Lesson 04: Practical Application of AI-Powered Code Generation Tools

Overview

In this guided practice, you will build a simple number guessing game using Python and GitHub Copilot in VS Code. By the end, you will have a working game where the player tries to guess a randomly generated number within a set number of attempts.

Instructions

1. Work individually or in small groups to discuss your approach
2. Use VS Code with the GitHub Copilot extension to complete each task given below
3. Spend approximately 20 minutes completing all tasks
4. Document your observations and strategies in the worksheet provided for review

Tasks

Scenario: You are a junior developer at a game development startup, and your team has been tasked with creating a simple text-based game using Python. The goal is to design an interactive number guessing game where players must guess a randomly generated number within a limited number of attempts. Since you are using GitHub Copilot, you want to leverage AI-assisted coding to speed up development and ensure best practices in your code.
Your task is to develop the game in VS Code, utilizing inline comments and docstrings to guide Copilot in generating structured and optimized code.

Steps to be followed:

1. Set up the project
2. Generate a random number
3. Implement user input handling
4. Write the game logic
5. Test the game

Tools required: VS Code with GitHub Copilot

Dataset to be used: None

Discussion questions (optional)

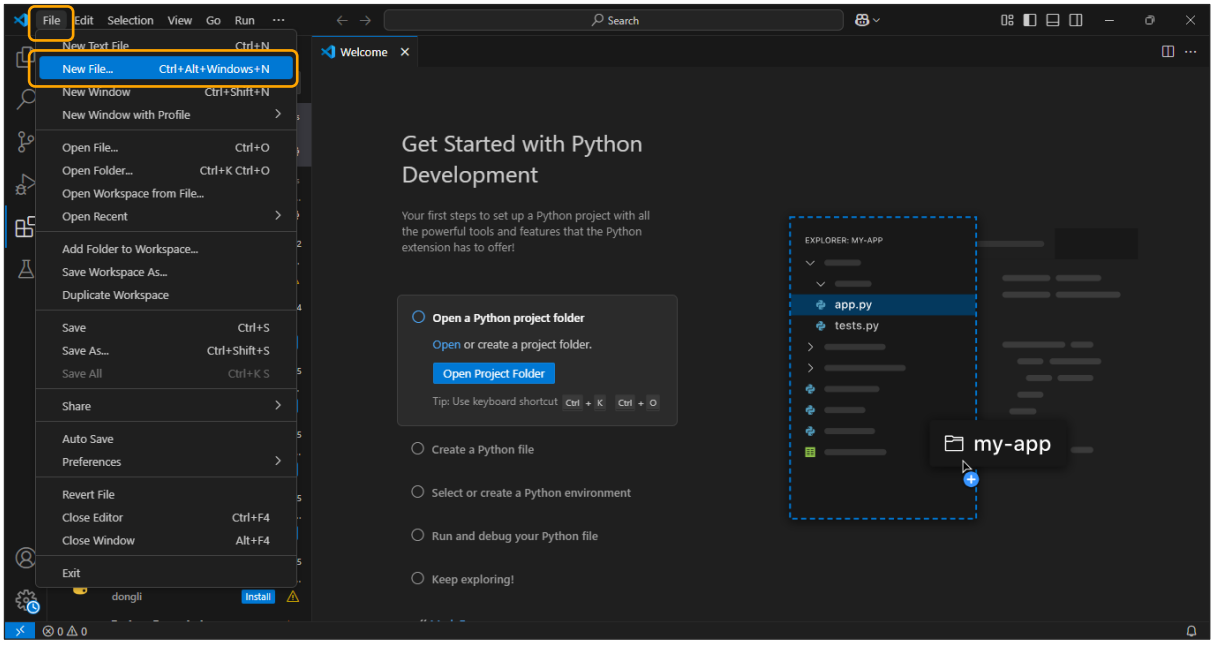
If time permits, discuss the following questions:

1. How did GitHub Copilot assist in generating the game logic, and what were its limitations?
2. Why is human oversight necessary when using AI-assisted coding tools like GitHub Copilot?

