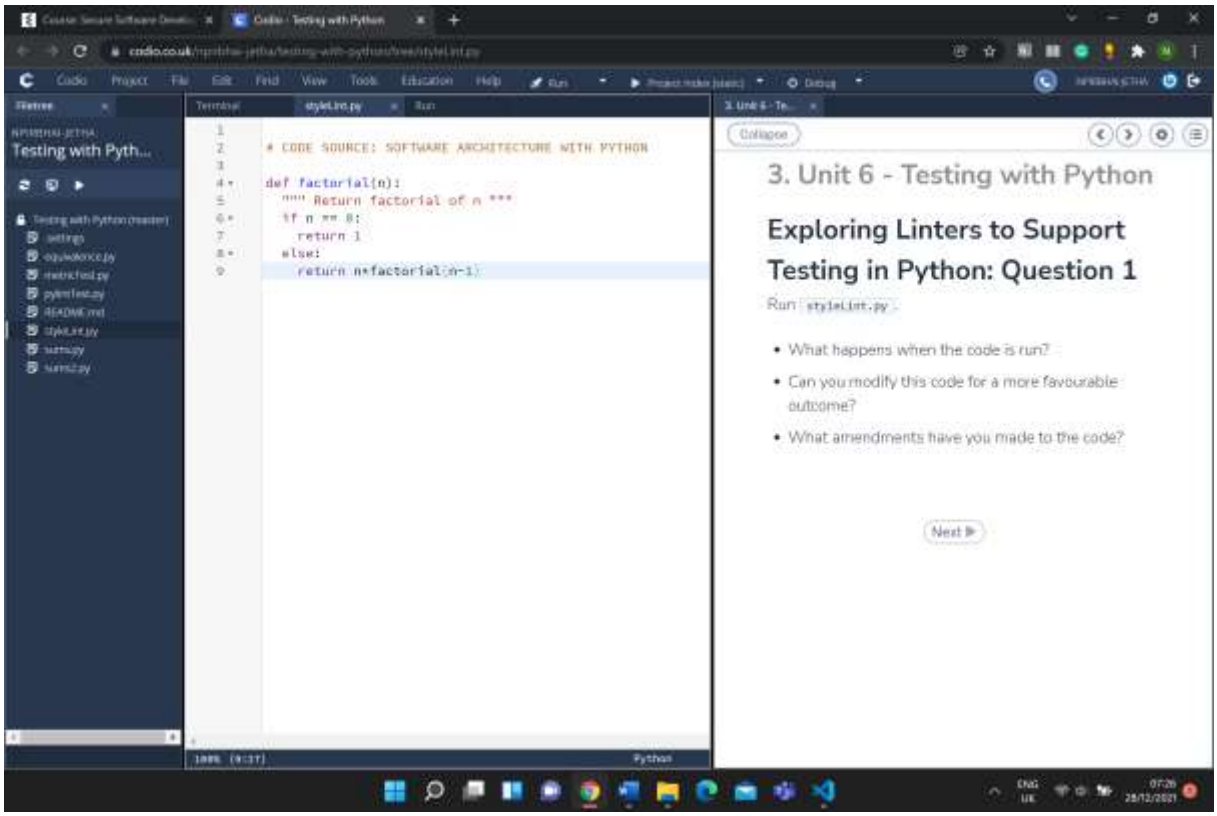


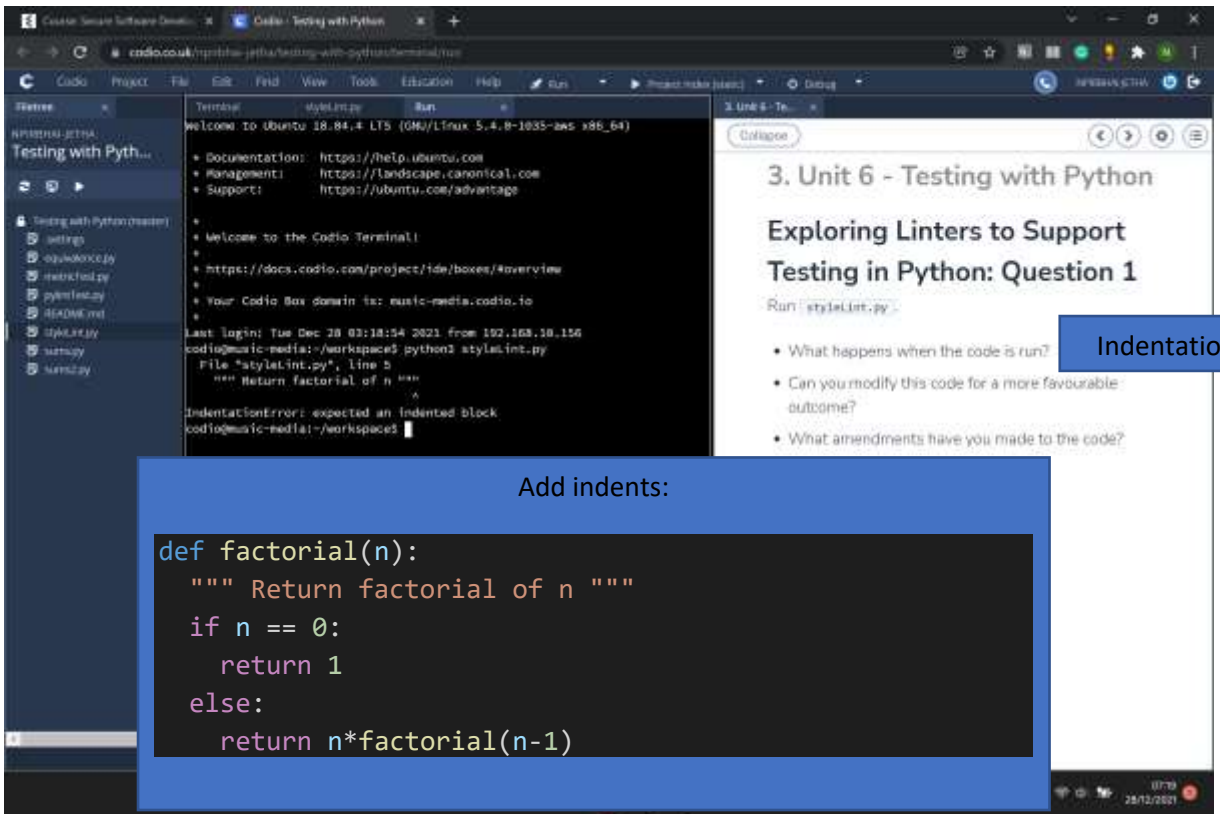


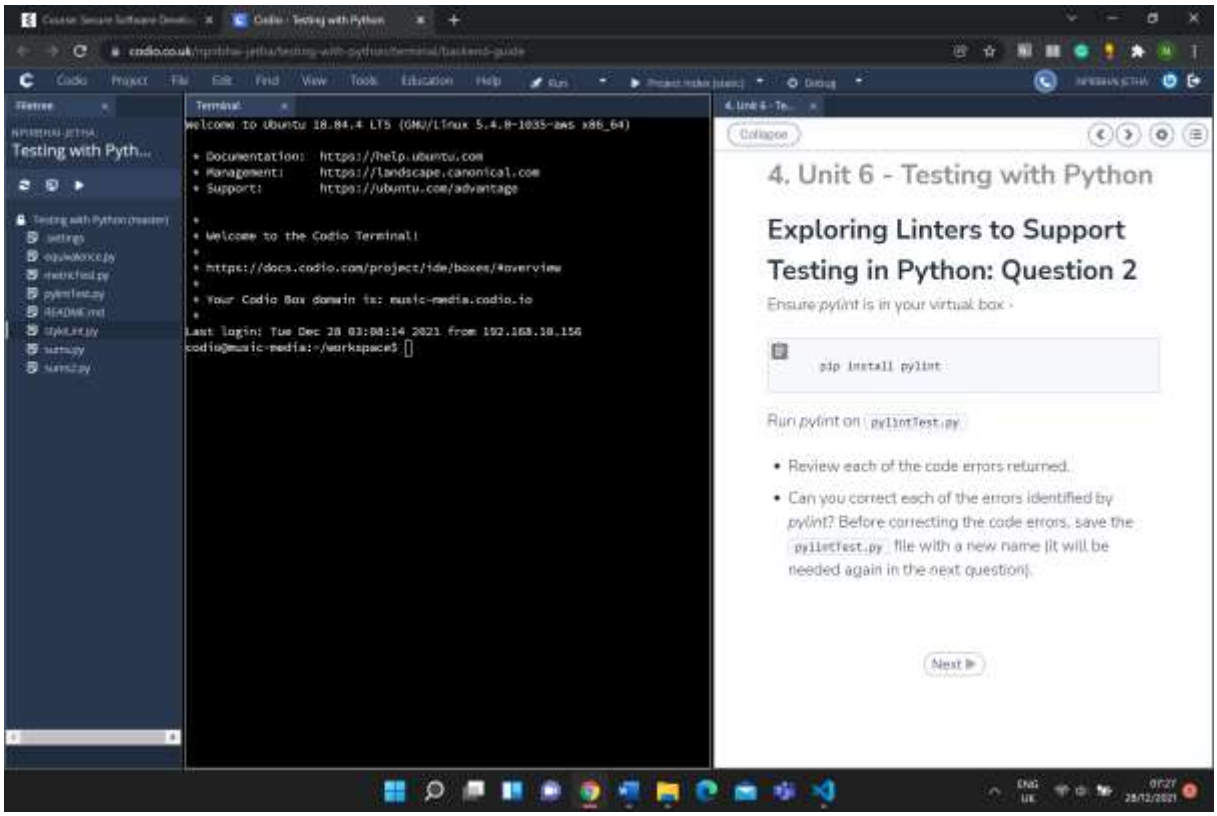
CODE SOURCE: SOFTWARE ARCHITECTURE WITH PYTHON

