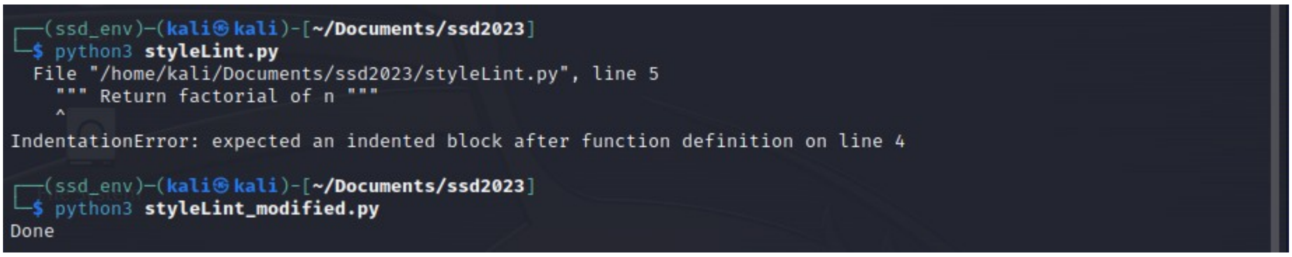
Linters

These questions are provided in the Codio workspace – Testing with Python – where the activities should be completed.

Question 1

Run styleLint.py in Codio.

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?



```
┌──(ssd_env)─(kali㉿kali)-[~/Documents/ssd2023]
└─$ python3 styleLint.py
  File "/home/kali/Documents/ssd2023/styleLint.py", line 5
    """ Return factorial of n """
    ^
IndentationError: expected an indented block after function definition on line 4

┌──(ssd_env)─(kali㉿kali)-[~/Documents/ssd2023]
└─$ python3 styleLint_modified.py
Done
```

- Various points of indentation needed to be fixed
- Code ran fine once fixed

Question 2

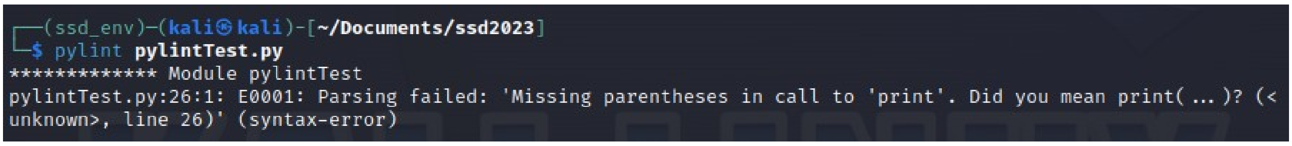
```
pip install pylint
```

Run

```
pylint
```

on pylintTest.py

Review each of the code errors returned. Can you correct each of the errors identified by pylint?



```
┌──(ssd_env)─(kali㉿kali)-[~/Documents/ssd2023]
└─$ pylint pylintTest.py
************* Module pylintTest
pylintTest.py:26:1: E0001: Parsing failed: 'Missing parentheses in call to 'print'. Did you mean print(...)? (<
unknown>, line 26)' (syntax-error)
```

- ‘print’ call needs parentheses
- Code runs fine with the fix

Before correcting the code errors, save the pylintTest.py file with a new name (it will be needed again in the next question).