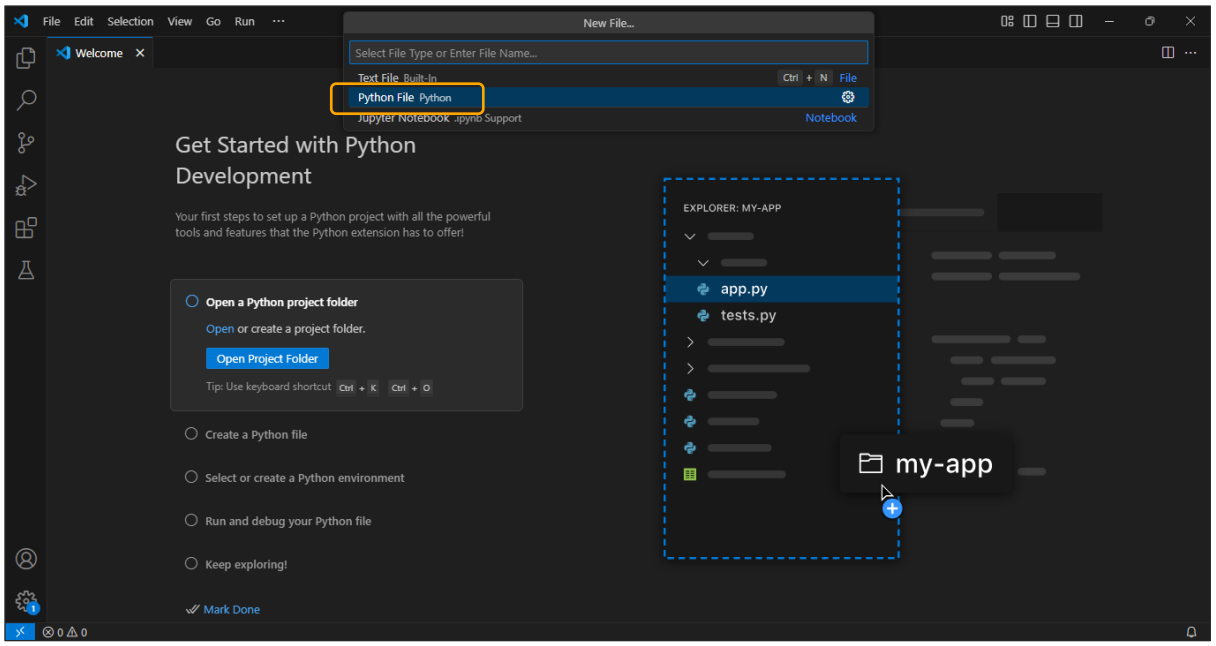
Answer Key

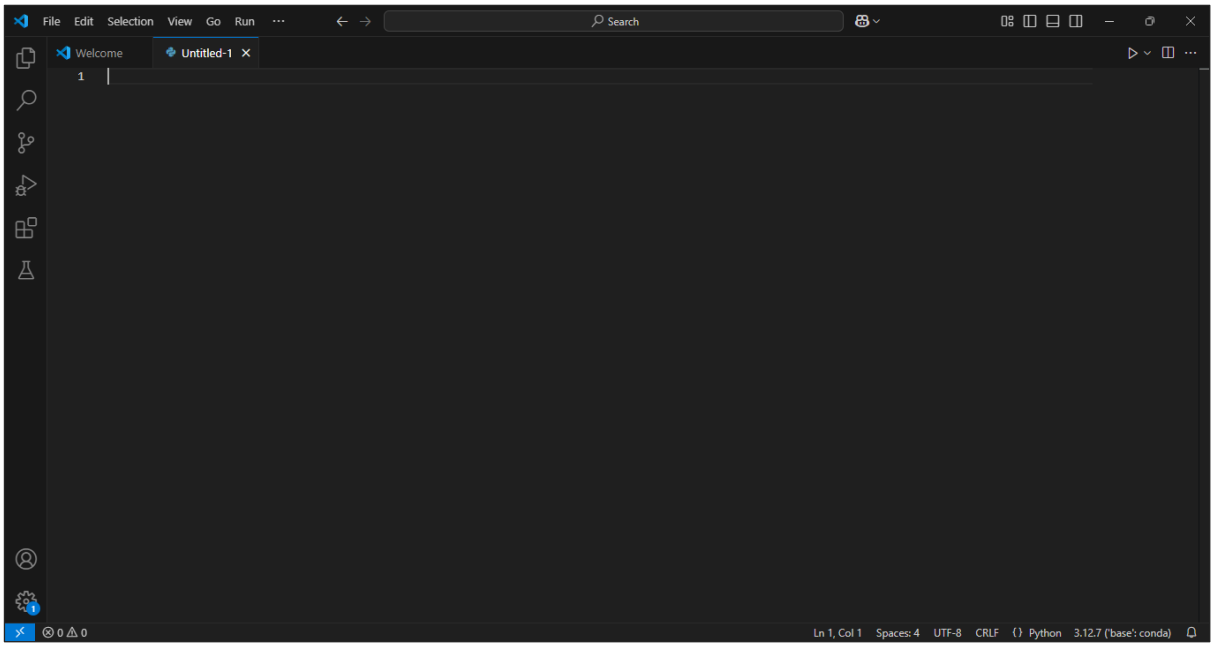
Step 1: Set up the project

1.1 Launch VS Code and click on **File** and then **New File**

