**Guess The Number Game**

Kai Qian S365690

**Introduction:**

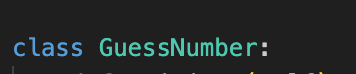
Our target is to develop a guessing game. The requirements is players need to guess a randomly generated four digit number. Then, the program will provide some clues about the number. Once the player guesses the number correctly, the program will display the number of attempts. I will use Test Driven Development (TDD) to develop our progect. The Tools We Will Use is: Visual Studio Code, flake8, pylint.

**Process**:

Use TDD to create our program. These tests will fully cover the basic requirements of the game, assuming that the game is constructed as a class (GuessNumber) that includes methods for generating numbers, verifying guesses, and exiting the game. Testing should initially fail until each test's functionality is implemented.

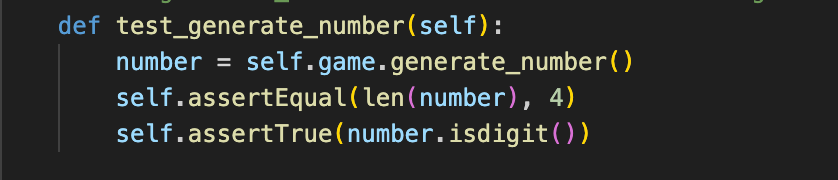
First, will create a test class named TestGuessTheNumber, and a main function class named GuessNumber.



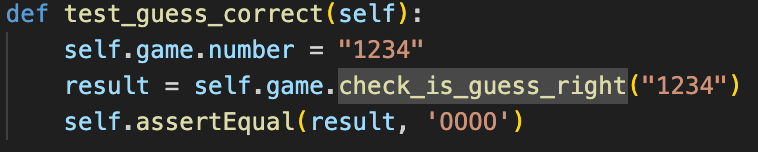


Then,Write our tests:

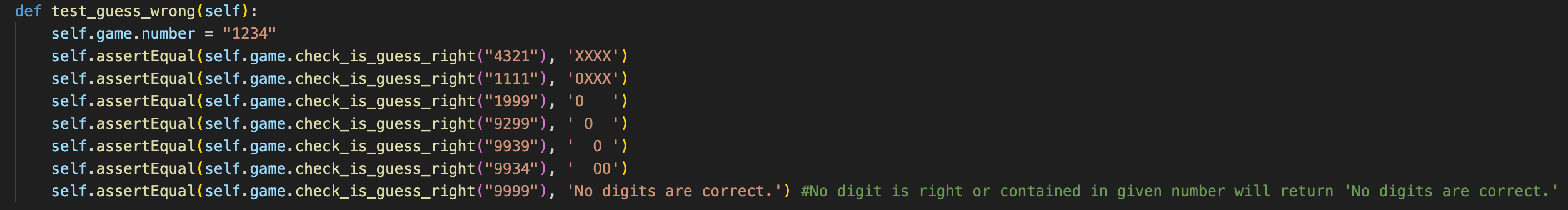
Write our first test function test\_generate\_number in our test class TestGuessTheNumber, in order to test if generate\_number function is successfully generate a 4 digit random number.



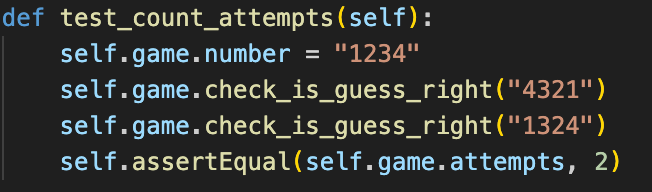
Write our second test function test\_guess\_correct to test the condition that the input number is right.



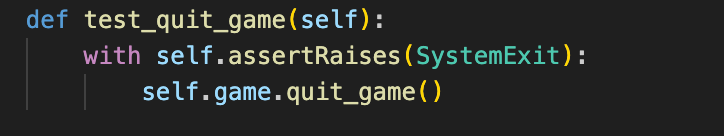
Write our third test function test\_guess\_wrong to test the condition that the input number is wrong.