Question 3

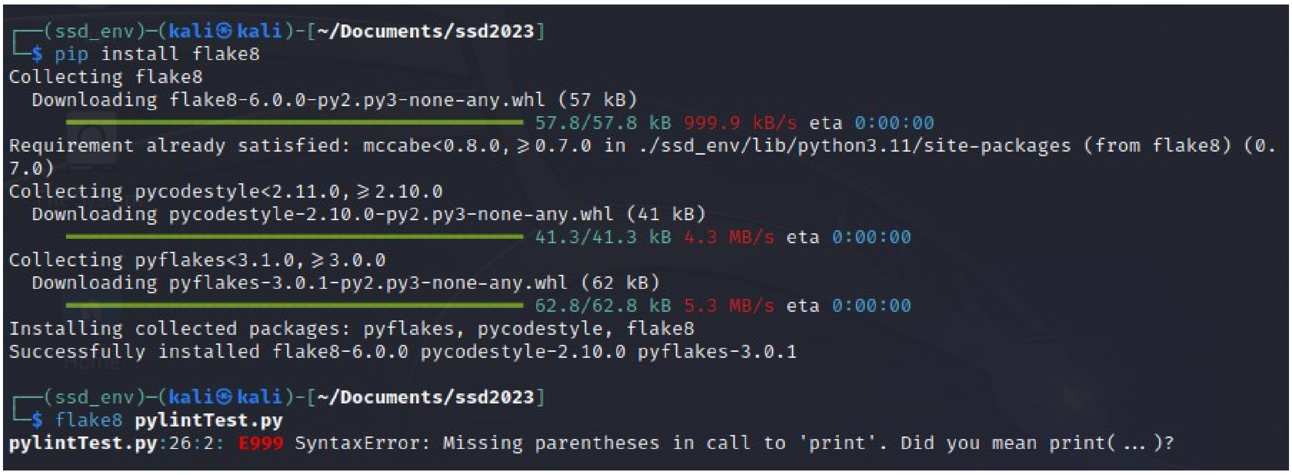
```
pip install flake8
```

Run

```
flake8
```

on pylintTest.py

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



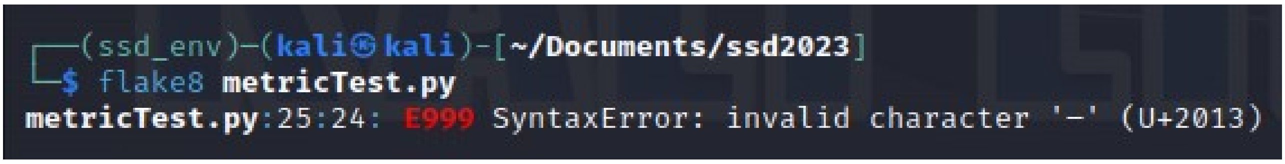
```
┌──(ssd_env)─(kali㉿kali)-[~/Documents/ssd2023]
└─$ pip install flake8
Collecting flake8
  Downloading flake8-6.0.0-py2.py3-none-any.whl (57 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 57.8/57.8 kB 999.9 kB/s eta 0:00:00
Requirement already satisfied: mccabe<0.8.0,≥0.7.0 in ./ssd_env/lib/python3.11/site-packages (from flake8) (0.
7.0)
Collecting pycodestyle<2.11.0,≥2.10.0
  Downloading pycodestyle-2.10.0-py2.py3-none-any.whl (41 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 41.3/41.3 kB 4.3 MB/s eta 0:00:00
Collecting pyflakes<3.1.0,≥3.0.0
  Downloading pyflakes-3.0.1-py2.py3-none-any.whl (62 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 62.8/62.8 kB 5.3 MB/s eta 0:00:00
Installing collected packages: pyflakes, pycodestyle, flake8
Successfully installed flake8-6.0.0 pycodestyle-2.10.0 pyflakes-3.0.1

┌──(ssd_env)─(kali㉿kali)-[~/Documents/ssd2023]
└─$ flake8 pylintTest.py
pylintTest.py:26:2: E999 SyntaxError: Missing parentheses in call to 'print'. Did you mean print(...)?
```

The error messages (pylint and flake8) have the following differences:

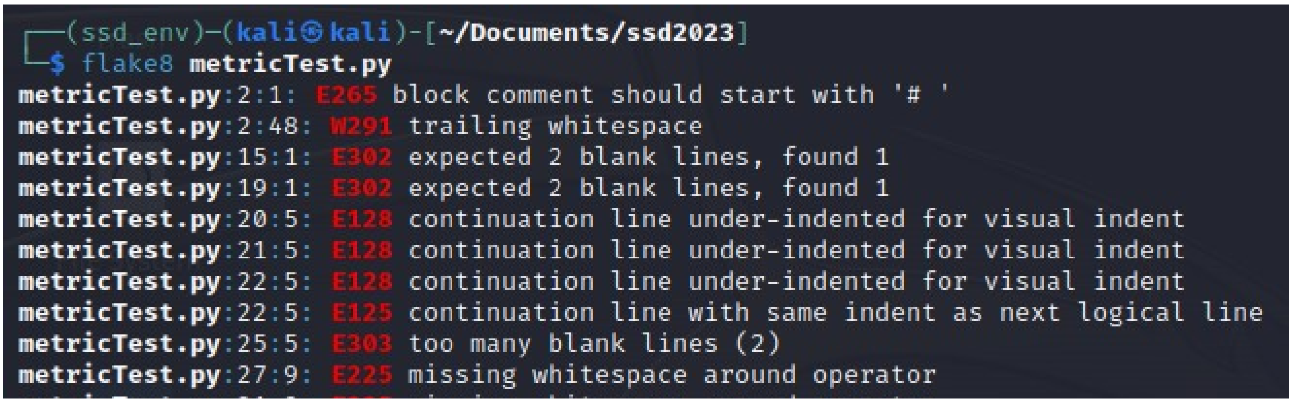
- a different character error place - 26.1 vs 26.2
- a different error code – E0001 vs E999
- a different error label – parsing failed vs syntax error
    - flake8 could be considered more accurate here

Run flake8 on metricTest.py. Can you correct each of the errors returned by flake8? What amendments have you made to the code?

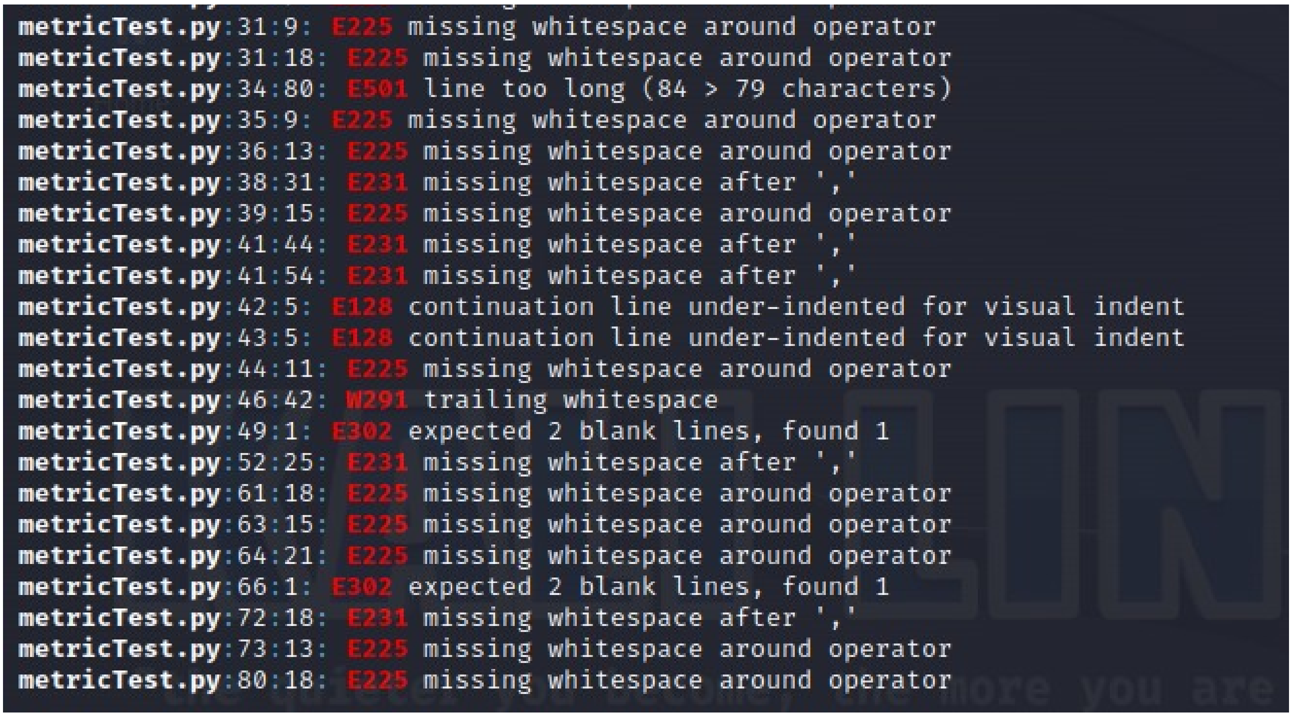
The following amendments were made to the code:



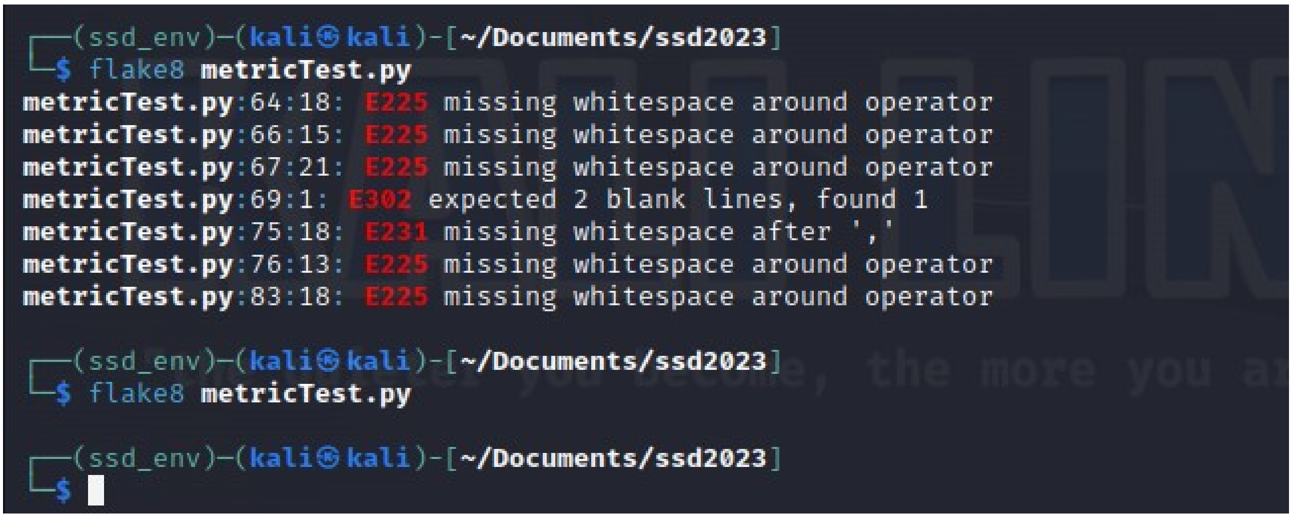
```
┌──(ssd_env)─(kali㉿kali)-[~/Documents/ssd2023]
└─$ flake8 metricTest.py
metricTest.py:25:24: E999 SyntaxError: invalid character '–' (U+2013)
```



```
┌──(ssd_env)─(kali㉿kali)-[~/Documents/ssd2023]
└─$ flake8 metricTest.py
metricTest.py:2:1: E265 block comment should start with '# '
metricTest.py:2:48: W291 trailing whitespace
metricTest.py:15:1: E302 expected 2 blank lines, found 1
metricTest.py:19:1: E302 expected 2 blank lines, found 1
metricTest.py:20:5: E128 continuation line under-indented for visual indent
metricTest.py:21:5: E128 continuation line under-indented for visual indent
metricTest.py:22:5: E128 continuation line under-indented for visual indent
metricTest.py:22:5: E125 continuation line with same indent as next logical line
metricTest.py:25:5: E303 too many blank lines (2)
metricTest.py:27:9: E225 missing whitespace around operator
```



```
metricTest.py:31:9: E225 missing whitespace around operator
metricTest.py:31:18: E225 missing whitespace around operator
metricTest.py:34:80: E501 line too long (84 > 79 characters)
metricTest.py:35:9: E225 missing whitespace around operator
metricTest.py:36:13: E225 missing whitespace around operator
metricTest.py:38:31: E231 missing whitespace after ','
metricTest.py:39:15: E225 missing whitespace around operator
metricTest.py:41:44: E231 missing whitespace after ','
metricTest.py:41:54: E231 missing whitespace after ','
metricTest.py:42:5: E128 continuation line under-indented for visual indent
metricTest.py:43:5: E128 continuation line under-indented for visual indent
metricTest.py:44:11: E225 missing whitespace around operator
metricTest.py:46:42: W291 trailing whitespace
metricTest.py:49:1: E302 expected 2 blank lines, found 1
metricTest.py:52:25: E231 missing whitespace after ','
metricTest.py:61:18: E225 missing whitespace around operator
metricTest.py:63:15: E225 missing whitespace around operator
metricTest.py:64:21: E225 missing whitespace around operator
metricTest.py:66:1: E302 expected 2 blank lines, found 1
metricTest.py:72:18: E231 missing whitespace after ','
metricTest.py:73:13: E225 missing whitespace around operator
metricTest.py:80:18: E225 missing whitespace around operator
```



```
┌──(ssd_env)─(kali㉿kali)-[~/Documents/ssd2023]
└─$ flake8 metricTest.py
metricTest.py:64:18: E225 missing whitespace around operator
metricTest.py:66:15: E225 missing whitespace around operator
metricTest.py:67:21: E225 missing whitespace around operator
metricTest.py:69:1: E302 expected 2 blank lines, found 1
metricTest.py:75:18: E231 missing whitespace after ','
metricTest.py:76:13: E225 missing whitespace around operator
metricTest.py:83:18: E225 missing whitespace around operator

┌──(ssd_env)─(kali㉿kali)-[~/Documents/ssd2023]
└─$ flake8 metricTest.py

┌──(ssd_env)─(kali㉿kali)-[~/Documents/ssd2023]
└─$ 
```