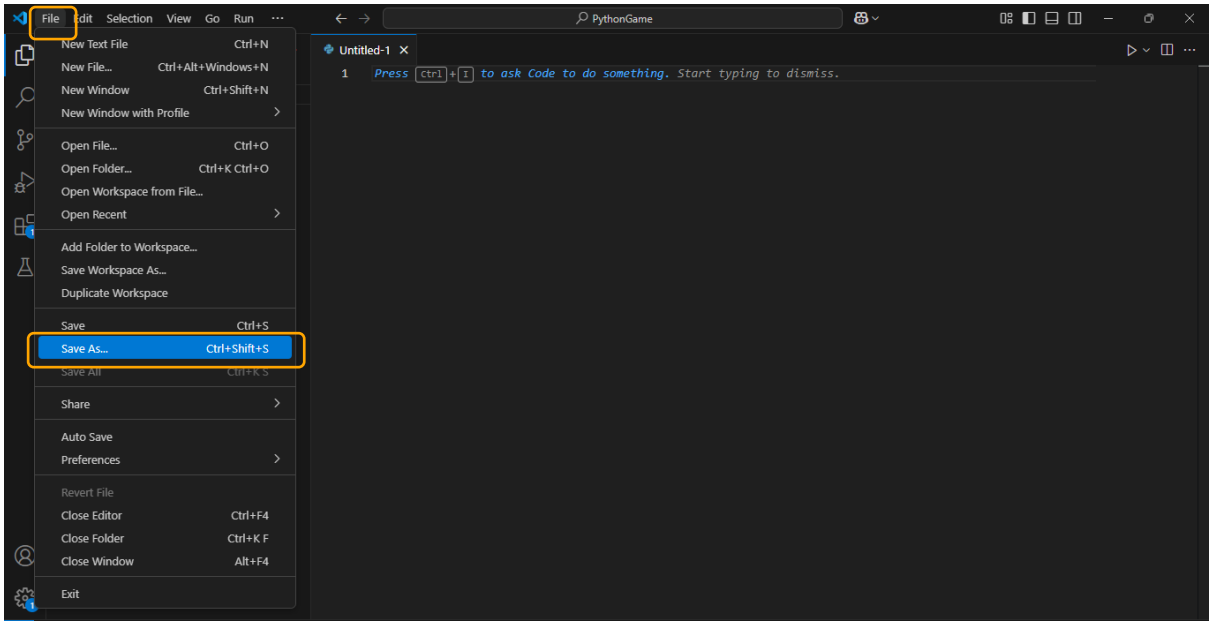


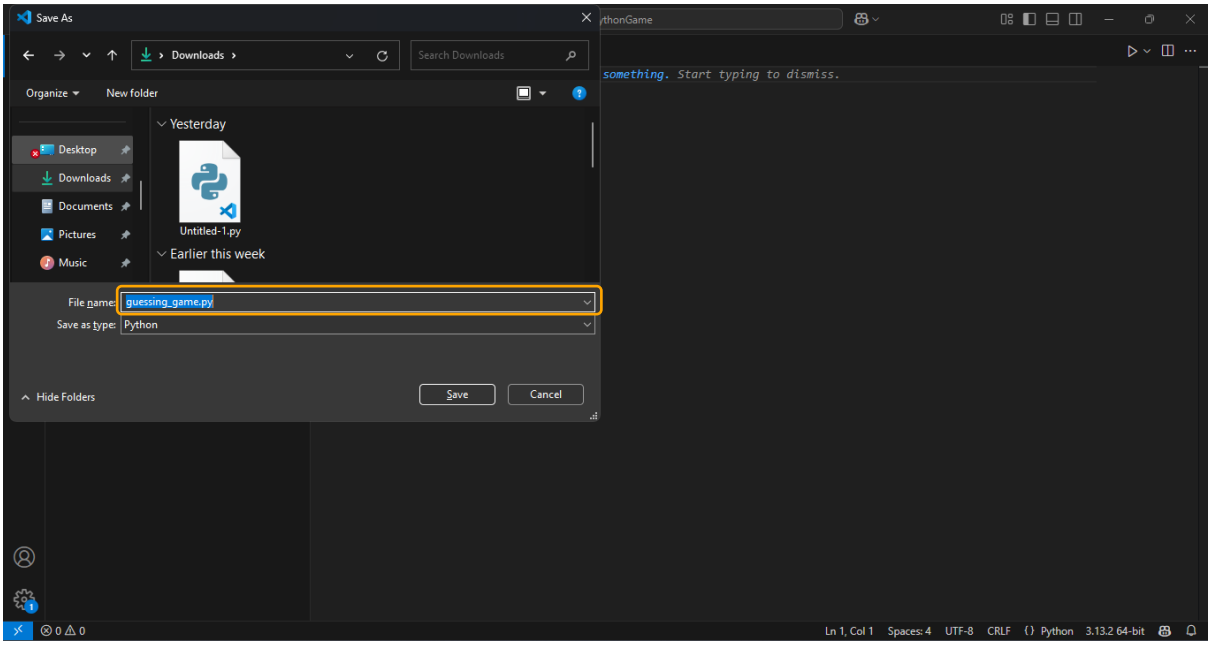
1.2 Select the **Python File** option from the name bar on top and a new Python file named Untitled-1 will open