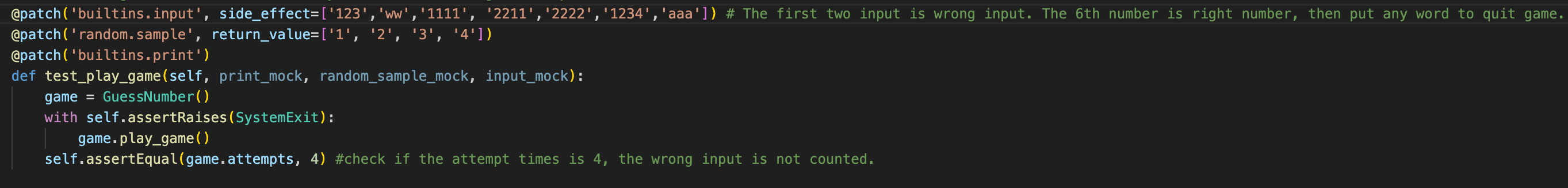
Write our next test function test\_count\_attempts to test if the attempt times is right.



Then, Write our next test function test\_quit\_game to test if the game will quit successfully.

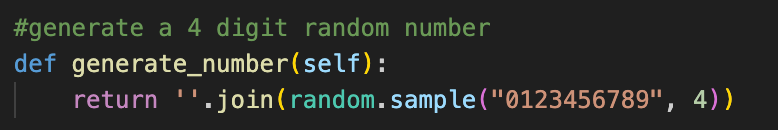


Write our last test function test\_play\_game with the given number and input numbers to test if the game will run successfully and return the right attempts.

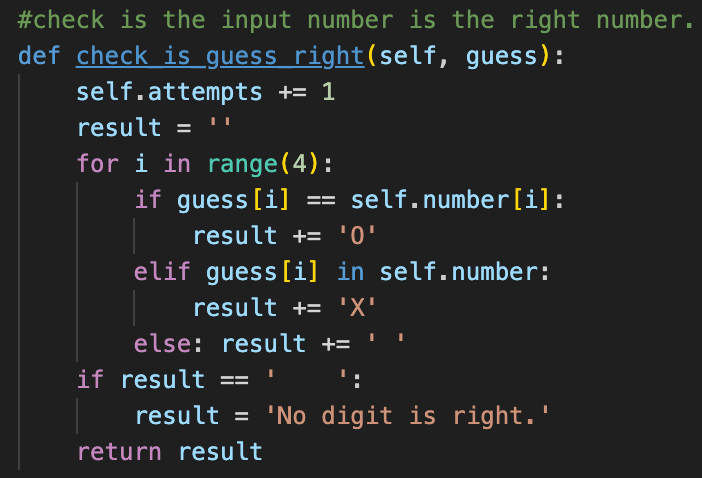


Now, we are starting to implement the required functions.

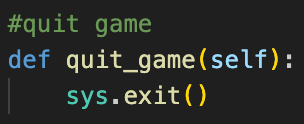
Implement the generate\_number function:



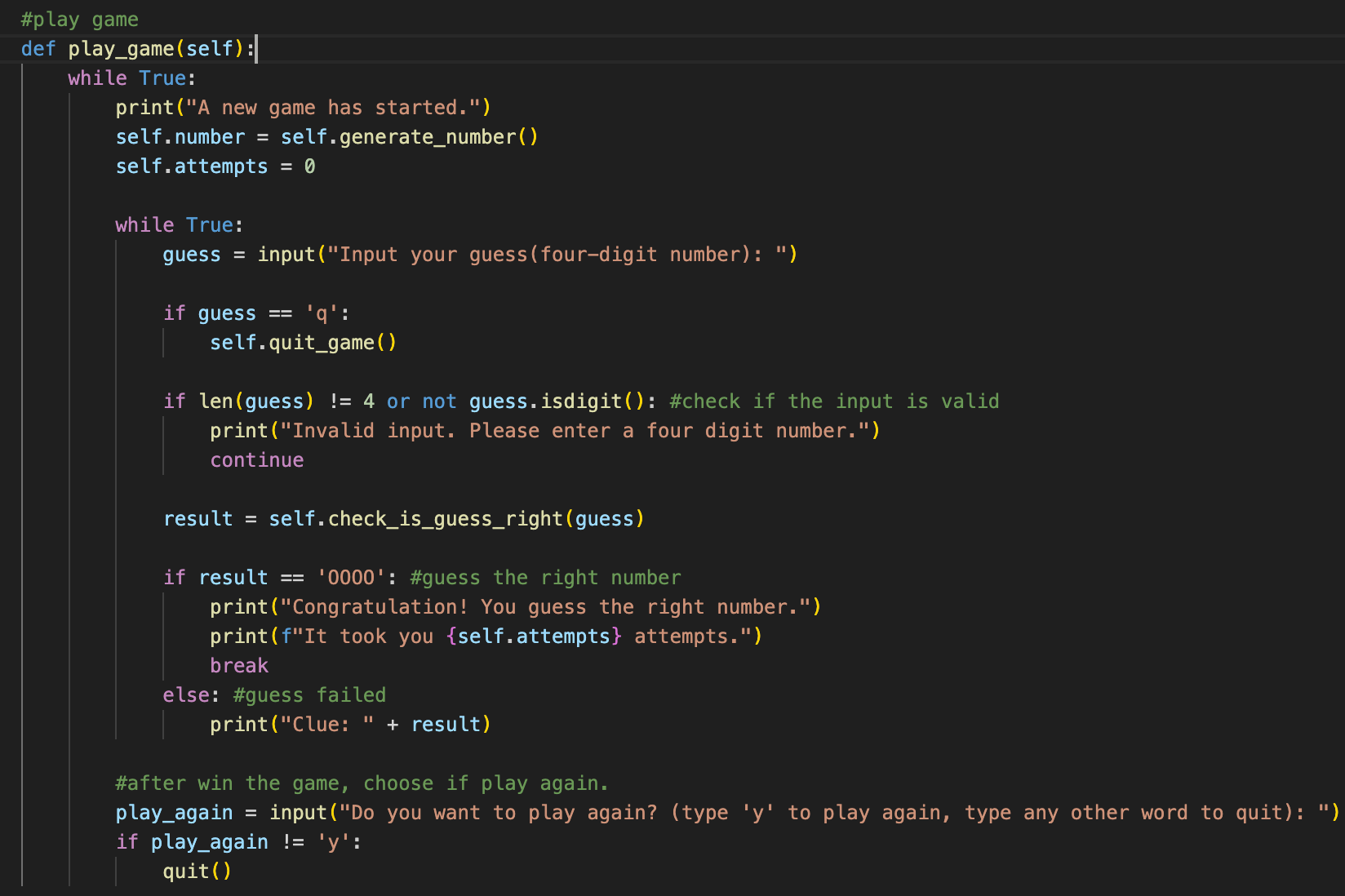
Implement the check\_is\_guess\_right function:



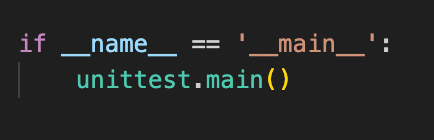
Implement the quit\_game function:



Implement the play\_game function:

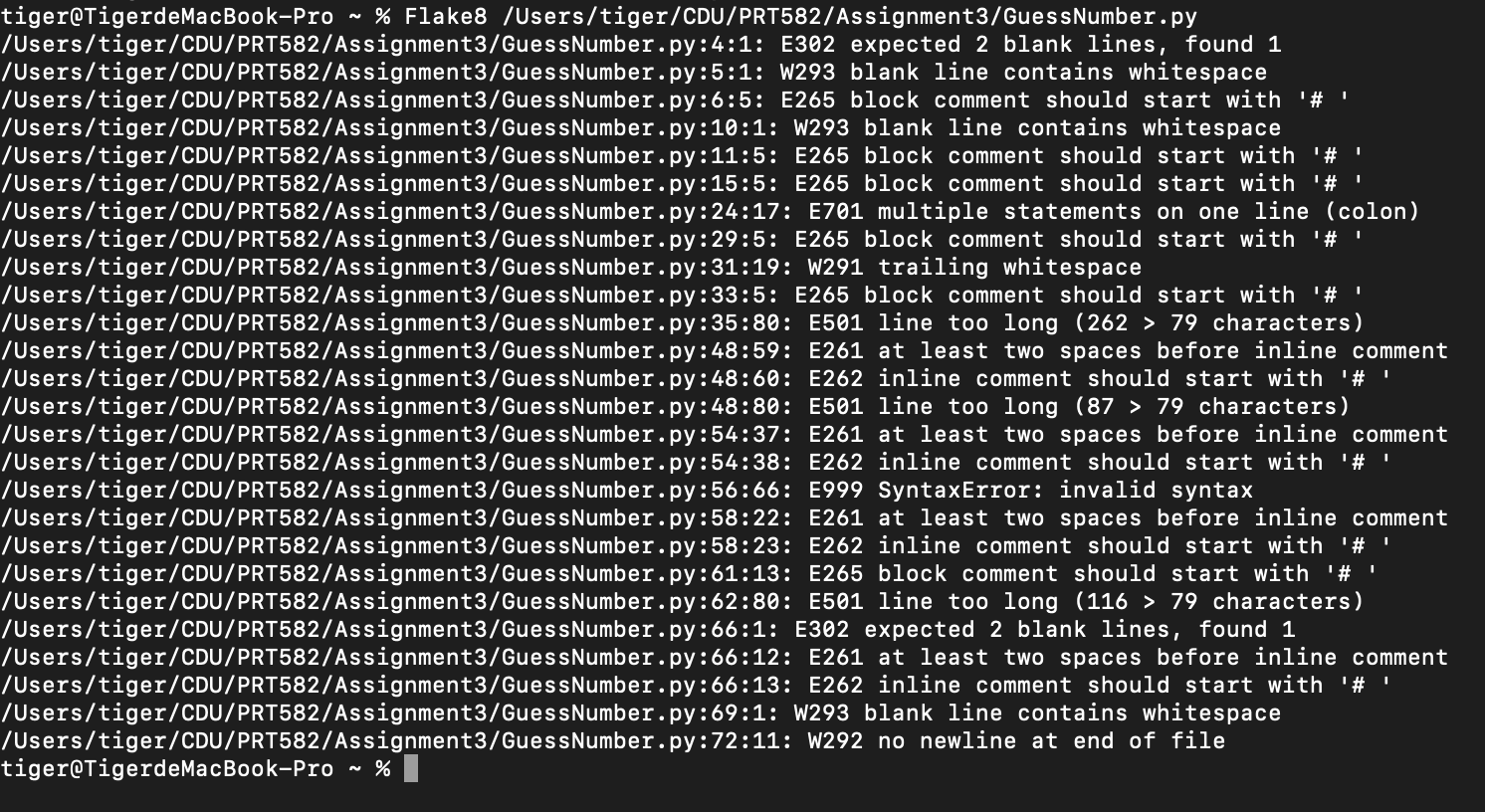


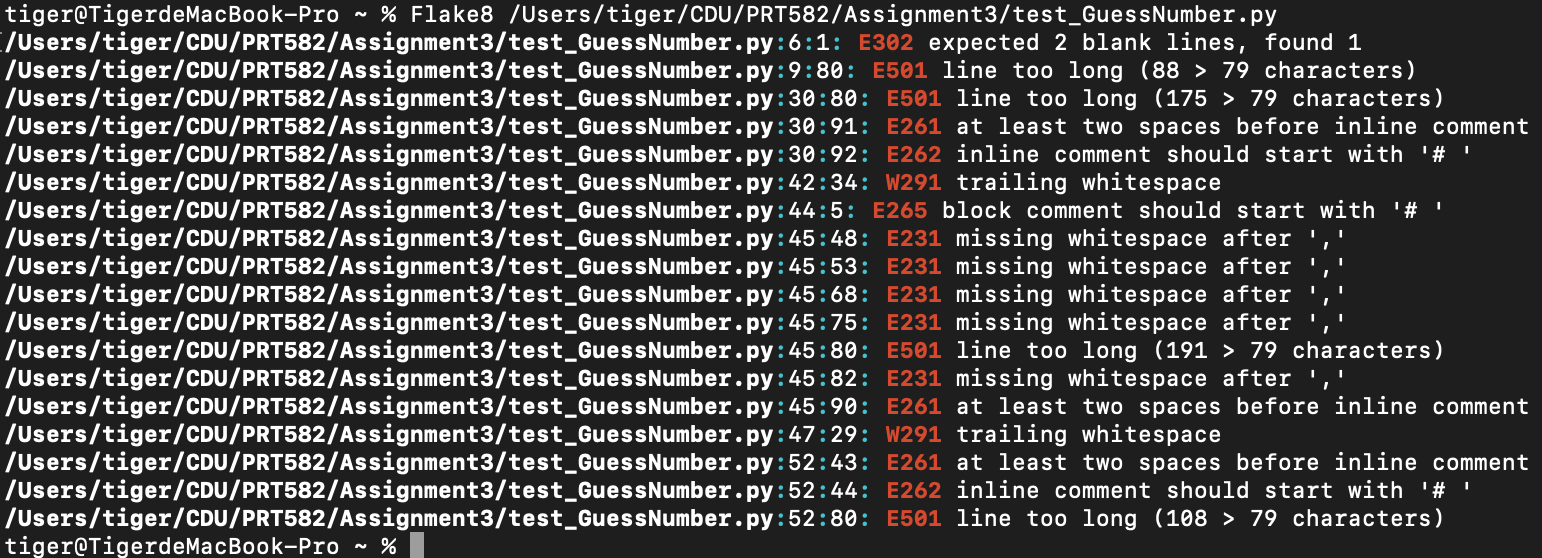
All functions are Implemented. Finally, run the unittest file to see if all tests have passed. If not, fix problems in GuessNumber class accordingly.



