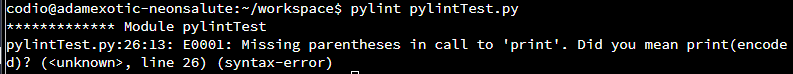
**Questions:**

Run *flake8* on pylintTest.py

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

Error returned by flake8:  

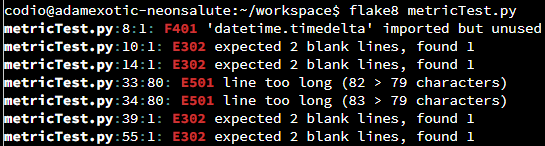

Error returned by PyLint:

The above returned errors are very similar, however the Pylint gives more detailed information and even proposes a correction.

Run *flake8* on metricTest.py.

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

The flak8 linter addressed the errors pretty fine, hence I didn’t have major issues. However, there were multiple errors with indentations and other errors, for example:



1. **What amendments have you made to the code?**

I had to remove unused module, shorten the size of a line, adding row between functions & classes, correct indentation.

The corrected code is:

"""

Module metricTest.py

Metric example - Module used as a testbed for static checkers.

This is a mix of different functions and classes doing different things.

"""

import random

from datetime import timedelta

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 30 < 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:

    """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