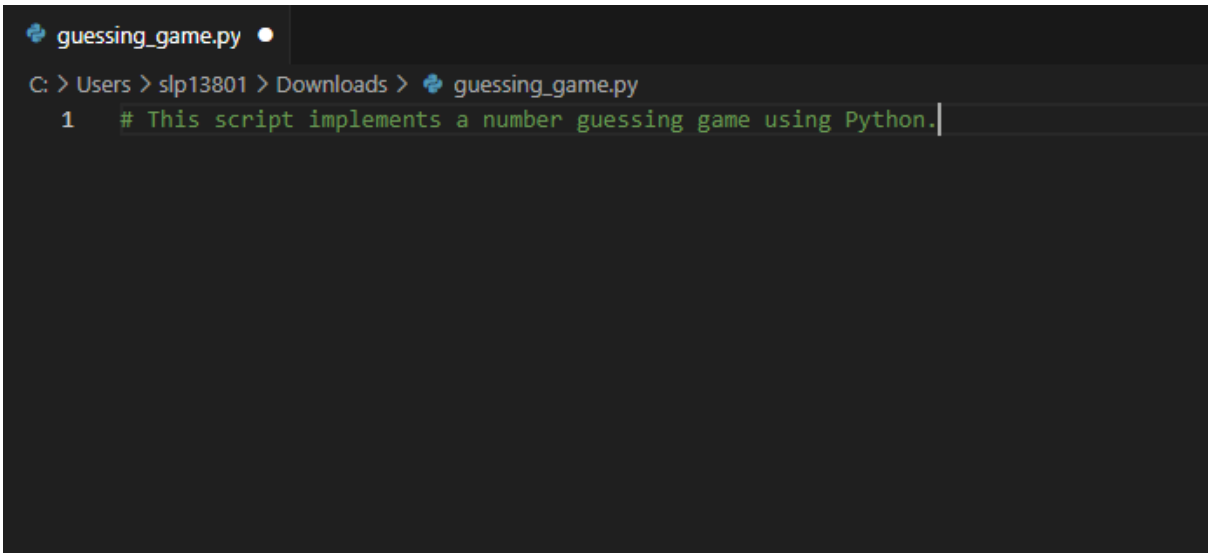
1.3 Click on **File**. Select **Save As...** and rename the file to *guessing_game.py* before saving it to your preferred location





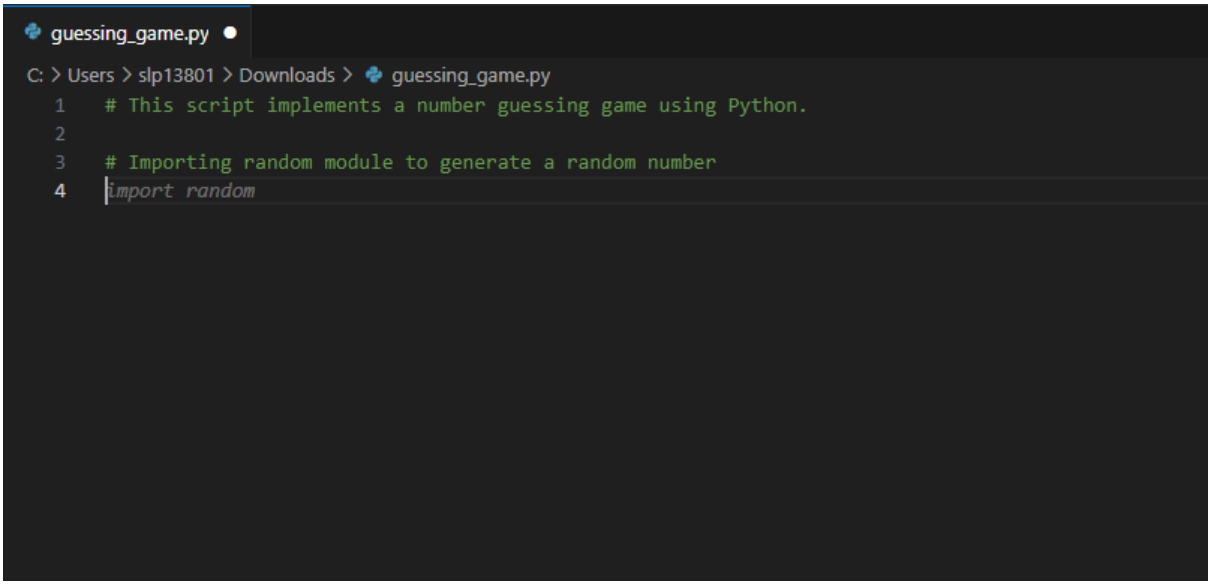
1.4 Type the following inline comment to guide Copilot:

 # This script implements a number-guessing game using Python.

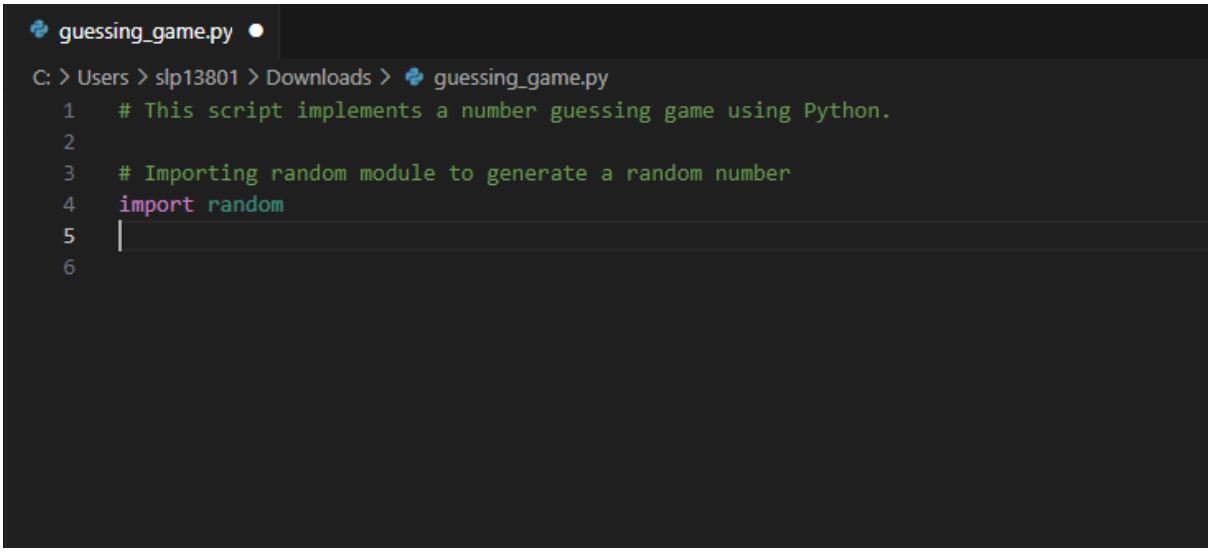


Step 2: Generate a random number

2.1 To import the random module, add an inline comment explaining why it's needed. Then, press enter to get Copilot's suggestions.

