**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?

**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?

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?

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

**Question 4**

**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?

**Question 1: Answers**

* Executing stylelint on the below source code:

Graphical user interface, text, application

Description automatically generated

* Produced:

Text

Description automatically generated

* The output points out the indentation error of the docstring within the function
* The code was amended to correct the error by indenting the docstring correctly
* The below is the next error after the correction:

Text

Description automatically generated

**Question 2: Answers**

* Pylint was run on the below code:

Text

Description automatically generated

* Below is the first error:

A screenshot of a computer

Description automatically generated with medium confidence

* The below errors were corrected to debug the entire program using pylint:

Text

Description automatically generated

**Question 3: Answers**

* Running the same code as above through the Linter of flake8 produces the below output:

Text

Description automatically generated

* The output is focussed on stylistic errors
* The below code was tested too:

Text

Description automatically generated with medium confidence

A picture containing graphical user interface

Description automatically generated

* The below output was produced:

codio@target-cobalt:~/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

metricTest.py:66:1: E112 expected an indented block

metricTest.py:66:12: E225 missing whitespace around operator

metricTest.py:69:1: E112 expected an indented block

metricTest.py:72:1: E112 expected an indented block

metricTest.py:74:17: E231 missing whitespace after ','

metricTest.py:75:1: E112 expected an indented block

metricTest.py:75:8: E225 missing whitespace around operator

metricTest.py:76:1: E112 expected an indented block

metricTest.py:77:8: E231 missing whitespace after ':'

metricTest.py:78:2: E201 whitespace after '['

metricTest.py:78:5: E202 whitespace before ']'

metricTest.py:82:1: E112 expected an indented block

metricTest.py:83:13: E225 missing whitespace around operator

metricTest.py:84:1: E112 expected an indented block

metricTest.py:86:1: E112 expected an indented block

metricTest.py:86:19: W292 no newline at end of file

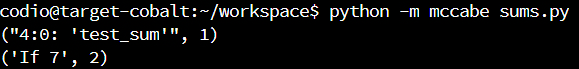
**Question 4 - Answers**

* Running the below code through the McCabe linter:

Graphical user interface, application

Description automatically generated

* Creates the below output:



* Running the below code through the mcabe linter:

Text

Description automatically generated

* Creates the below output:

Text

Description automatically generated

* The cyclomatic complexity of a portion of code is determined by the number/depth of conditional control statements