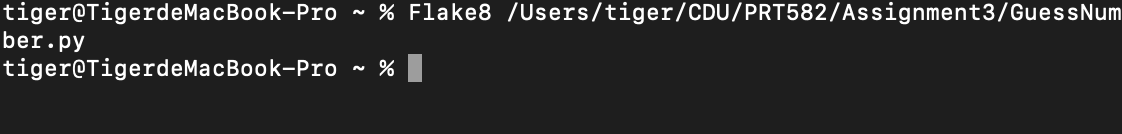
**Use Flake8 to check the code:**

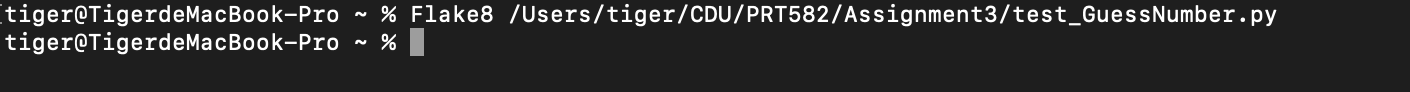
We find some flaws on two files, and fix it one by one.





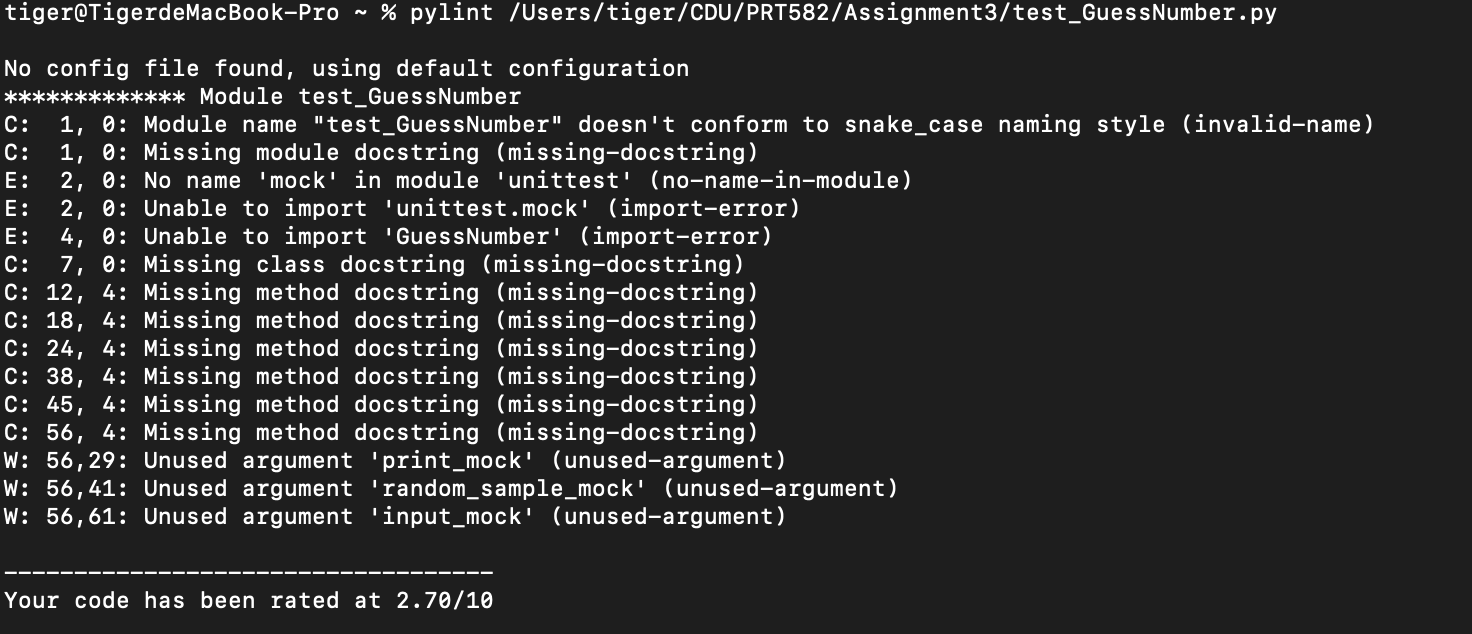
After fix these:

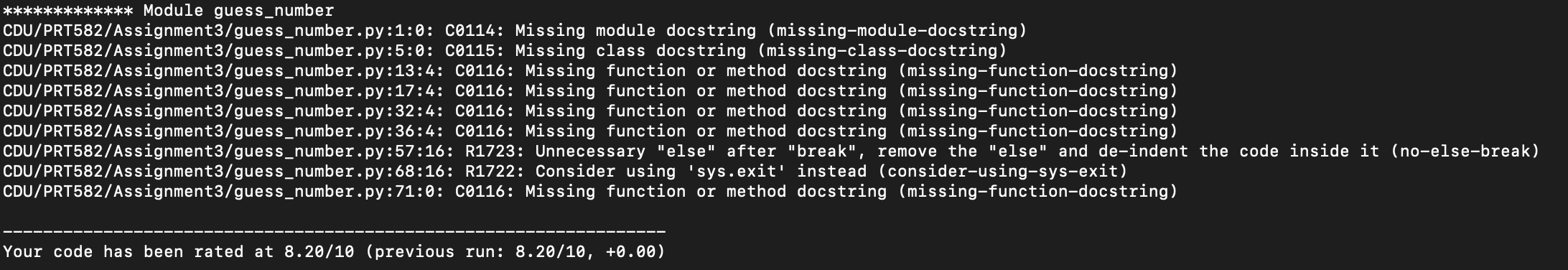




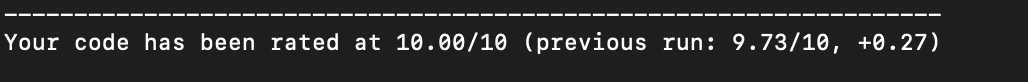
**Use pylint to check the code:**

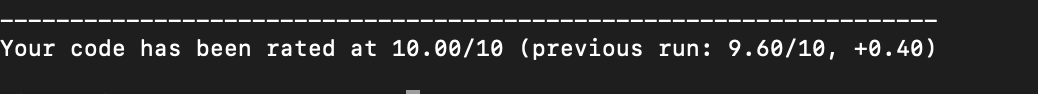
We find some flaws on two files:





After fix these:





Our coding work is done.

**Conclusion:**

TDD helps me to ensure that the code executes exactly as it was designed rather than unintentionally. In addition, if any errors are found in the future, test cases can show the errors, and then I can fix them. This time, I write all the tests first, then Implement these. It makes the coding a little bit hard. I could write a test and then implement it immediately. It will make the process much easier. Flake8 and Lint help me improve code quality, but it also takes time. Next time, I’ll try to put the auto-check and format plugin into the VS Code. It will save many time.

The GitHub link of the progect: