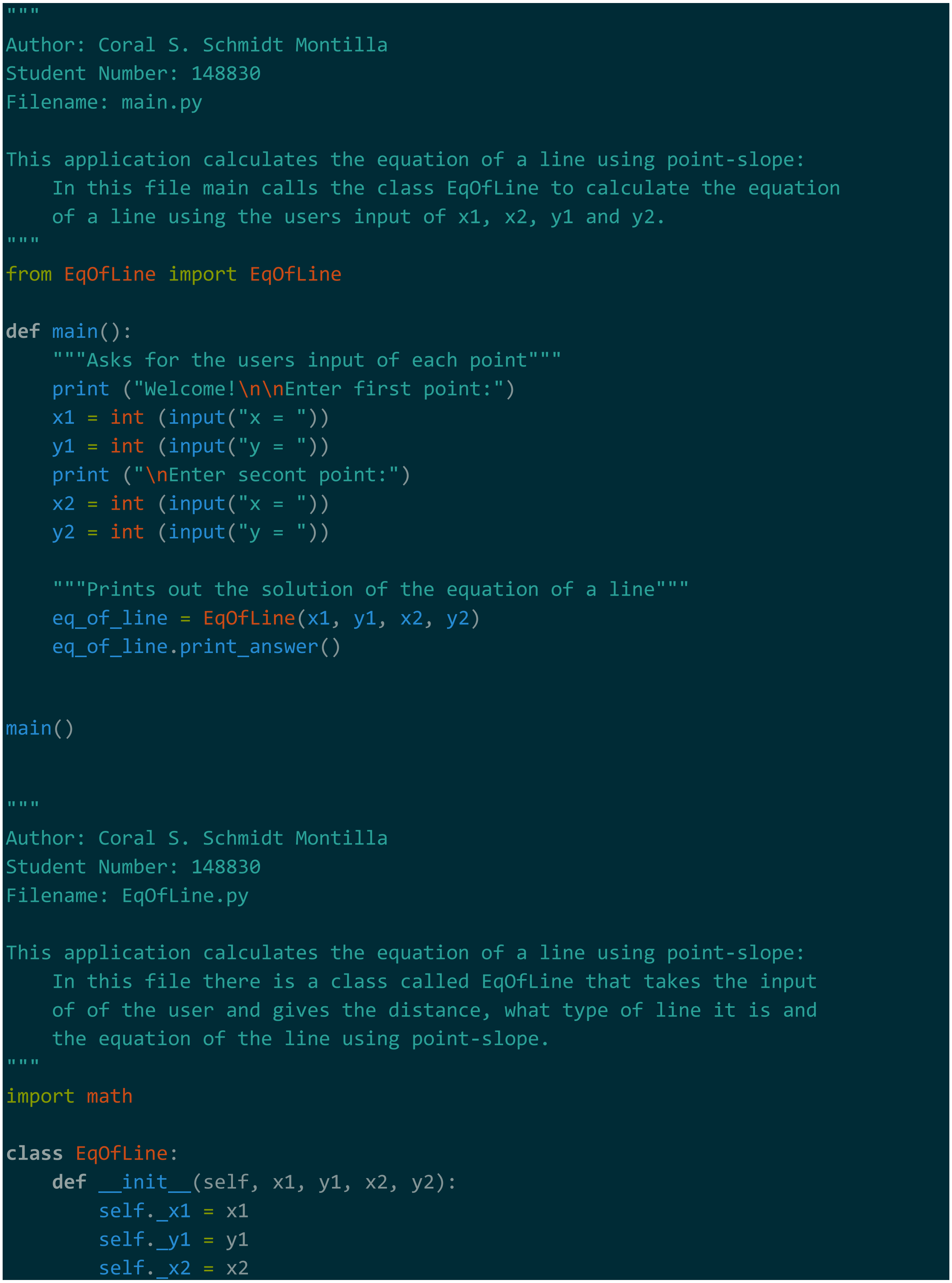
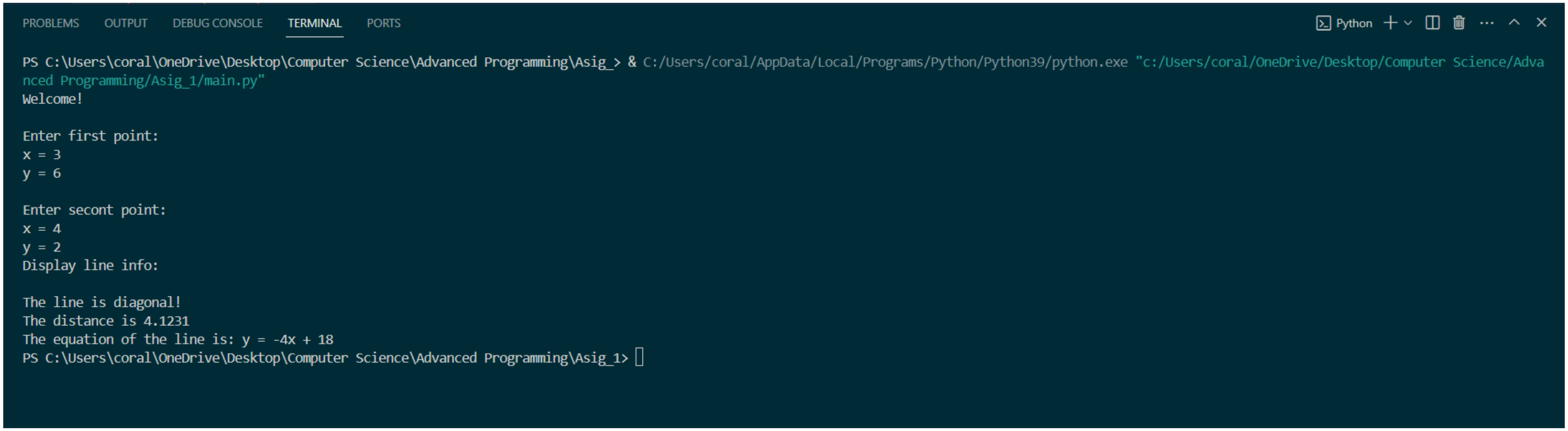


```
"""
Author: Coral S. Schmidt Montilla
Student Number: 148830
Filename: main.py

This application calculates the equation of a line using point-slope:
    In this file main calls the class EqOfLine to calculate the equation
    of a line using the users input of x1, x2, y1 and y2.
"""
from EqOfLine import EqOfLine

def main():
    """Asks for the users input of each point"""
    print ("Welcome!\n\nEnter first point:")
    x1 = int (input("x = "))
    y1 = int (input("y = "))
    print ("\nEnter secont point:")
    x2 = int (input("x = "))
    y2 = int (input("y = "))

    """Prints out the solution of the equation of a line"""
    eq_of_line = EqOfLine(x1, y1, x2, y2)
    eq_of_line.print_answer()


main()


"""
Author: Coral S. Schmidt Montilla
Student Number: 148830
Filename: EqOfLine.py

This application calculates the equation of a line using point-slope:
    In this file there is a class called EqOfLine that takes the input
    of of the user and gives the distance, what type of line it is and
    the equation of the line using point-slope.
"""
import math

class EqOfLine:
    def __init__(self, x1, y1, x2, y2):
        self._x1 = x1
        self._y1 = y1
        self._x2 = x2
        self._y2 = y2

    def get_x1(self):
        return self._x1

    def get_y1(self):
        return self._y1

    def get_x2(self):
        return self._x2

    def get_y2(self):
        return self._y2

    def set_x1(self, x1):
        self._x1 = x1

    def set_y1(self, y1):
        self._y1 = y1

    def set_x2(self, x2):
        self._x2 = x2