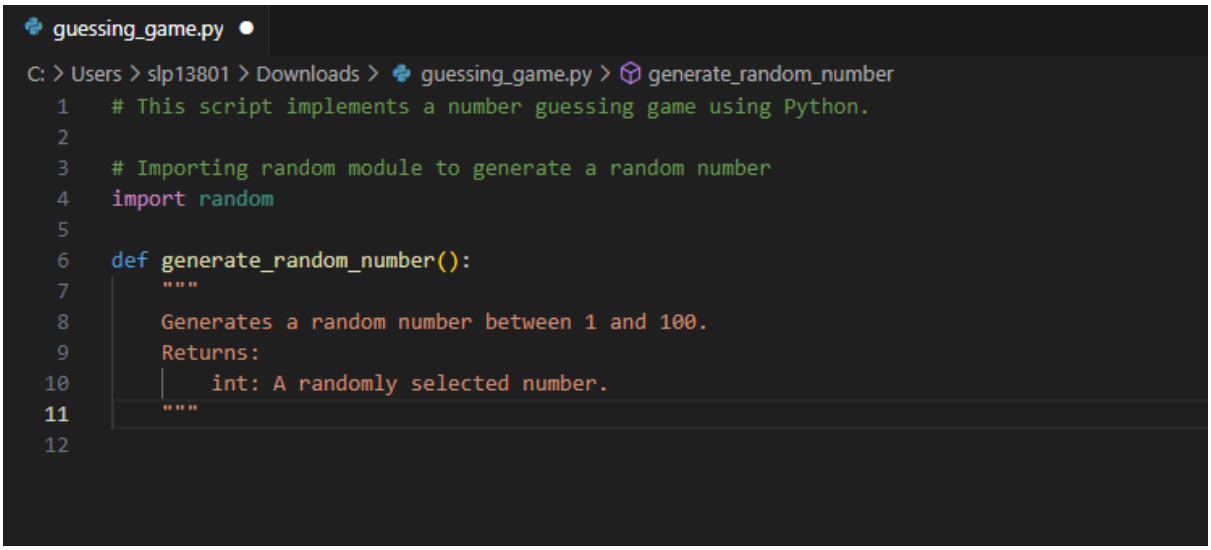


2.2 Observe Copilot's suggestions and press tab to accept them

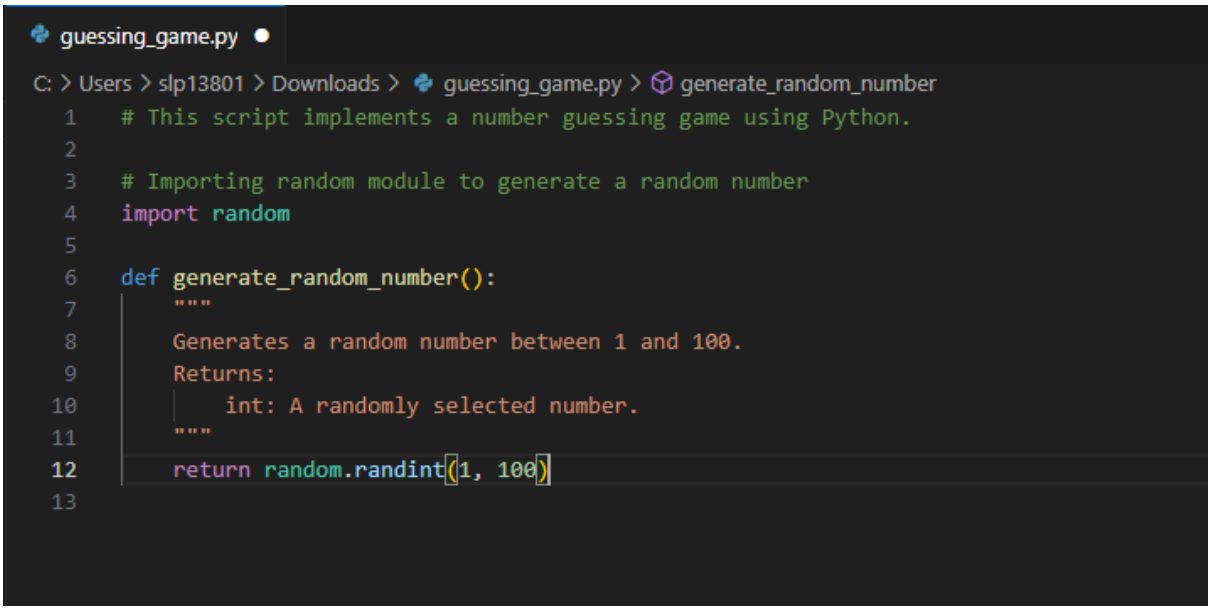


2.3 Write the following function with a docstring to guide Copilot in generating a random number:

```
def generate_random_number():
    """
    Generates a random number between 1 and 100.
    Returns:
        int: A randomly selected number.
    """
```



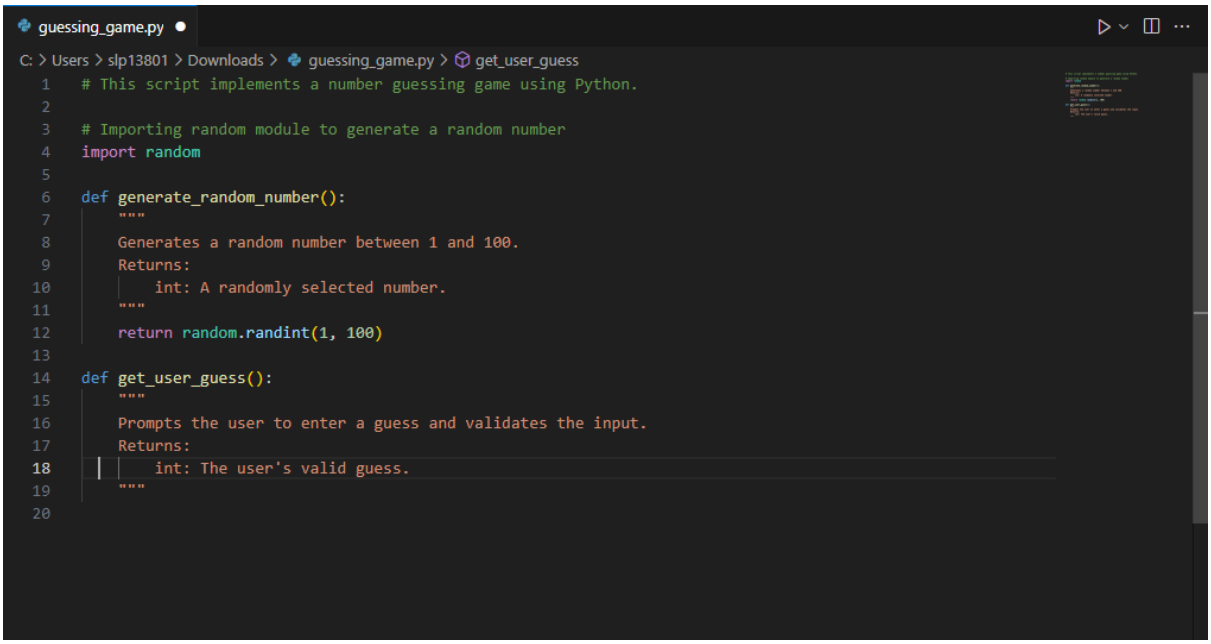
2.4 Press enter, observe Copilot's suggestions, and press tab to accept them



