**Software Unit Testing Report**

**Introduction:**

The objective of this project is to develop a "Guess the Number" game using Test Driven Development (TDD) in Python. The game requires players to guess a randomly generated four-digit number. The program provides feedback to the player through 'circle' and 'x' clues, indicating correct digits in the right or wrong positions. The player can continue guessing until they guess correctly or decide to quit. After each game, the program displays the number of attempts made. For testing, we'll be using the built-in testing module `unittest` in Python. The `unittest` framework provides a structure for defining test cases, writing test methods, and running tests automatically. It helps ensure that our code functions as expected and that any changes we make do not introduce unexpected errors.

**Process:**

Write Failing Tests:

We will start by writing tests that define the expected behavior of the game's components. These tests will be defined in a separate test file, `test\_guess\_the\_number.py`.

Write Simplest Code:

We will then write the simplest code to make the failing tests pass. This will involve implementing the core game logic and functions.

Refactor if Needed:

After the tests pass, we may refactor the code to improve its structure, readability, and maintainability.

Iterate:

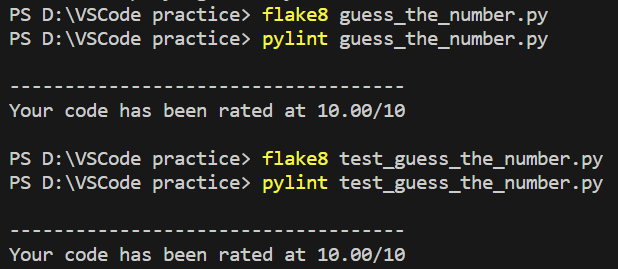
We will repeat the process of writing tests, implementing code, and refactoring until the entire game is developed.

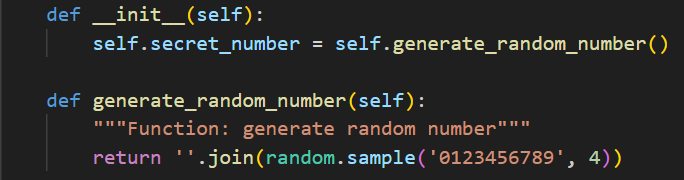
Implement the Game Loop:

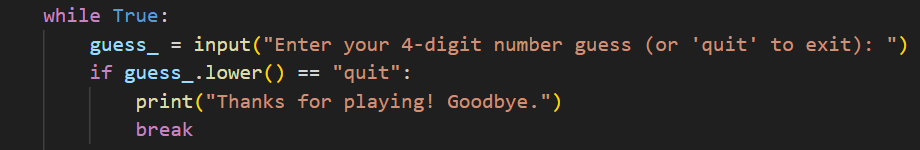
Once the core game logic is complete, we will implement the game loop, which will handle user input, checking guesses, providing clues, and managing game flow.

Run Tests:

Throughout development, we will run our automated tests to ensure that any changes we make do not break existing functionality.

Has used and addressed all the comments of flake8 and pylint：

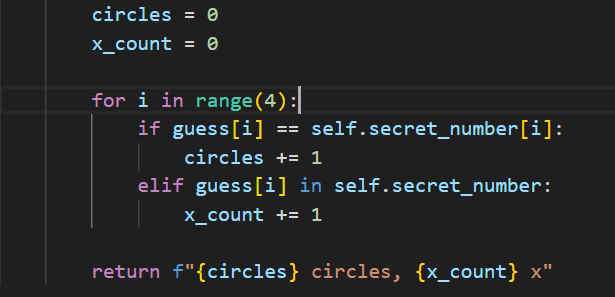
Randomly generate a four-digit number：

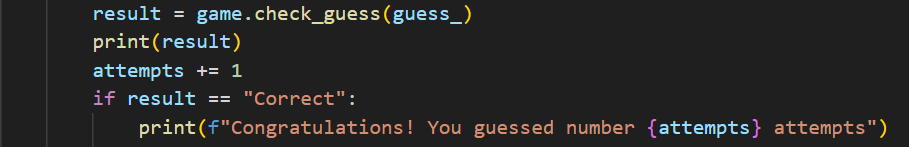
The program will keep asking the user to guess the number until the player guesses it correctly or has quitted：

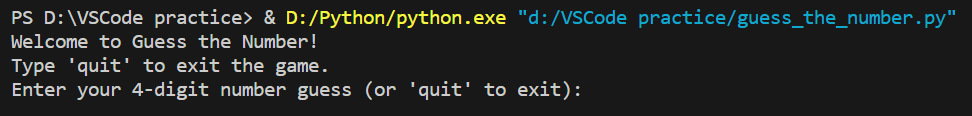
When the number is entered, the program will respond with hints using ‘circle’ and ‘x’ to show how accurate the guess was.

A ‘circle’ indicates that one digit is correct and is in the right spot

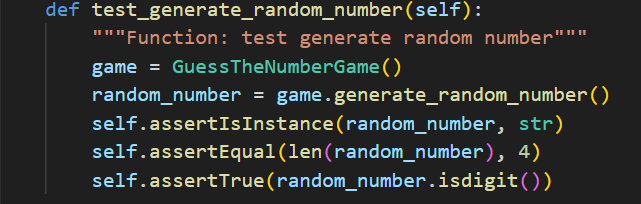
A ‘x’ indicates that one digit is correct but in the wrong spot



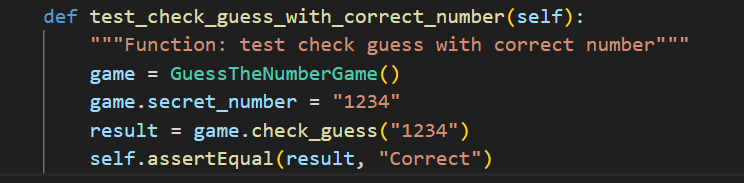
Once the game is finished, the number of attempts taken will be displayed:

Run guess\_the\_number.py:

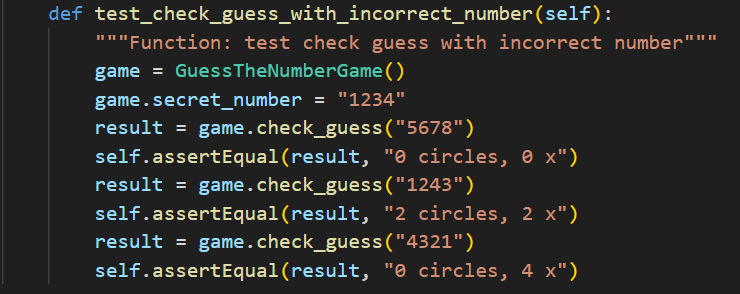
Test generate random number:



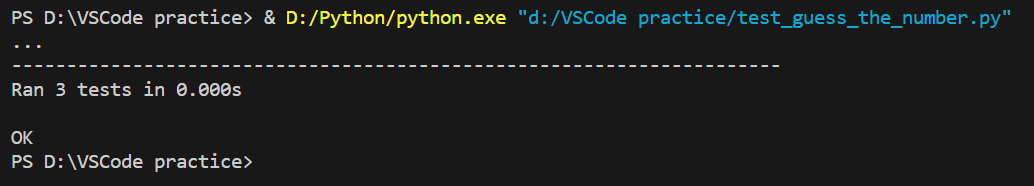
Test check guess with correct number:



Test check guess with incorrect number:



Run test\_guess\_the\_number.py:



**Conclusion:**

In conclusion, developing the "Guess the Number" game using Test Driven Development and the `unittest` framework provides a structured approach to building reliable and functional software. This process starts with writing test cases to define the desired behavior, followed by iterative code implementation and testing. By writing tests before the code, we ensure that the implementation is guided by a clear set of requirements. This methodology not only results in well-tested code but also provides a framework for maintaining and enhancing the software over time. Through this project, we learn the importance of test-driven development in building robust and maintainable applications.

GitHub link: https://github.com/MichaelChen0w/Software-Unit-Testing.git