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




Testing with Pyth...
# CODE SOURCE: SOFTWARE ARCHITECTURE WITH PYTHON
def factorial(n):
""" Return factorial of n """
if n == 0:
return 1
else:
return n*factorial(n-1)
3. Unit 6 - Testing with Python
Exploring Linters to Support Testing in Python: Question 1
Run styleLint.py
What happens when the code is run?
Can you modify this code for a more favourable outcome?
What amendments have you made to the code?
Next

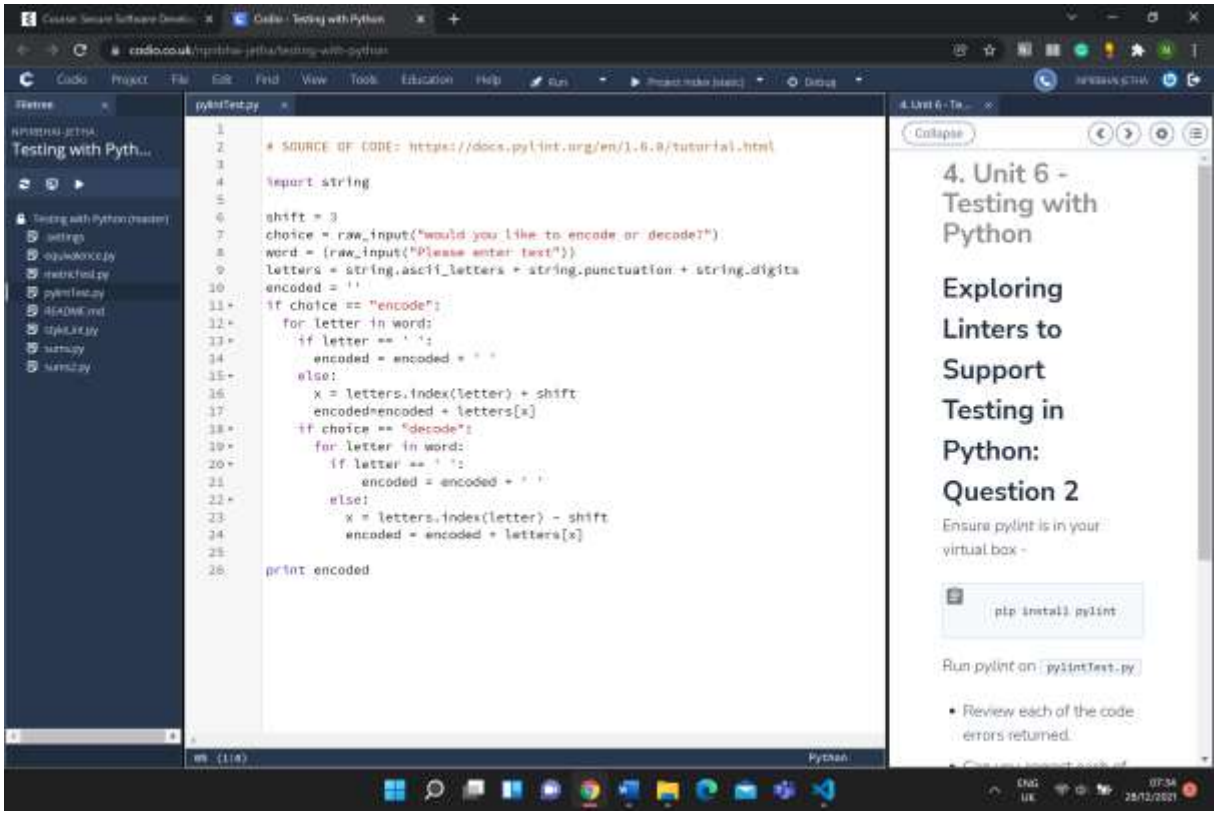

Testing with Pyth...
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 5.4.0-1035-aws x86_64)
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage
* Welcome to the Codio Terminal!
* https://docs.codio.com/project/ide/boxes/#overview
* Your Codio Box domain is: music-media.codio.io
Last login: Tue Dec 28 03:18:54 2021 from 192.168.18.156
codio@music-media:~/workspace$ python3 styleLint.py
File "styleLint.py", line 5
""" Return factorial of n """
^
IndentationError: expected an indented block
codio@music-media:~/workspace$
3. Unit 6 - Testing with Python
Exploring Linters to Support Testing in Python: Question 1
Run styleLint.py
What happens when the code is run?
Can you modify this code for a more favourable outcome?
What amendments have you made to the code?
Indentation Error
Add indents:
def factorial(n):
""" Return factorial of n """
if n == 0:
return 1
else:
return n*factorial(n-1)