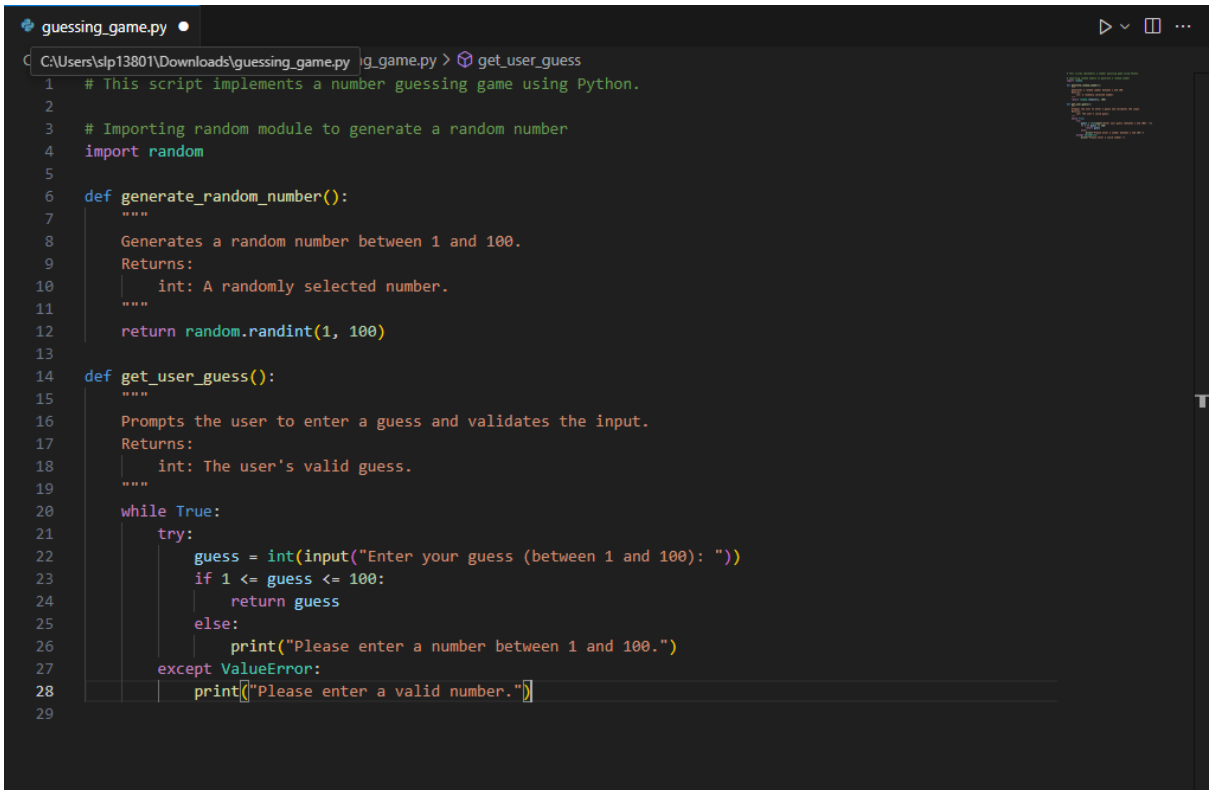
Step 3: Implement user input handling

3.1 Define the following function with a docstring to take user input and validate it:

```
def get_user_guess():
    """
    Prompts the user to enter a guess and validates the input.
    Returns:
        int: The user's valid guess.
    """
```



3.2 Press enter, observe Copilot’s suggestions, and press tab to accept them



Step 4: Write the game logic

4.1 Define the following main game function and explain its logic with a docstring:

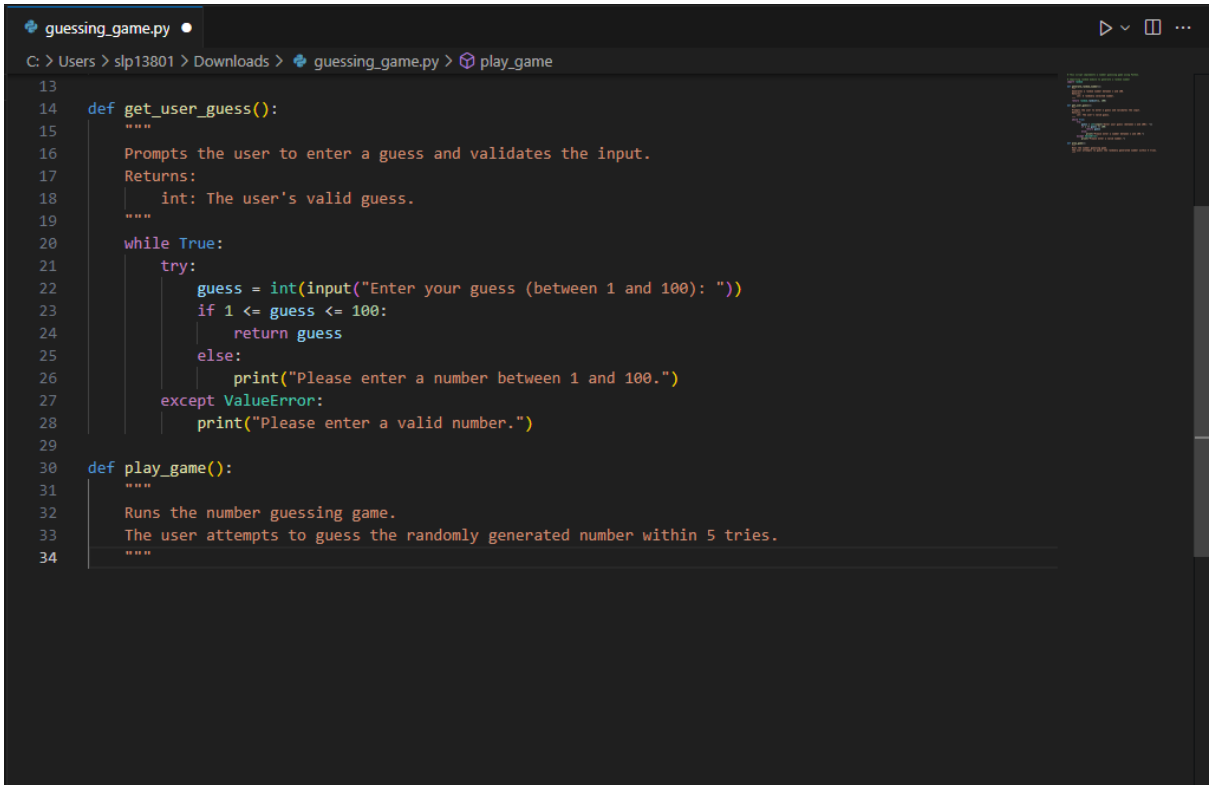
def play_game():

"""

Runs the number guessing game.