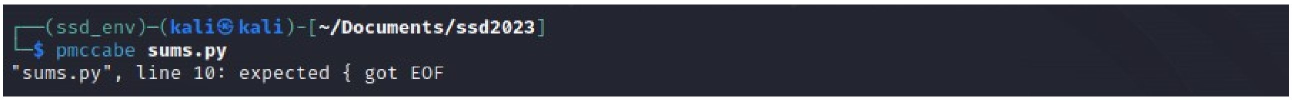
Question 4

```
pip install mccabe
```

Run

```
mccabe
```

on sums.py. What is the result?

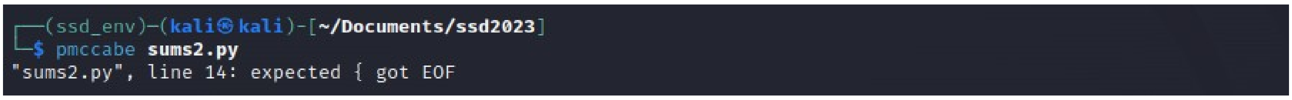


```
┌──(ssd_env)─(kali㉿kali)-[~/Documents/ssd2023]
└─$ pmccabe sums.py
"sums.py", line 10: expected { got EOF
```

Run

```
mccabe
```

on sums2.py. What is the result?



```
┌──(ssd_env)─(kali㉿kali)-[~/Documents/ssd2023]
└─$ pmccabe sums2.py
"sums2.py", line 14: expected { got EOF
```

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

Question 5 (e-portfolio entry): Exploring the Cyclomatic Complexity's Relevance Today

The Cyclomatic Complexity is commonly considered in modules on testing the validity of code design today. However, in your opinion, should it be? Does it remain relevant today? Specific to the focus of this module, is it relevant in our quest to develop secure software? Justify all opinions which support your argument and share your responses with your team.