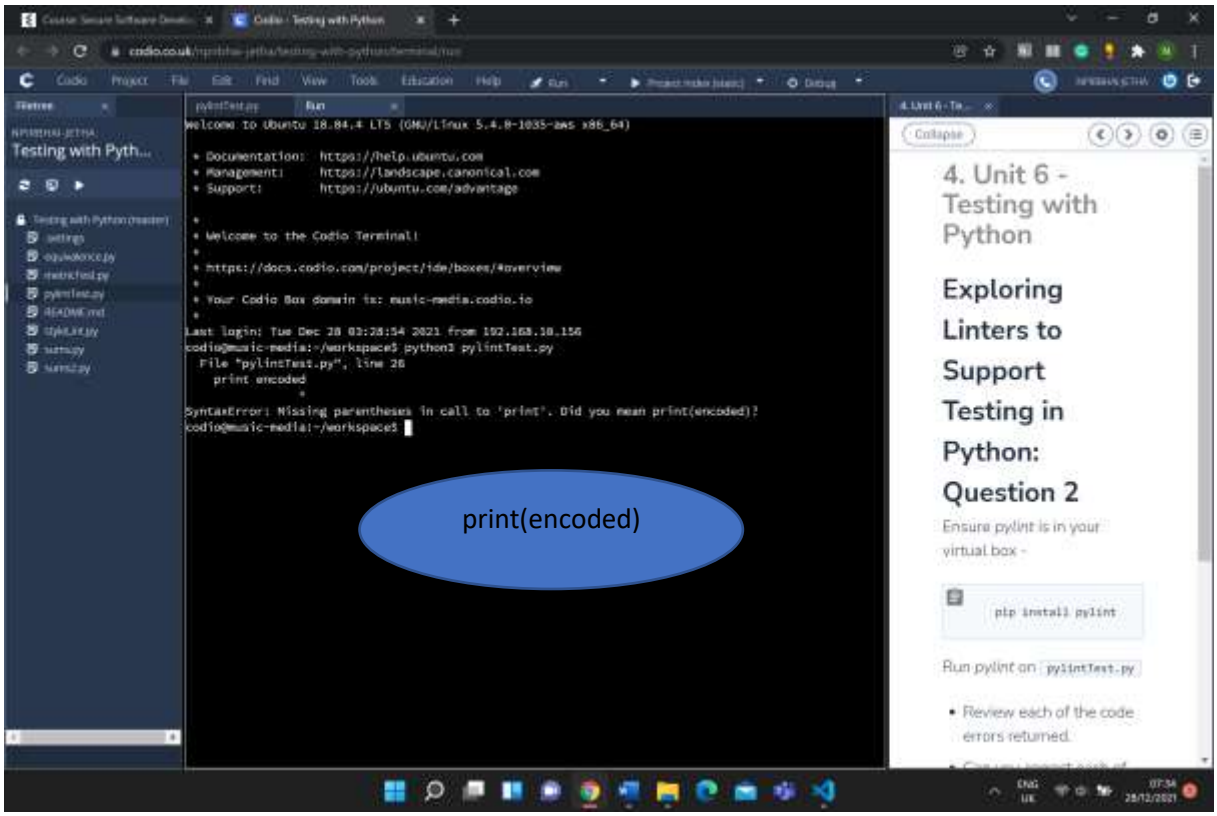


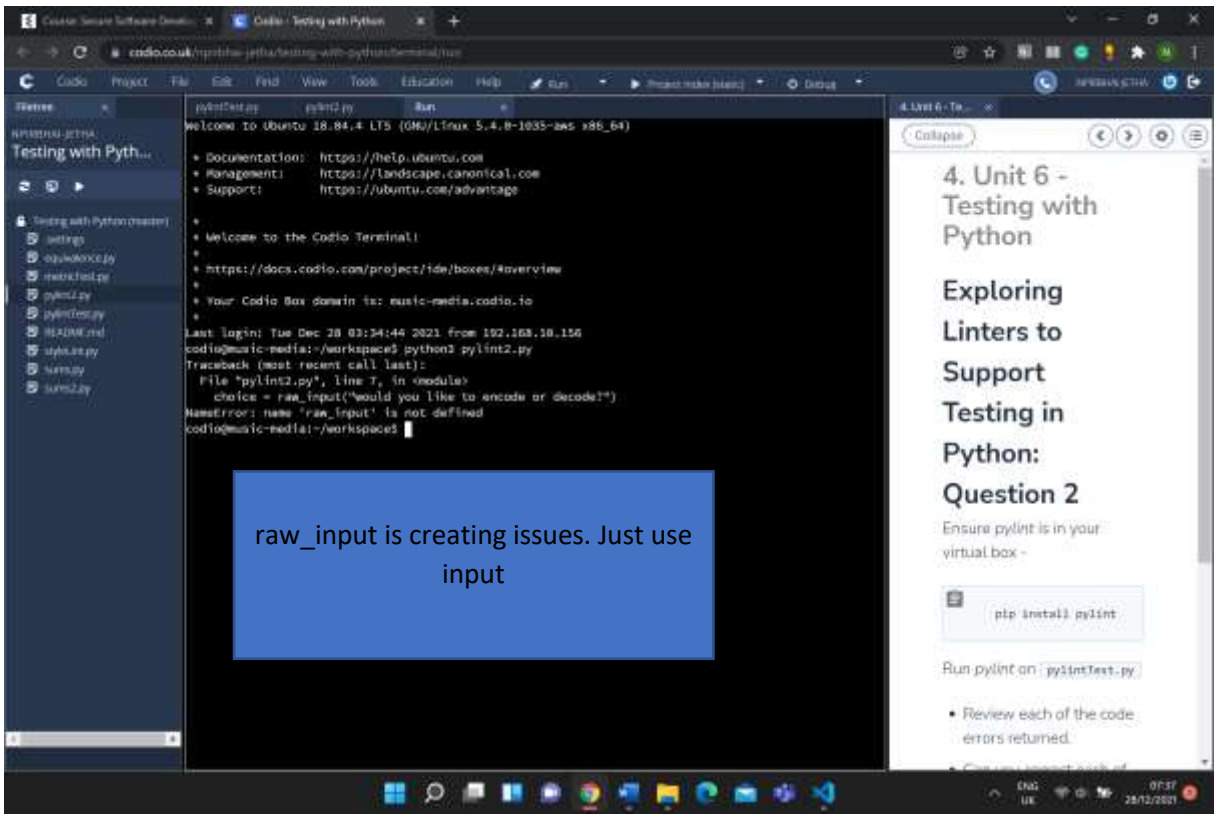
SOURCE OF CODE: https://docs.pylint.org/en/1.6.0/tutorial.html

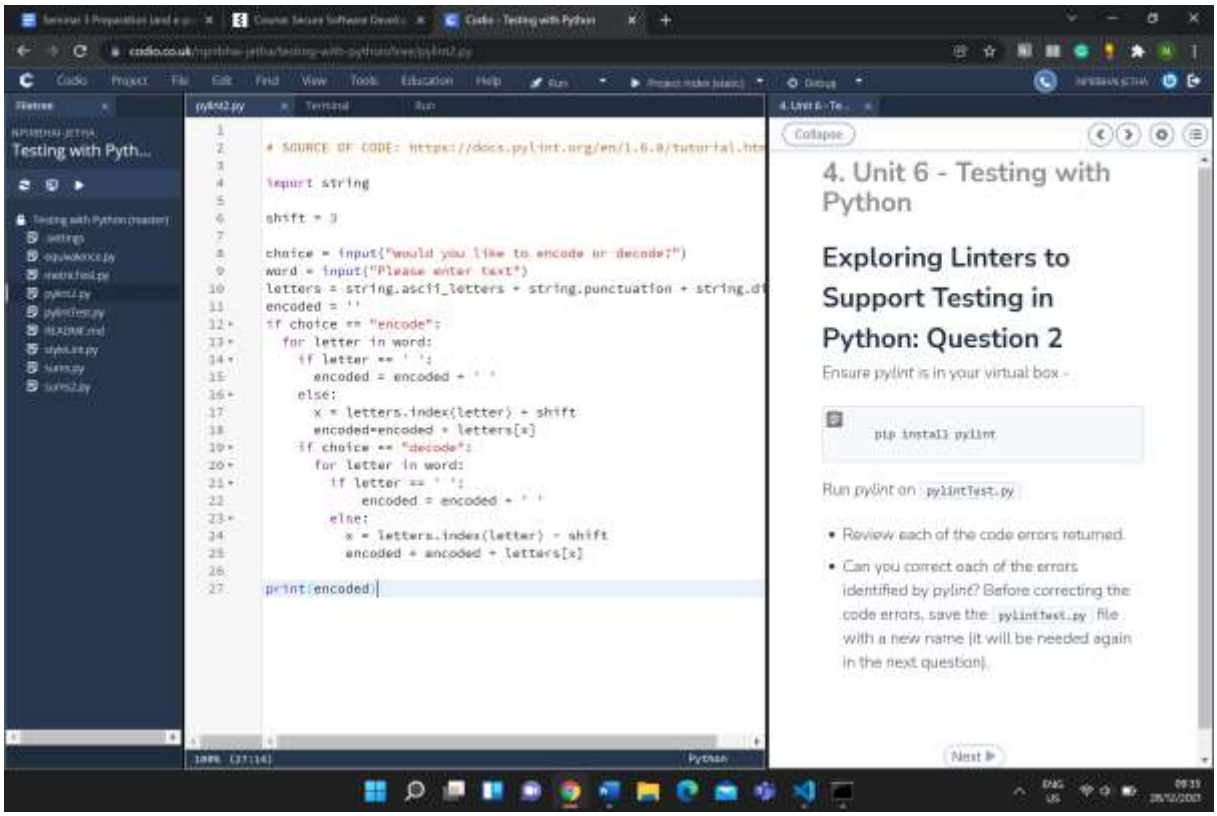
```
import string


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]
```

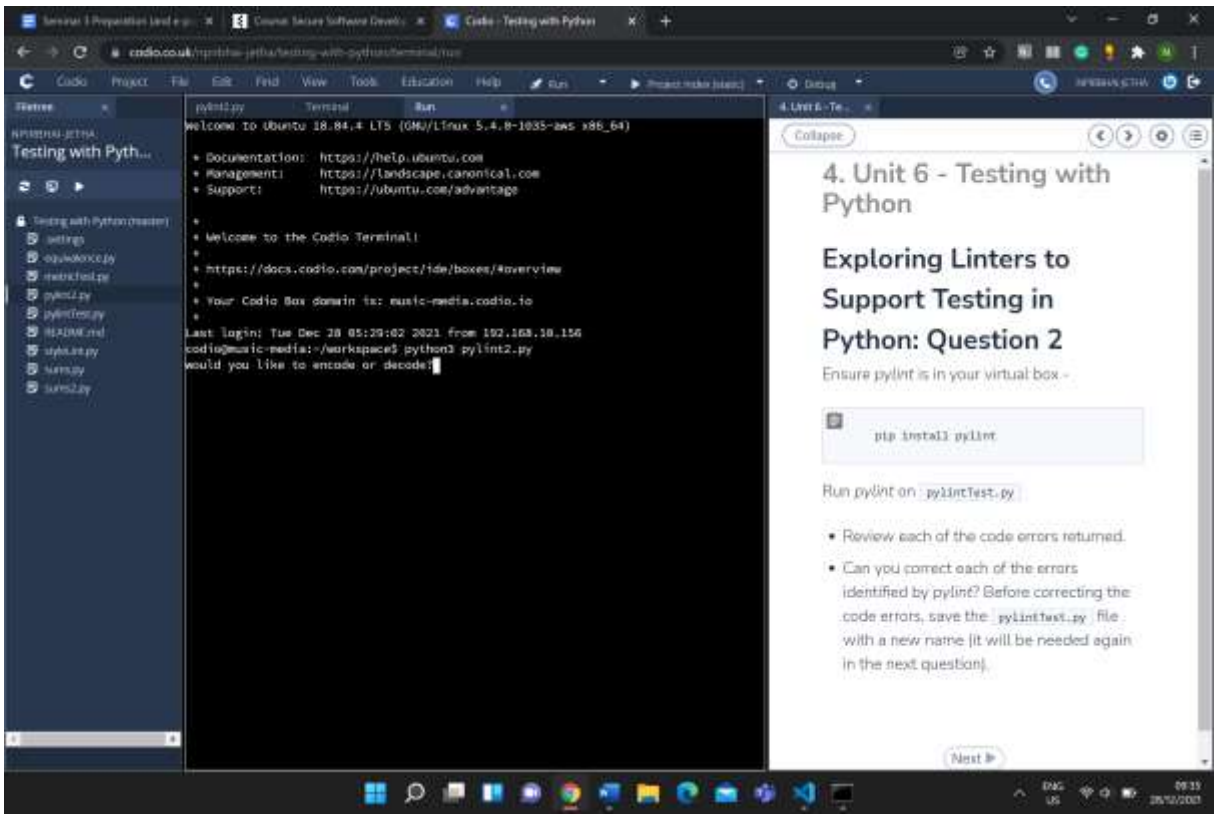
```
    if choice == "decode":
      for letter in word:
        if letter == ' ':
           encoded = encoded + ' '
        else:
          x = letters.index(letter) - shift
          encoded = encoded + letters[x]

print encoded
```

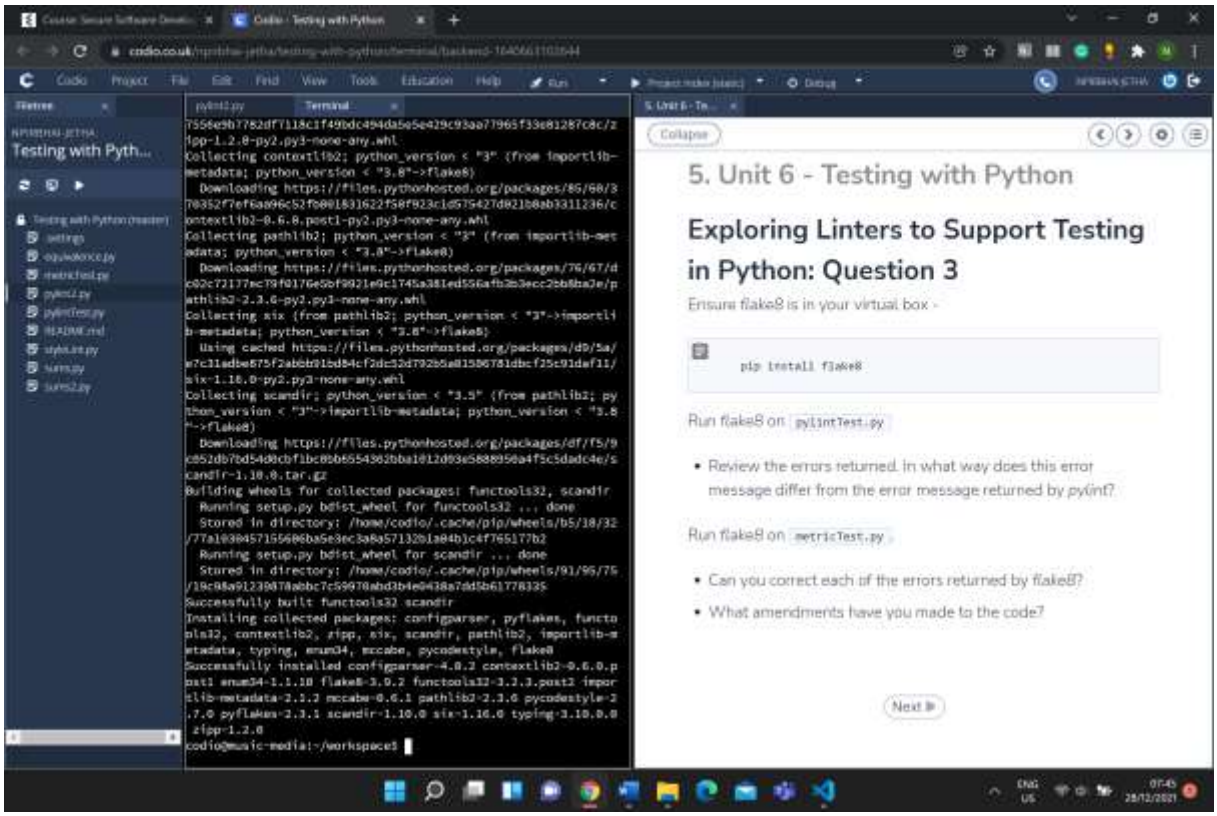



Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 5.4.0-1035-aws x86_64)
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage
* Welcome to the Codio Terminal!
* https://docs.codio.com/project/ide/boxes/#overview
* Your Codio Box domain is: music-media.codio.io
Last login: Tue Dec 28 03:28:54 2021 from 192.168.18.156
codio@music-media:~/workspace$ python3 pylintTest.py
File "pylintTest.py", line 28
print encoded
SyntaxError: Missing parentheses in call to 'print'. Did you mean print(encoded)?
codio@music-media:~/workspace$
print(encoded)
4. Unit 6 - Testing with Python
Exploring Linters to Support Testing in Python: Question 2
Ensure pylint is in your virtual box -
pip install pylint
Run pylint on pylintTest.py
Review each of the code errors returned.

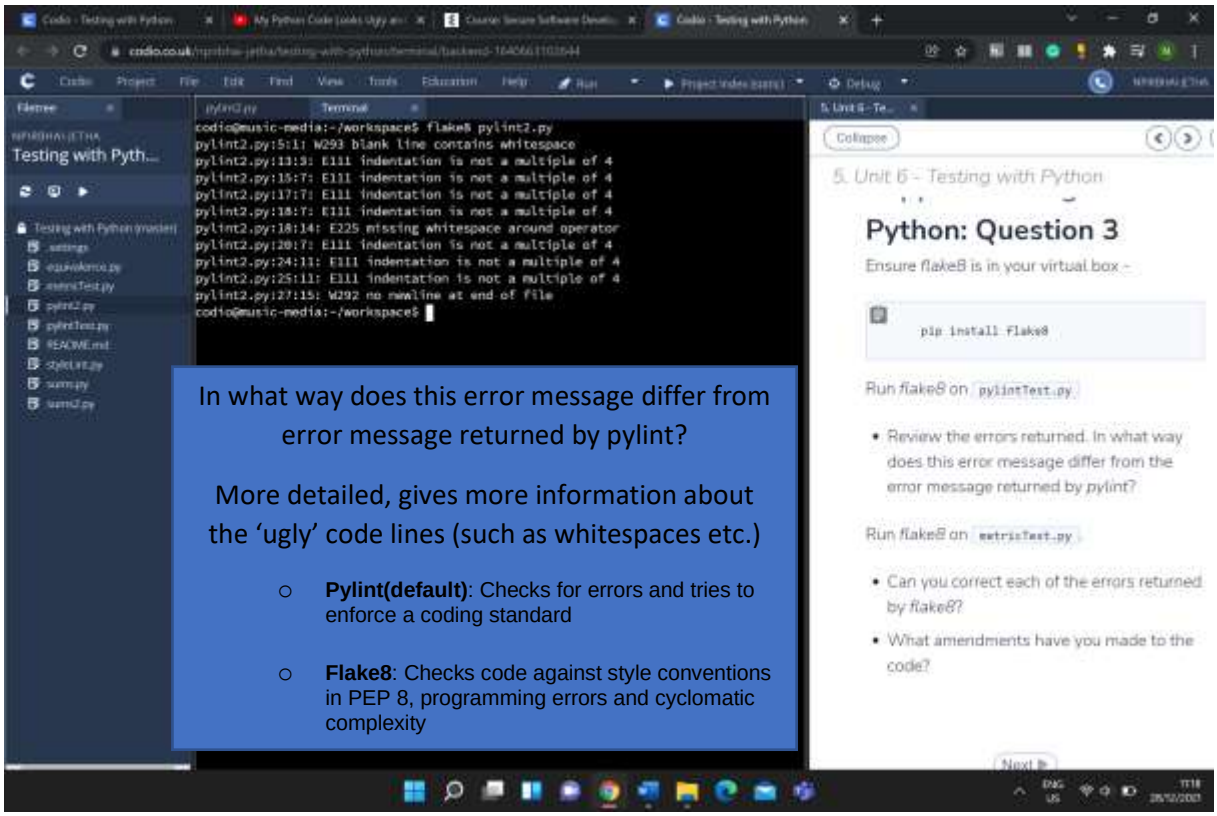

Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 5.4.0-1035-aws x86_64)
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage
* Welcome to the Codio Terminal!
* https://docs.codio.com/project/ide/boxes/#overview
* Your Codio Box domain is: music-media.codio.io
Last login: Tue Dec 28 03:34:44 2021 from 192.168.18.156
codio@music-media:~/workspace$ python3 pylint2.py
Traceback (most recent call last):
File "pylint2.py", line 7, in <module>
choice = raw_input("Would you like to encode or decode?")
NameError: name 'raw_input' is not defined
codio@music-media:~/workspace$
raw_input is creating issues. Just use input
4. Unit 6 - Testing with Python
Exploring Linters to Support Testing in Python: Question 2
Ensure pylint is in your virtual box -
pip install pylint
Run pylint on pylintTest.py
Review each of the code errors returned.

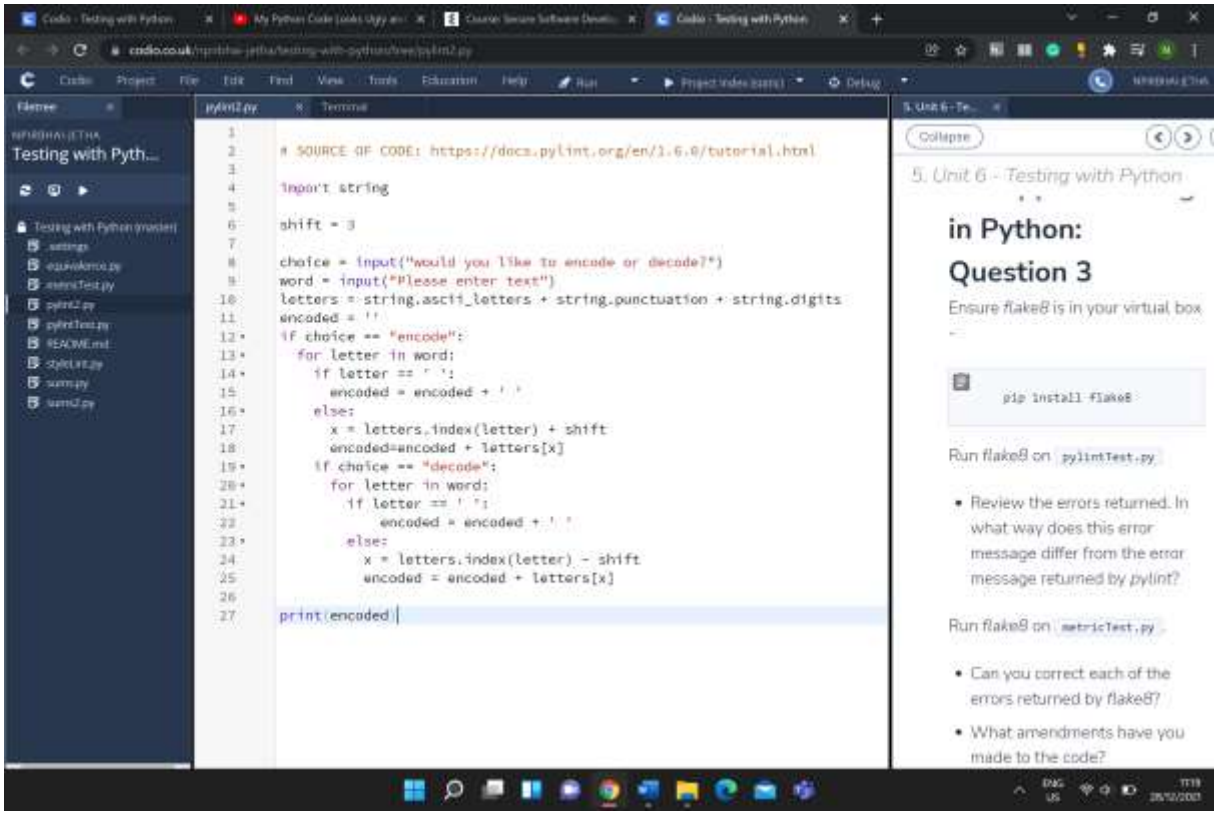

Testing with Pyth...
pylint2.py
# SOURCE OF CODE: https://docs.pylint.org/en/1.6.0/tutorial.ht
import string
shift = 3
choice = input("would you like to encode or decode?")
word = input("Please enter text")
letters = string.ascii_letters + string.punctuation + string.d
encoded = ''
if choice == "encode":
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)
4. Unit 6 - Testing with Python
Exploring Linters to Support Testing in Python: Question 2
Ensure pylint is in your virtual box -
pip install pylint
Run pylint on pylintTest.py
Review each of the code errors returned.
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).
Next


Testing with Pyth...
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 5.4.0-1035-aws x86_64)
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage
* Welcome to the Codio Terminal!
* https://docs.codio.com/project/ide/boxes/#overview
* Your Codio Box domain is: music-media.codio.io
Last login: Tue Dec 28 05:29:02 2021 from 192.168.10.156
codio@music-media:~/workspace$ python3 pylint2.py
would you like to encode or decode?
4. Unit 6 - Testing with Python
Exploring Linters to Support Testing in Python: Question 2
Ensure pylint is in your virtual box -
pip install pylint
Run pylint on pylintTest.py
Review each of the code errors returned.
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).
Next



flake8 pylint2.py

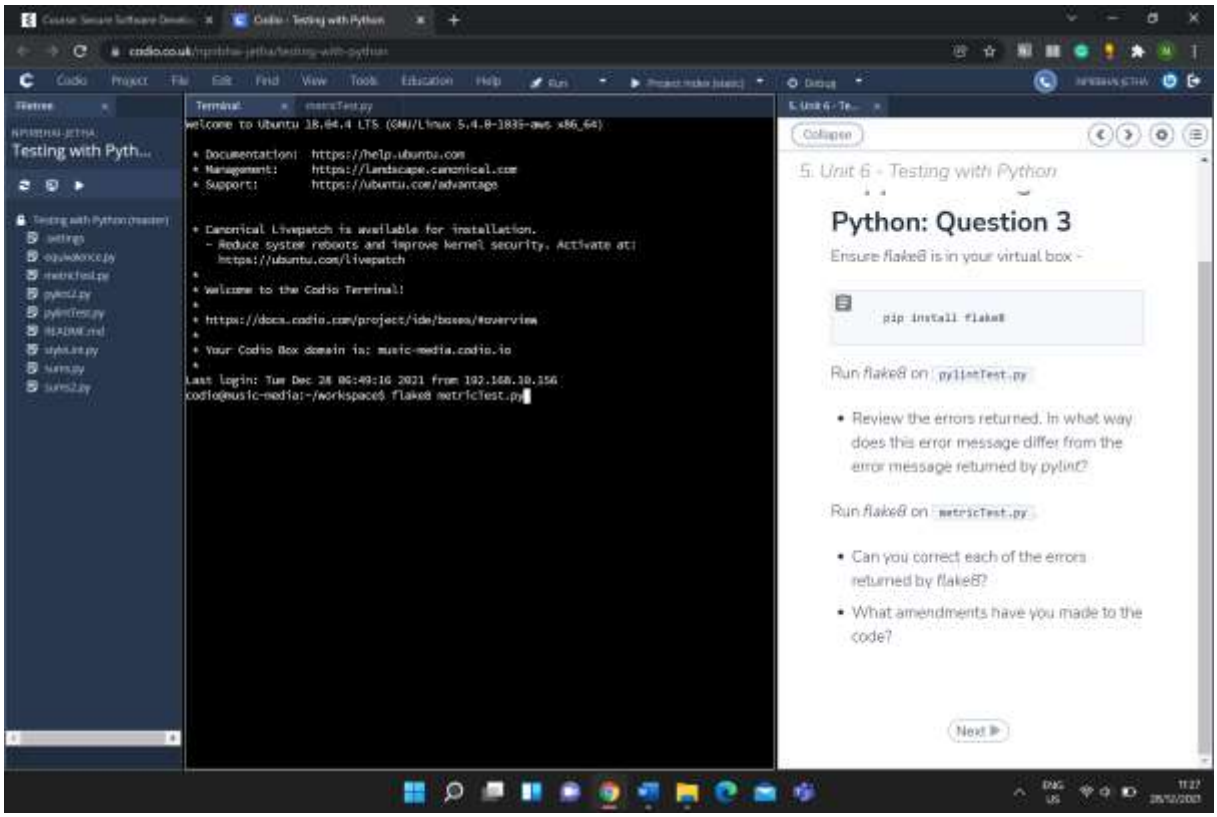


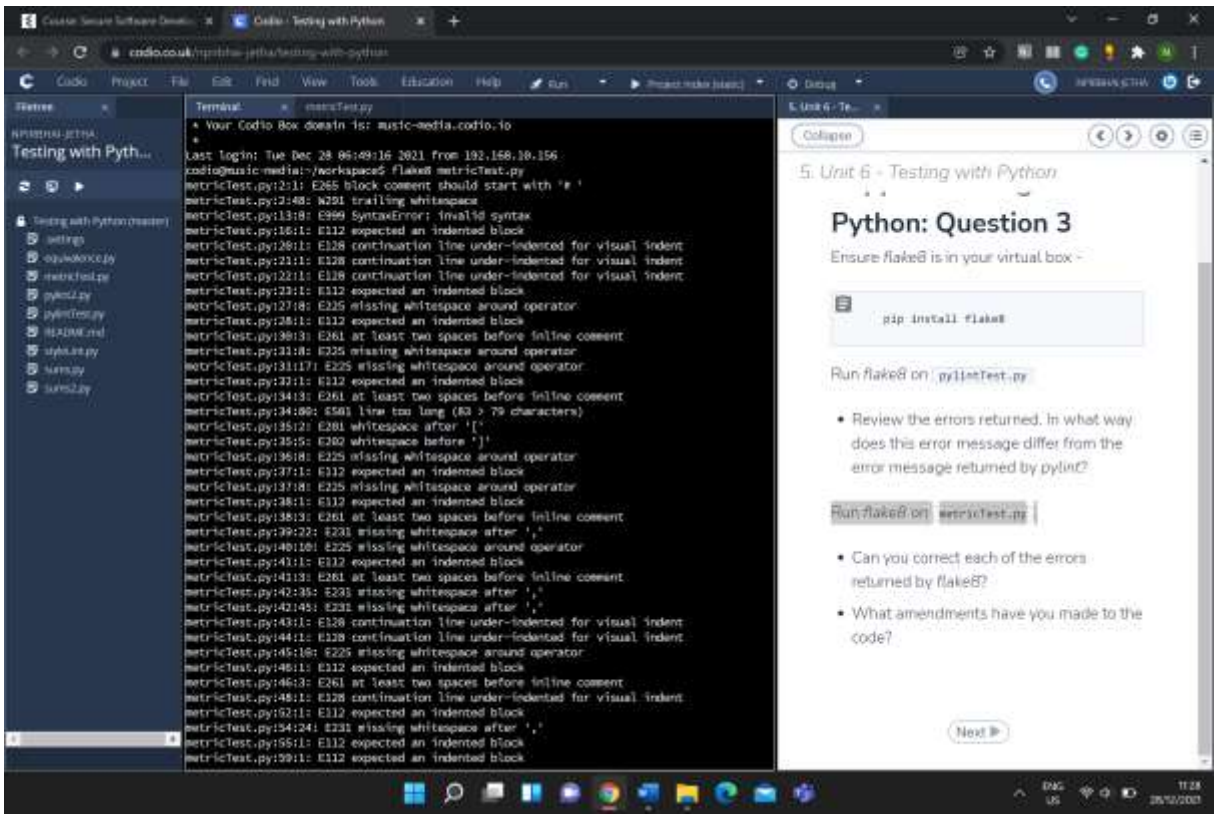


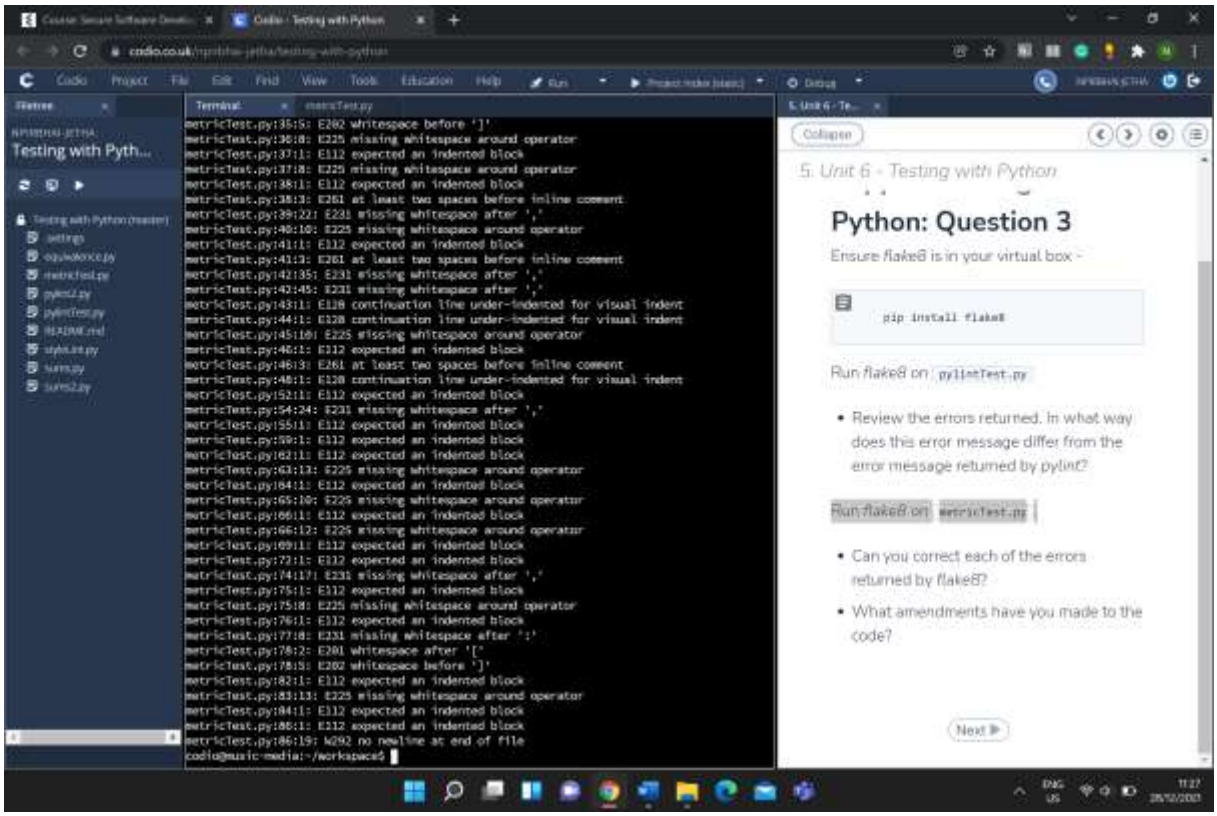
CHECK THIS YOUTUBE VIDEO FOR FLAKE8

https://www.youtube.com/watch?v=TDUf93vqq3g

Run *flake8* on `metricTest.py`.





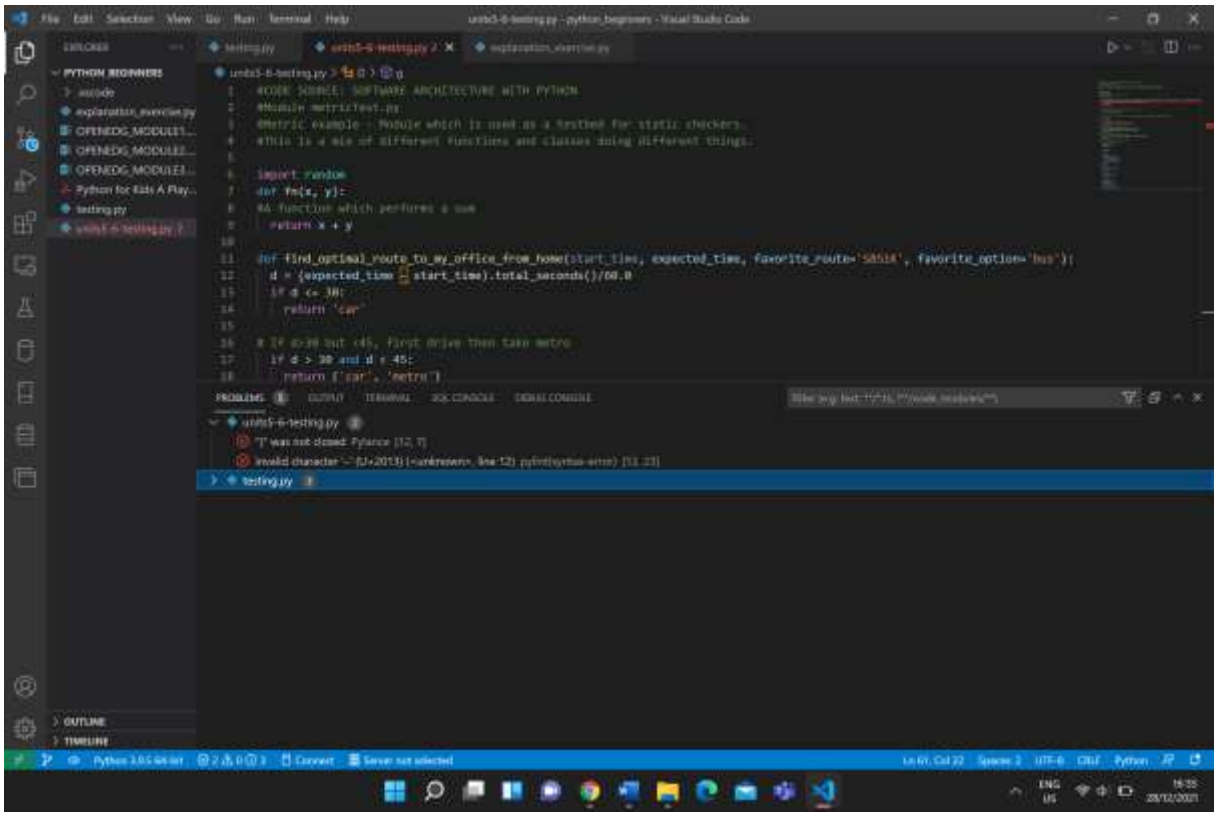


Amendments made:

- Remove all numberings
- Check all indents

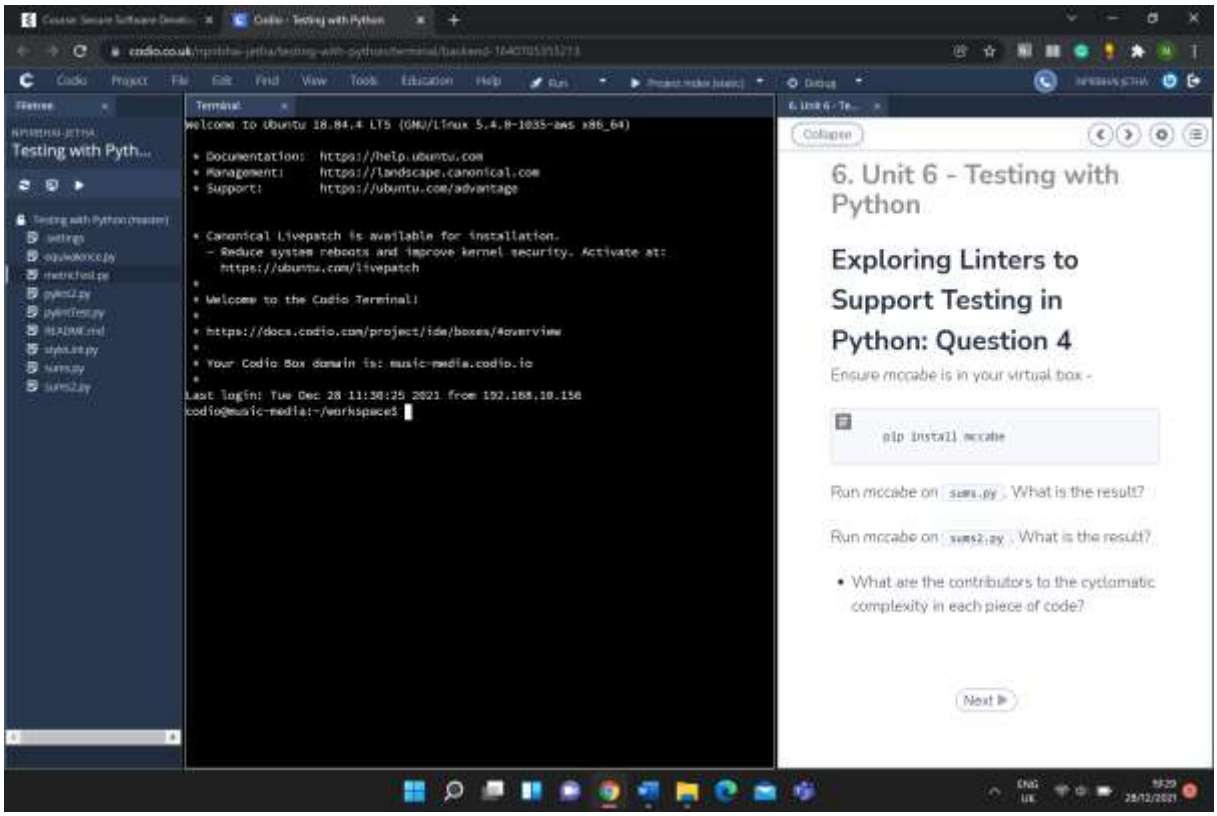
For "**Indentation is not a multiple of four (E111)",** PEP8 recommends that Python code indentation be a multiple of four. [https://www.python.org/dev/peps/pep-0008/#indentation or https://www.flake8rules.com/rules/E111.html]

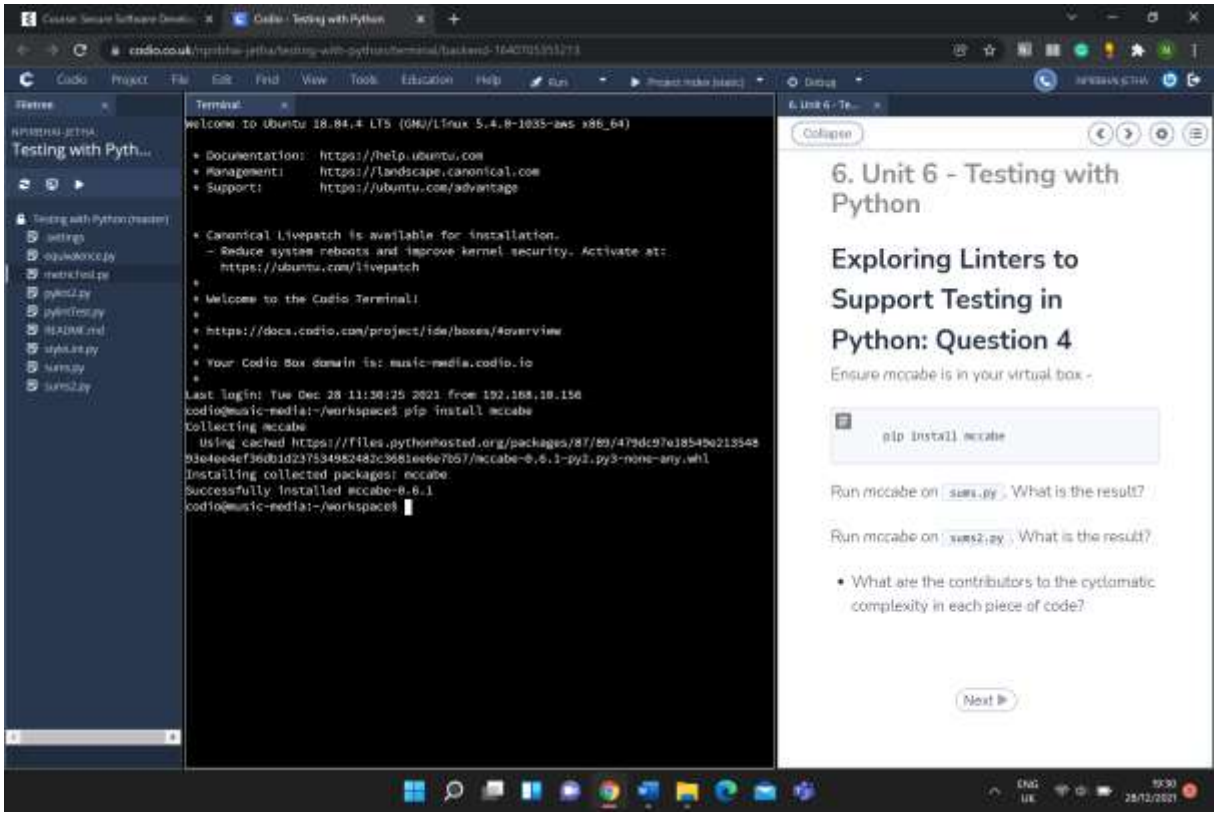
- Minus sign checked

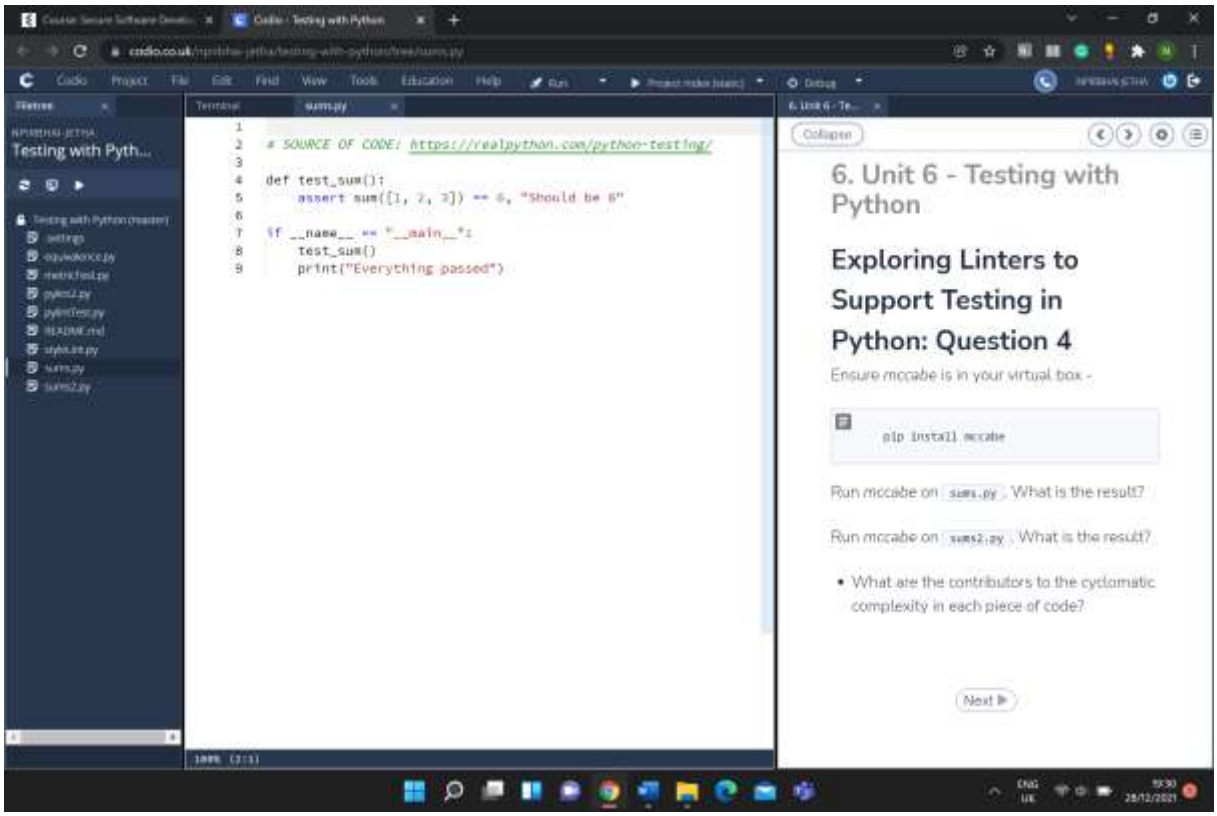


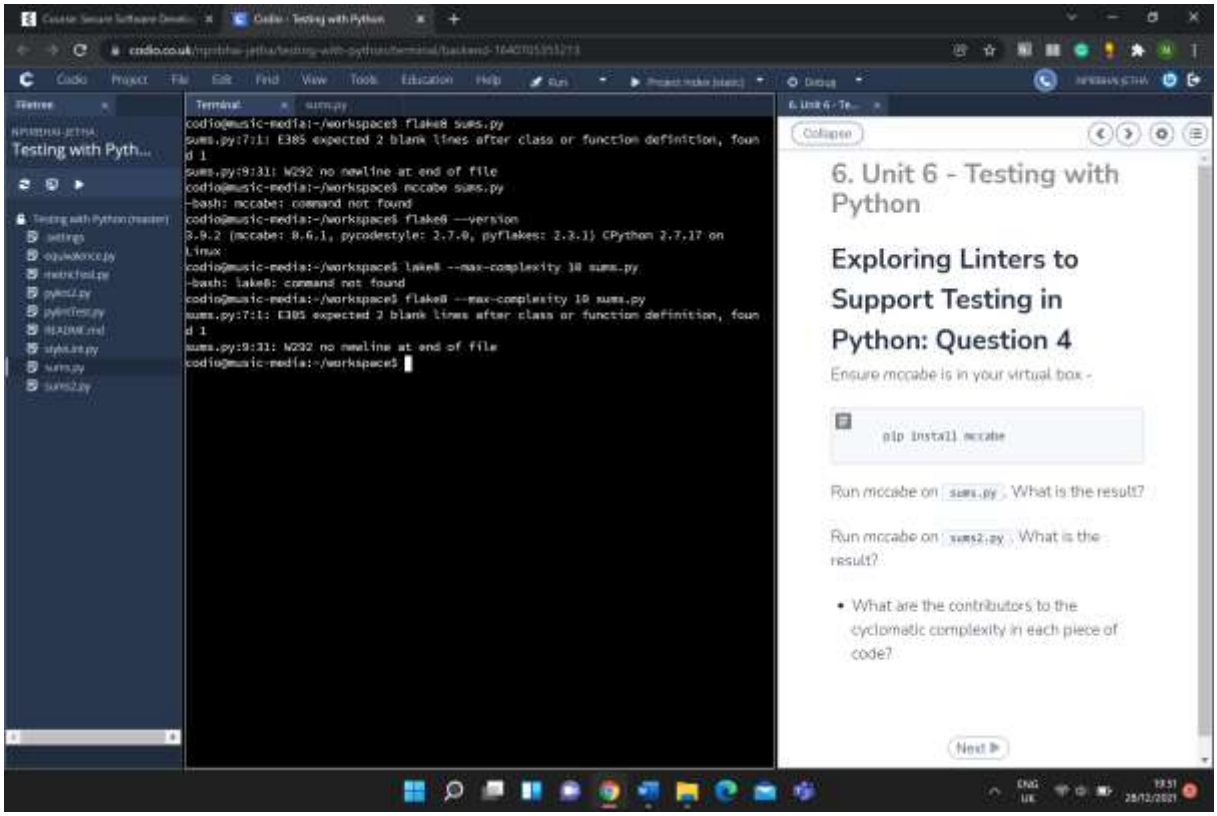
```
#Module metricTest.py
#Metric example - Module which is used as a testbed for static checkers.
#This is a mix of different functions and classes doing different things.

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

def find_optimal_route_to_my_office_from_home(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 optionsWriting Modifiable and Readable
Code
  if d > 45:
    if d < 60:
# First volvo,then connecting bus
      return ('bus:335E','bus:connector')
```

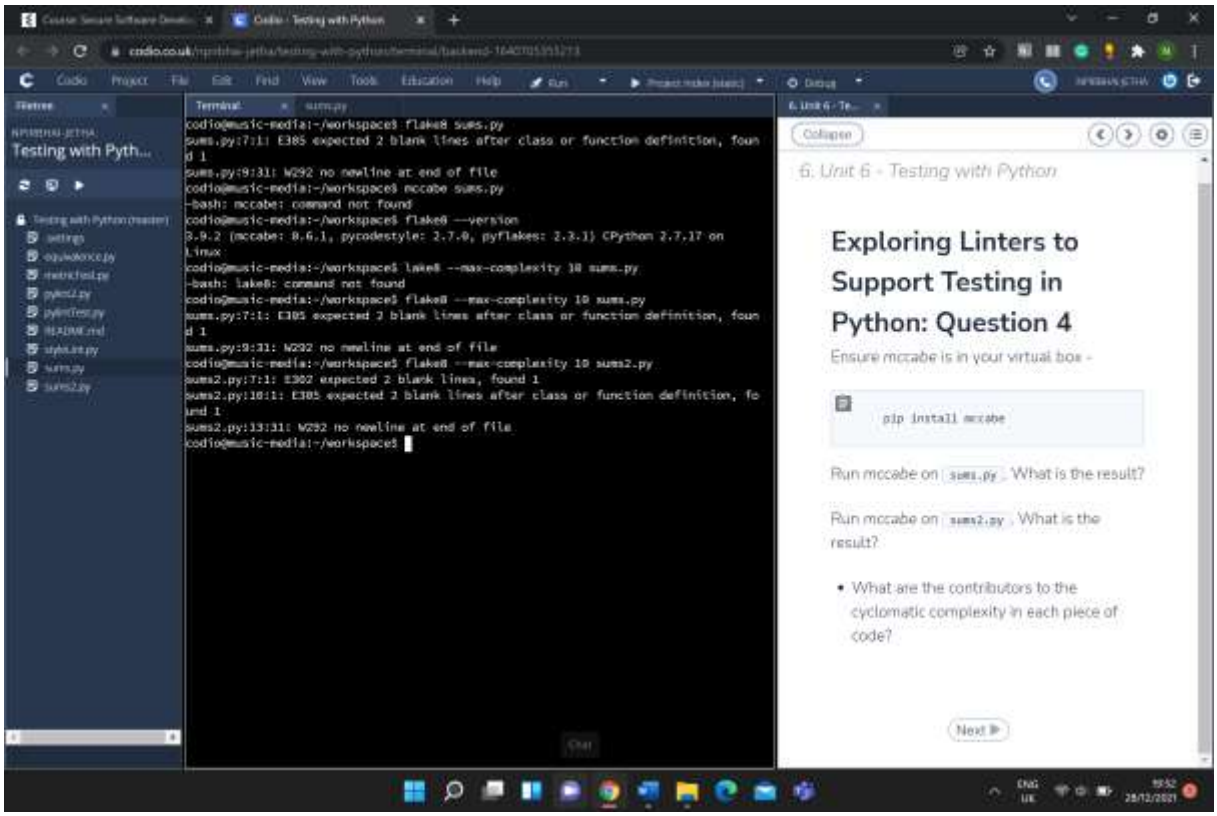
```
  elif d > 80:
# Might as well go by normal bus
    return random.choice(('bus:330','bus:331',':'.
    join((favorite_option, favorite_route))))
  elif d > 90:
# Relax and choose favorite route
    return ':'.join((favorite_option, favorite_route))
class C(object):
  #A class which does almost nothing

  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
```


Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 5.4.0-1035-aws x86_64)
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage
* Canonical Livepatch is available for installation.
- Reduce system reboots and improve kernel security. Activate at:
https://ubuntu.com/livepatch
* Welcome to the Codio Terminal!
* https://docs.codio.com/project/ide/boxes/#overview
* Your Codio Box domain is: music-media.codio.io
Last login: Tue Dec 28 11:30:25 2021 from 192.168.10.156
codio@music-media:~/workspace$
6. Unit 6 - Testing with Python
Exploring Linters to Support Testing in 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?
What are the contributors to the cyclomatic complexity in each piece of code?
Next


Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 5.4.0-1035-aws x86_64)
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage
* Canonical Livepatch is available for installation.
- Reduce system reboots and improve kernel security. Activate at:
https://ubuntu.com/livepatch
* Welcome to the Codio Terminal!
* https://docs.codio.com/project/ide/boxes/#overview
* Your Codio Box domain is: music-media.codio.io
Last login: Tue Dec 28 11:30:25 2021 from 192.168.10.156
codio@music-media:~/workspace$ pip install mccabe
Collecting mccabe
Using cached https://files.pythonhosted.org/packages/87/89/479dc97e18549e21354893e4ee4ef36db1d237534982482c3681ee6e7b57/mccabe-0.6.1-py2.py3-none-any.whl
Installing collected packages: mccabe
Successfully installed mccabe-0.6.1
codio@music-media:~/workspace$
6. Unit 6 - Testing with Python
Exploring Linters to Support Testing in 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?
What are the contributors to the cyclomatic complexity in each piece of code?
Next


sums.py
# SOURCE OF CODE: https://realpython.com/python-testing/
def test_sum():
assert sum([1, 2, 3]) == 6, "Should be 6"
if __name__ == "__main__":
test_sum()
print("Everything passed")
6. Unit 6 - Testing with Python
Exploring Linters to Support Testing in 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?
What are the contributors to the cyclomatic complexity in each piece of code?
Next


codio@music-media:~/workspace$ flake8 sums.py
sums.py:7:1: E305 expected 2 blank lines after class or function definition, found 1
sums.py:9:31: W292 no newline at end of file
codio@music-media:~/workspace$ mccabe sums.py
-bash: mccabe: command not found
codio@music-media:~/workspace$ flake8 --version
3.9.2 (mccabe: 0.6.1, pycodestyle: 2.7.0, pyflakes: 2.3.1) CPython 2.7.17 on Linux
codio@music-media:~/workspace$ lake8 --max-complexity 10 sums.py
-bash: lake8: command not found
codio@music-media:~/workspace$ flake8 --max-complexity 10 sums.py
sums.py:7:1: E305 expected 2 blank lines after class or function definition, found 1
sums.py:9:31: W292 no newline at end of file
codio@music-media:~/workspace$
6. Unit 6 - Testing with Python
Exploring Linters to Support Testing in 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?
What are the contributors to the cyclomatic complexity in each piece of code?
Next



7. Unit 6 - Testing with Python

Exploring Linters to Support Testing in Python: Question 5 (e-Portfolio Entry)

From Section 5 of the Firdaus et al (2014) reading, select a test technique from the following categories:

- Specification-based techniques
- Structure-based techniques
- Experience-based techniques

Discuss the scenario(s) in which each technique would be important to be used

• Specification-based techniques	• Structure-based techniques	• Experience-based techniques
There are four specification-based or black-box technique: • Equivalence partitioning. • Boundary value analysis. • Decision tables. • State transition testing. • The testers have no knowledge of how the system or component is structured inside the box. In black-box testing the tester is concentrating on what the software does, not how it does it. • The definition mentions both functional and non-functional testing. Functional testing is concerned with what the system does its features or	White-box testing e.g path testing Structure-based techniques serve two purposes: test coverage measurement and structural test case design • They are often used first to assess the amount of testing performed by tests derived from specification-based techniques, i.e. to assess coverage. • They are then used to design additional tests with the aim of increasing the test coverage. • Structure-based test design techniques are a good way of generating additional test cases that are	In **experience-based techniques,** people's knowledge, skills and background are of prime importance to the test conditions and test cases. The experience of both technical and business people is required, as they bring different perspectives to the test analysis and design process. Because of the previous experience with similar systems, they may have an idea as what could go wrong, which is very useful for testing. • Experience-based techniques go together with specification-based and structure-based techniques, and are also used when there is

functions. Non-functional testing is concerned with examining how well the system does. Non-functional testing like performance, usability, portability, maintainability, etc. • Specification-based techniques are appropriate at all levels of testing (component testing through to acceptance testing) where a specification exists. For example, when performing system or acceptance testing, the requirements specification or functional specification may form the	different from existing tests. • They can help ensure more breadth of testing, in the sense that test cases that achieve 100% **coverage** in any measure will be exercising all parts of the software from the point of view of the items being covered. Source: http://tryqa.com/what-is-structure-based-technique-in-software-testing/	no specification, or if the specification is inadequate or out of date. • This may be the only type of technique used for low-risk systems, but this approach may be particularly useful under extreme time pressure – in fact this is one of the factors leading to exploratory testing. Source: http://tryqa.com/what-is-experience-based-testing-technique/

basis of the tests. Source: http://tryqa.com/what-is-black-box-specification-based-also-known-as-behavioral-testing-techniques/		