The user attempts to guess the randomly generated number within 5 tries.

"""



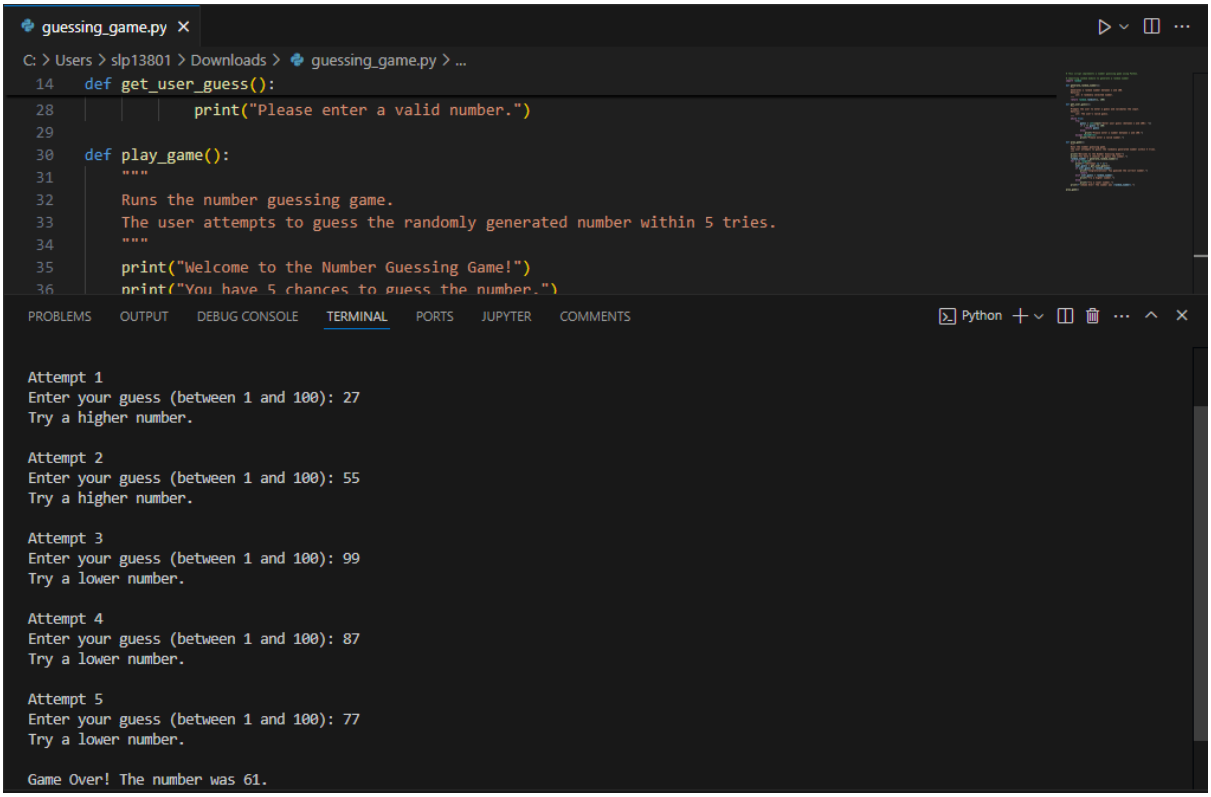
4.2 Press enter, observe Copilot's suggestions, and press tab to accept them

4.3 Create a call to the function to test the logic using the following syntax:
play_game()

Step 5: Test the game

5.1 Click the play button on the top-right corner to run the Python file

The following output is generated after running the file:



By following these steps, you have successfully built a number guessing game using Python and GitHub Copilot in VS Code. You learned how to use inline comments and docstrings to guide Copilot in generating structured and efficient code. Additionally, you explored how Copilot assists in writing game logic, handling user input, and managing game conditions. This practice reinforces the importance of AI-assisted coding while ensuring human oversight for accuracy and optimization. You can further enhance the game by adding difficulty levels, score tracking, or additional game mechanics.

Discussion questions (optional)

If time permits, discuss the following questions:

1. How did GitHub Copilot assist in generating the game logic, and what were its limitations?
   Answer: GitHub Copilot provided quick code suggestions based on inline comments and docstrings, helping to structure the game efficiently. However, it was limited in understanding specific game logic requirements, requiring adjustments and testing to refine the functionality.

2. Why is human oversight necessary when using AI-assisted coding tools like GitHub Copilot?
   Answer: AI-assisted tools generate code based on patterns and existing data but lack a true understanding of specific project goals. Human oversight is essential to verify logic, optimize performance, and ensure the final code meets functional requirements without unnecessary errors.