    def set_y2(self, y2):
        self._y2 = y2

    """Calculates the distance of the line"""
    def distance(self):
        x1 = self.get_x1()
        y1 = self.get_y1()
        x2 = self.get_x2()
        y2 = self.get_y2()

        return math.sqrt(pow(x2 - x1, 2) + pow(y2 - y1, 2))

    """Identifies what type of line it is"""
    def line(self):
        if self.get_y1() == self.get_y2():
            return "horizontal"
        elif self.get_x1() == self.get_x2():
            return "vertical"
        else:
            return "diagonal"

    """Gets the equation of the line"""
    def equation_of_line(self):
        x1 = self.get_x1()
        y1 = self.get_y1()
        x2 = self.get_x2()
        y2 = self.get_y2()

        m = (y2 - y1) / (x2 - x1)
        b = y1 - m * x1

        if m.is_integer() and b.is_integer():
            return f"y = {int(m)}x + {int(b)}"
        else:
            return f"y = {m:.2f}x + {b:.2f}"

    def print_answer(self):
        print("Display line info:\n\nThe line is %s!\nThe distance is %3.4f\nThe
equation of the line is: %s" % (self.line(), self.distance(),
self.equation_of_line()))
```

Output:



```
PS C:\Users\coral\OneDrive\Desktop\Computer Science\Advanced Programming\Asig_> & C:/Users/coral/AppData/Local/Programs/Python/Python39/python.exe "c:/Users/coral/OneDrive/Desktop/Computer Science/Adva
nced Programming/Asig_1/main.py"
Welcome!

Enter first point:
x = 3
y = 6

Enter secont point:
x = 4
y = 2
Display line info:

The line is diagonal!
The distance is 4.1231
The equation of the line is: y = -4x + 18
PS C:\Users\coral\OneDrive\Desktop\Computer Science\Advanced Programming\Asig_1> 
```