Table of Contents

1. Code Completion and Context: **2**

2. Chat Interface: **4**

3. Ghost Text and Code Generation: **5**

4. Inline Chat and Collaboration: **7**

5. Best Practices and Dos and Don'ts: **7**

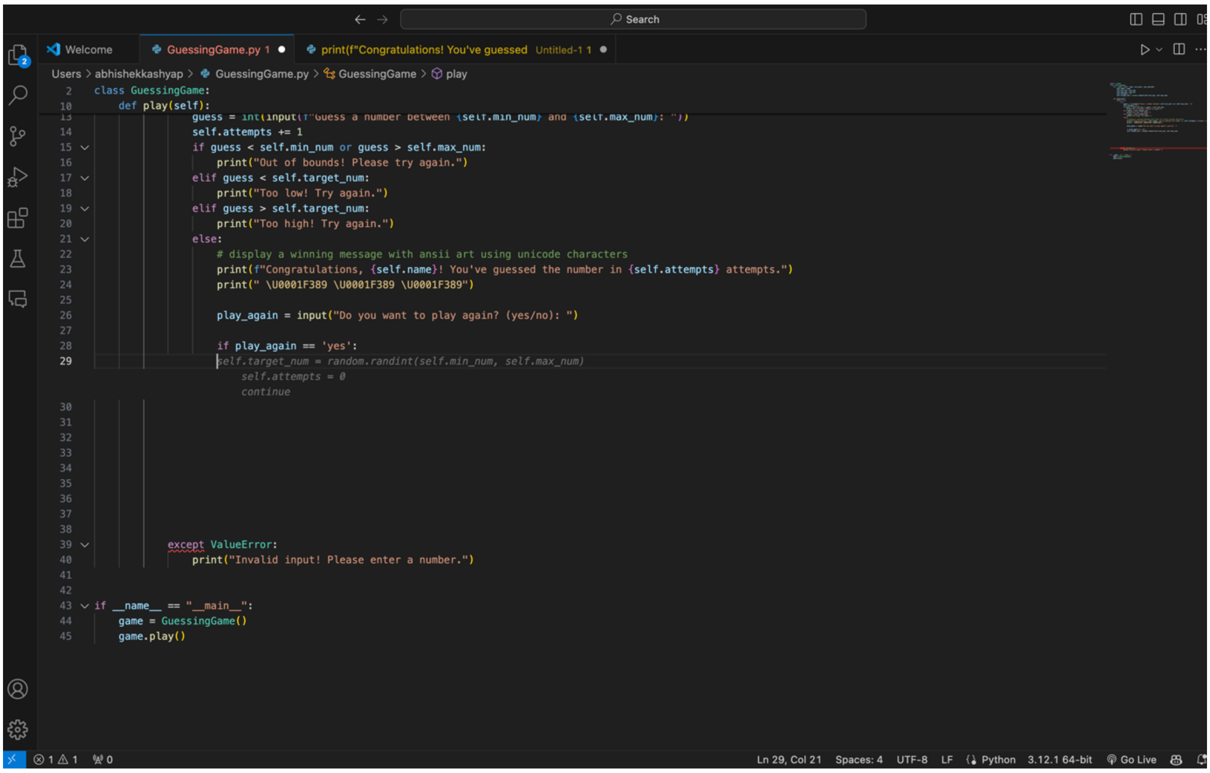
6. Security and Privacy Considerations: Unauthorized Code Access **9**

7. Game Screenshots **11**

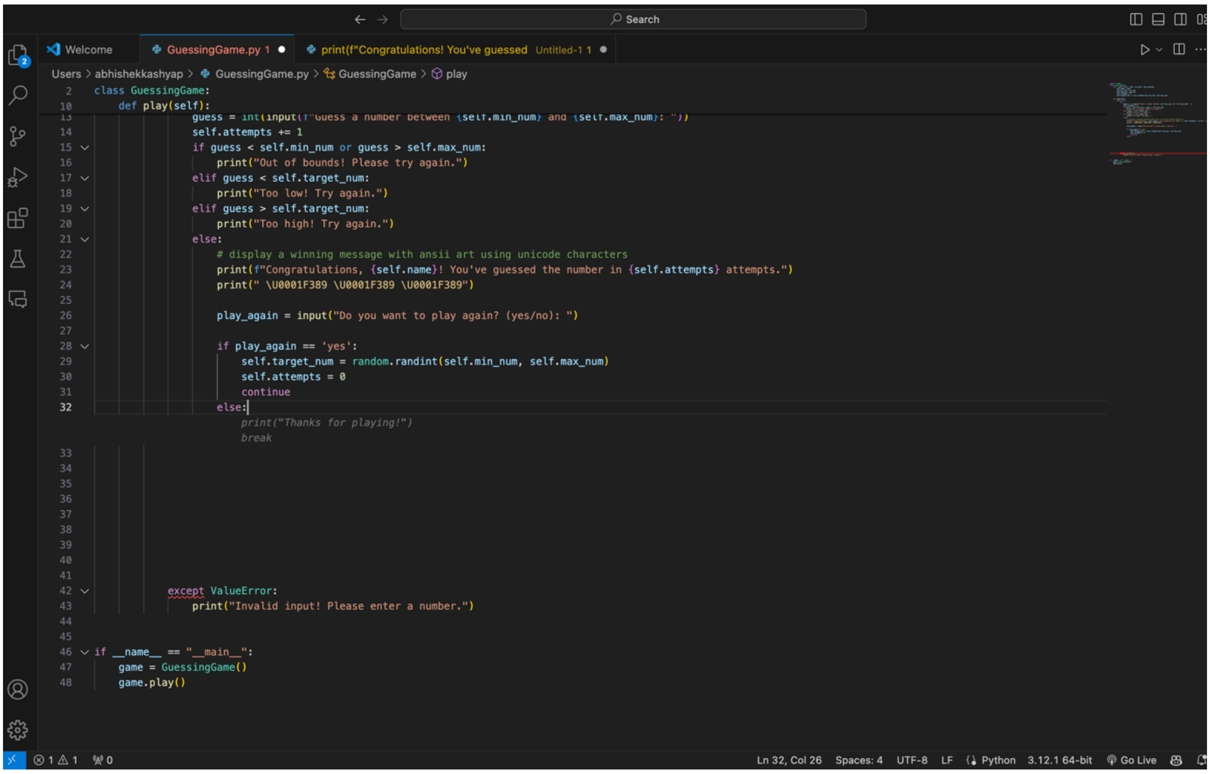
1. Code Completion and Context:

- **Open relevant files in your project.**
- **Write a code snippet where you need assistance (e.g., a function, class, or method).**
- **Observe how GitHub Copilot provides code suggestions based on the context.**
- **Take a screenshot of the code suggestion and include it in your assignment document.**

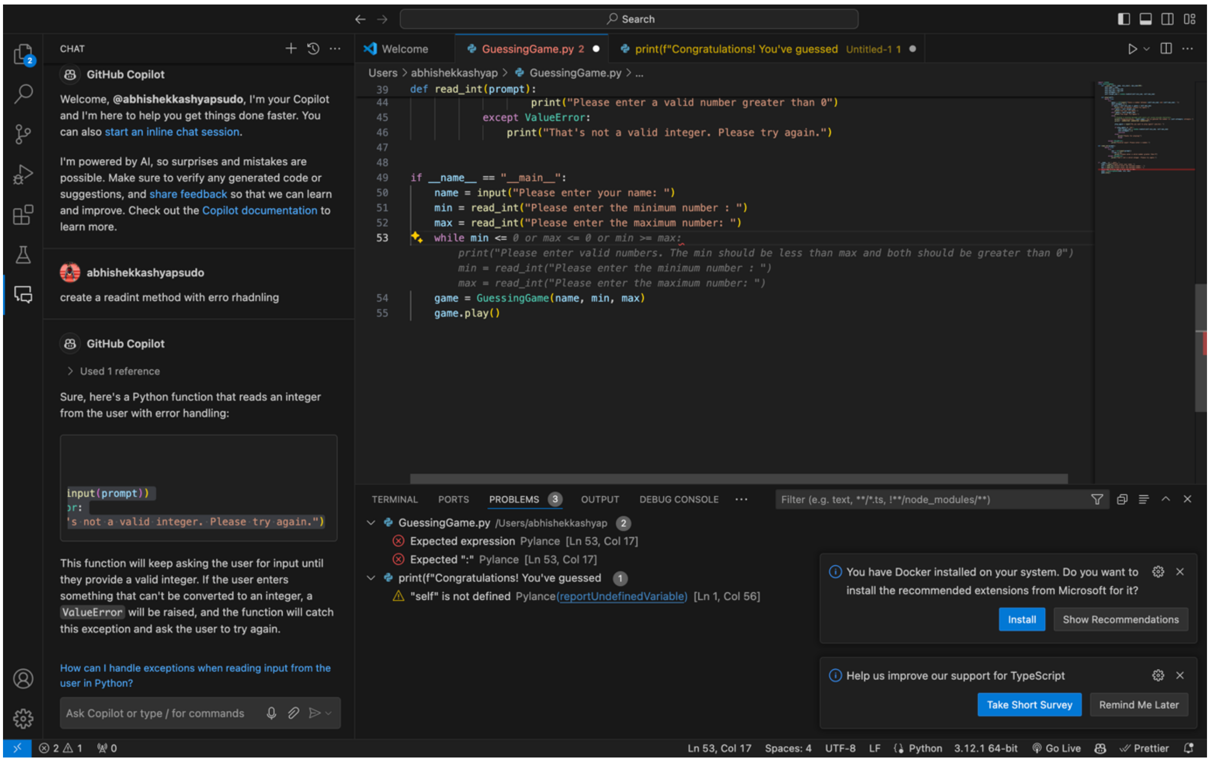
Screenshots showing the GitHub copilot providing suggestions:


Search
Welcome
GuessingGame.py 1
print(f"Congratulations! You've guessed Untitled-1 1
Users > abhishekkashyap > GuessingGame.py > GuessingGame > play
Ln 29, Col 21
Spaces: 4
UTF-8
LF
Python
3.12.1 64-bit
Go Live

```python
class GuessingGame:
    def play(self):
                guess = int(input(f"Guess a number between {self.min_num} and {self.max_num}: "))
                self.attempts += 1
                if guess < self.min_num or guess > self.max_num:
                    print("Out of bounds! Please try again.")
                elif guess < self.target_num:
                    print("Too low! Try again.")
                elif guess > self.target_num:
                    print("Too high! Try again.")
                else:
                    # display a winning message with ansii art using unicode characters
                    print(f"Congratulations, {self.name}! You've guessed the number in {self.attempts} attempts.")
                    print(" \U0001F389 \U0001F389 \U0001F389")

                    play_again = input("Do you want to play again? (yes/no): ")

                    if play_again == 'yes':
                    self.target_num = random.randint(self.min_num, self.max_num)
                        self.attempts = 0
                        continue










            except ValueError:
                print("Invalid input! Please enter a number.")


if __name__ == "__main__":
    game = GuessingGame()
    game.play()
```


Search
Welcome
GuessingGame.py 1
print(f"Congratulations! You've guessed Untitled-1 1
Users > abhishekkashyap > GuessingGame.py > GuessingGame > play
1 1
0
Ln 32, Col 26 Spaces: 4 UTF-8 LF Python 3.12.1 64-bit Go Live

```
class GuessingGame:
    def play(self):
                guess = int(input(f"Guess a number between {self.min_num} and {self.max_num}: "))
                self.attempts += 1
                if guess < self.min_num or guess > self.max_num:
                    print("Out of bounds! Please try again.")
                elif guess < self.target_num:
                    print("Too low! Try again.")
                elif guess > self.target_num:
                    print("Too high! Try again.")
                else:
                    # display a winning message with ansii art using unicode characters
                    print(f"Congratulations, {self.name}! You've guessed the number in {self.attempts} attempts.")
                    print(" \U0001F389 \U0001F389 \U0001F389")

                    play_again = input("Do you want to play again? (yes/no): ")

                    if play_again == 'yes':
                        self.target_num = random.randint(self.min_num, self.max_num)
                        self.attempts = 0
                        continue
                    else:
                        print("Thanks for playing!")
                        break









            except ValueError:
                print("Invalid input! Please enter a number.")


if __name__ == "__main__":
    game = GuessingGame()
    game.play()
```


Search
CHAT
Welcome
GuessingGame.py 2
print(f"Congratulations! You've guessed Untitled-1 1
Users > abhishekkashyap > GuessingGame.py > ...
TERMINAL
PORTS
PROBLEMS 3
OUTPUT
DEBUG CONSOLE
Filter (e.g. text, **/*.ts, !**/node_modules/**)
Install
Show Recommendations
Take Short Survey
Remind Me Later
Ask Copilot or type / for commands
2 1
0
Ln 53, Col 17 Spaces: 4 UTF-8 LF Python 3.12.1 64-bit Go Live Prettier

GitHub Copilot

Welcome, **@abhishekkashyapsudo**, I'm your Copilot and I'm here to help you get things done faster. You can also start an inline chat session.

I'm powered by AI, so surprises and mistakes are possible. Make sure to verify any generated code or suggestions, and share feedback so that we can learn and improve. Check out the Copilot documentation to learn more.

abhishekkashyapsudo

create a readint method with erro rhadnling

GitHub Copilot

> Used 1 reference

Sure, here's a Python function that reads an integer from the user with error handling:

```
input(prompt))
or:
's not a valid integer. Please try again.")
```

This function will keep asking the user for input until they provide a valid integer. If the user enters something that can't be converted to an integer, a `ValueError` will be raised, and the function will catch this exception and ask the user to try again.

How can I handle exceptions when reading input from the user in Python?

```
def read_int(prompt):
                    print("Please enter a valid number greater than 0")
            except ValueError:
                print("That's not a valid integer. Please try again.")


if __name__ == "__main__":
    name = input("Please enter your name: ")
    min = read_int("Please enter the minimum number : ")
    max = read_int("Please enter the maximum number: ")
    while min <= 0 or max <= 0 or min >= max:
        print("Please enter valid numbers. The min should be less than max and both should be greater than 0")
        min = read_int("Please enter the minimum number : ")
        max = read_int("Please enter the maximum number: ")
    game = GuessingGame(name, min, max)
    game.play()
```

```
GuessingGame.py /Users/abhishekkashyap 2
    Expected expression Pylance [Ln 53, Col 17]
    Expected ":" Pylance [Ln 53, Col 17]
print(f"Congratulations! You've guessed 1
    "self" is not defined Pylance(reportUndefinedVariable) [Ln 1, Col 56]
```

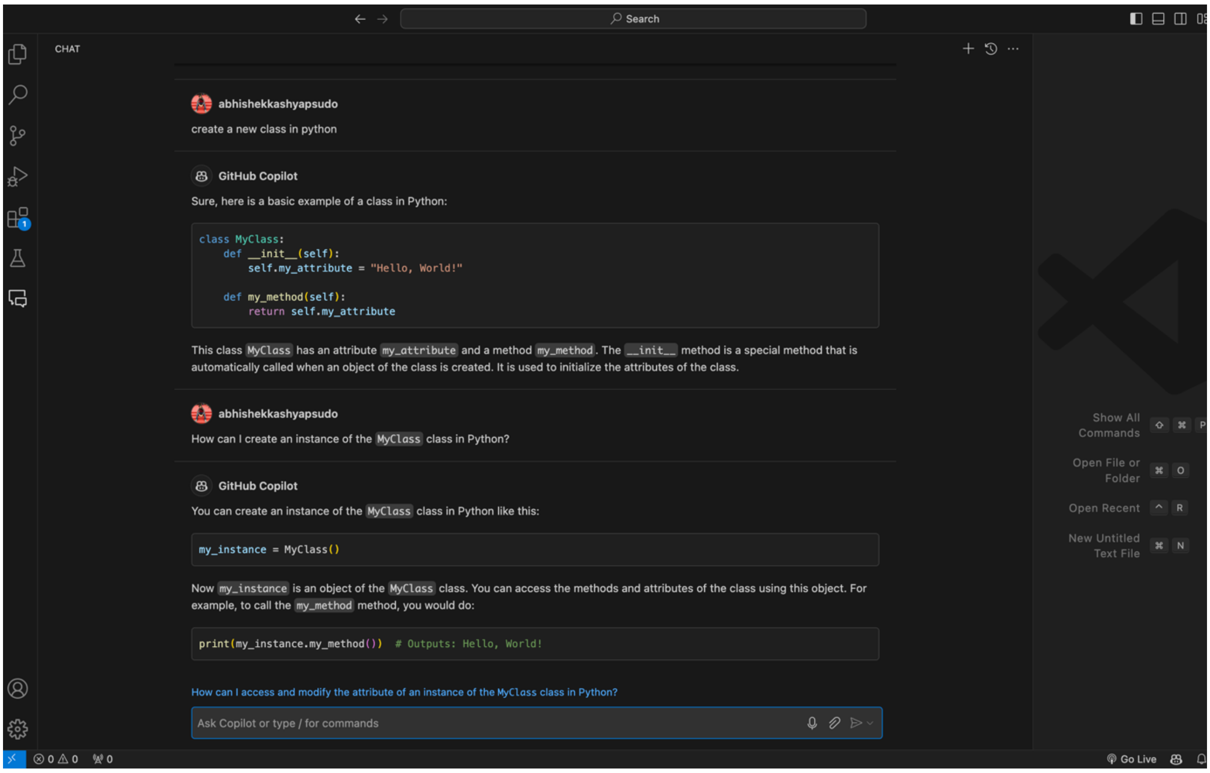
You have Docker installed on your system. Do you want to install the recommended extensions from Microsoft for it?

Help us improve our support for TypeScript

2. Chat Interface:

- **Use the chat interface within GitHub Copilot to ask for help with a specific coding problem.**
- **Describe the problem briefly and capture a screenshot of the chat interaction.**
- **Explain how the chat interface assisted you in solving the problem.**

Screenshots showing the GitHub chat interface:



abhishekkashyapsudo

create a new class in python

GitHub Copilot

Sure, here is a basic example of a class in Python:

```
class MyClass:
    def __init__(self):
        self.my_attribute = "Hello, World!"

    def my_method(self):
        return self.my_attribute
```

This class `MyClass` has an attribute `my_attribute` and a method `my_method`. The `__init__` method is a special method that is automatically called when an object of the class is created. It is used to initialize the attributes of the class.

abhishekkashyapsudo

How can I create an instance of the `MyClass` class in Python?

GitHub Copilot

You can create an instance of the `MyClass` class in Python like this:

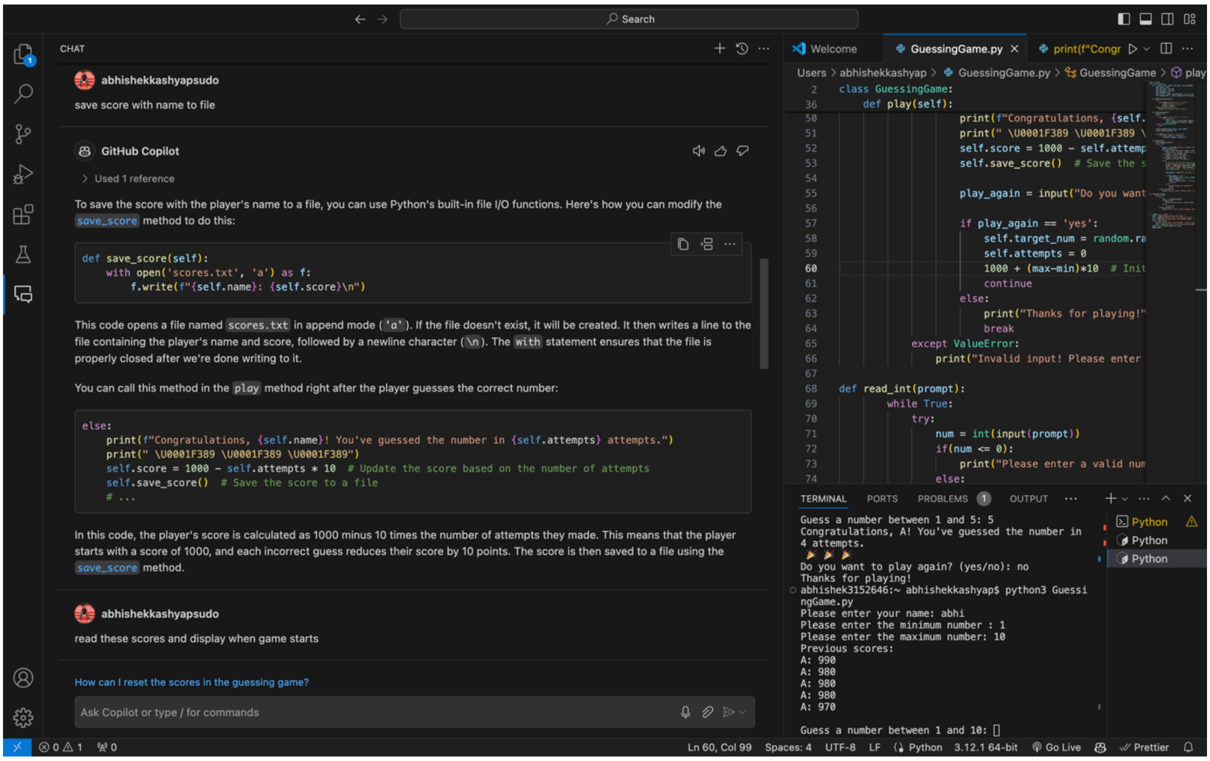
```
my_instance = MyClass()
```

Now `my_instance` is an object of the `MyClass` class. You can access the methods and attributes of the class using this object. For example, to call the `my_method` method, you would do:

```
print(my_instance.my_method())  # Outputs: Hello, World!
```

How can I access and modify the attribute of an instance of the MyClass class in Python?

Ask Copilot or type / for commands



abhishekkashyapsudo

save score with name to file

GitHub Copilot

> Used 1 reference

To save the score with the player's name to a file, you can use Python's built-in file I/O functions. Here's how you can modify the `save_score` method to do this:

```
def save_score(self):
    with open('scores.txt', 'a') as f:
        f.write(f"{self.name}: {self.score}\n")
```

This code opens a file named `scores.txt` in append mode (`'a'`). If the file doesn't exist, it will be created. It then writes a line to the file containing the player's name and score, followed by a newline character (`\n`). The `with` statement ensures that the file is properly closed after we're done writing to it.

You can call this method in the `play` method right after the player guesses the correct number:

```
else:
    print(f"Congratulations, {self.name}! You've guessed the number in {self.attempts} attempts.")
    print(" \U0001F389 \U0001F389 \U0001F389")
    self.score = 1000 - self.attempts * 10  # Update the score based on the number of attempts
    self.save_score()  # Save the score to a file
    # ...
```

In this code, the player's score is calculated as 1000 minus 10 times the number of attempts they made. This means that the player starts with a score of 1000, and each incorrect guess reduces their score by 10 points. The score is then saved to a file using the `save_score` method.

abhishekkashyapsudo

read these scores and display when game starts

How can I reset the scores in the guessing game?

Ask Copilot or type / for commands

```
class GuessingGame:
    def play(self):
                    print(f"Congratulations, {self.
                    print(" \U0001F389 \U0001F389 \
                    self.score = 1000 - self.attemp
                    self.save_score()  # Save the s

                    play_again = input("Do you want

                    if play_again == 'yes':
                        self.target_num = random.ra
                        self.attempts = 0
                        1000 + (max-min)*10  # Init
                        continue
                    else:
                        print("Thanks for playing!"
                        break
            except ValueError:
                print("Invalid input! Please enter

def read_int(prompt):
        while True:
            try:
                num = int(input(prompt))
                if(num <= 0):
                    print("Please enter a valid num
                else:
```

```
Guess a number between 1 and 5: 5
Congratulations, A! You've guessed the number in
4 attempts.
 🎉 🎉 🎉
Do you want to play again? (yes/no): no
Thanks for playing!
abhishek3152646:~ abhishekkashyap$ python3 Guessi
ngGame.py
Please enter your name: abhi
Please enter the minimum number : 1
Please enter the maximum number: 10
Previous scores:
A: 990
A: 980
A: 980
A: 980
A: 970

Guess a number between 1 and 10: 
```

Problem

As a Java developer, this was my first time using Python. I was unsure about how to create classes and read/write files in Python.

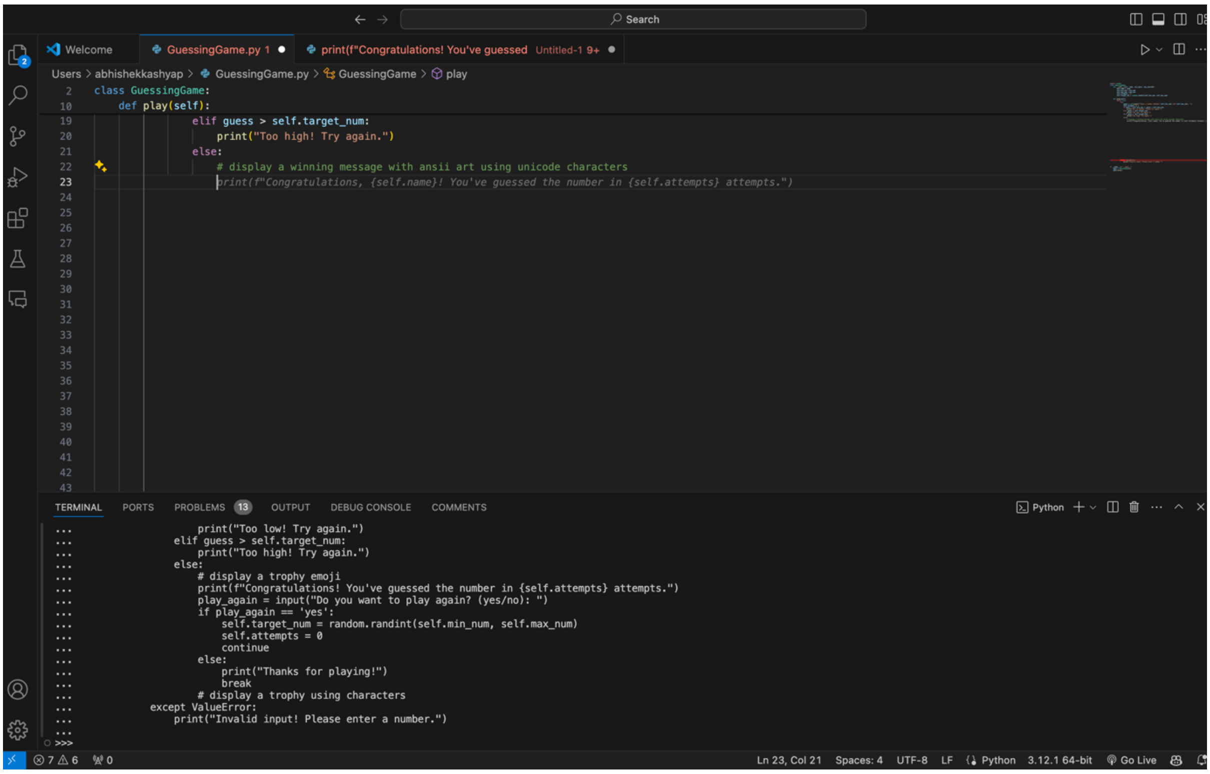
How Chat Interface assisted me in the problem:

I asked the chat interface how to create classes in Python and save scores to a file. The GitHub chat interface promptly provided code snippets and thorough explanations on how to achieve these tasks.

3. Ghost Text and Code Generation:

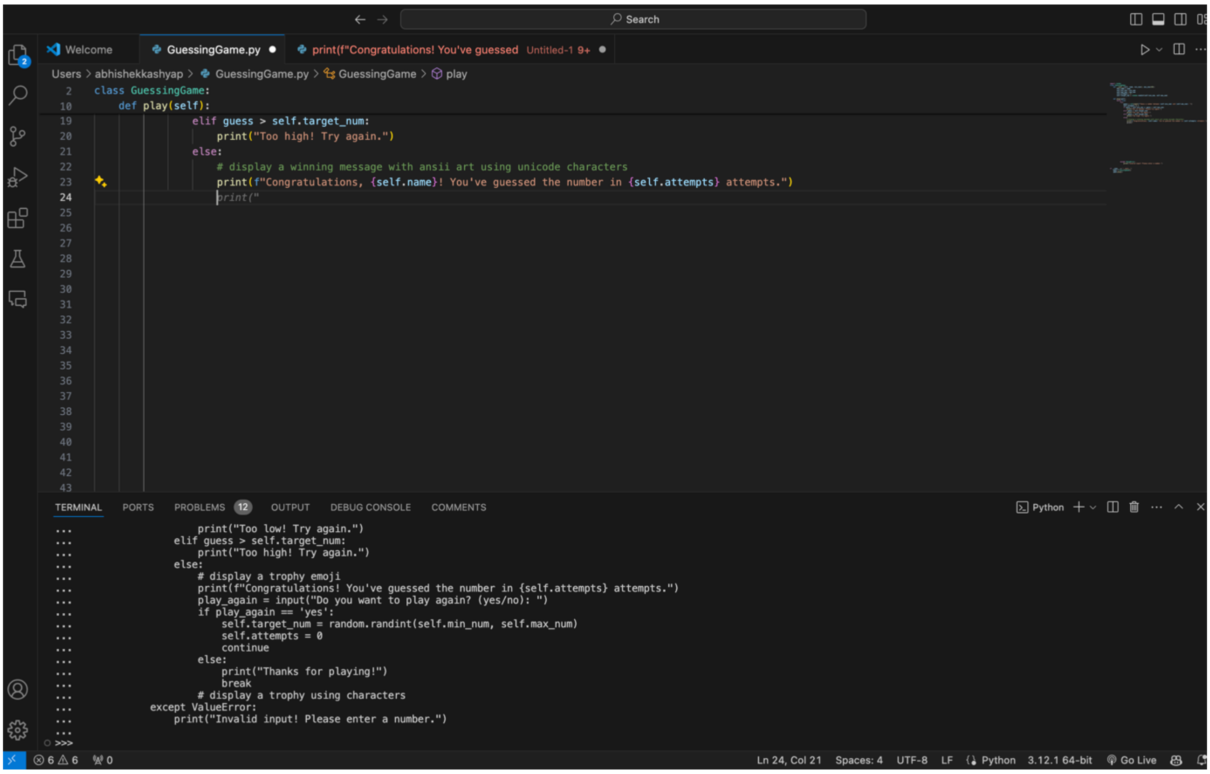
- **Write a comment or placeholder in your code (ghost text).**
- **Observe how GitHub Copilot generates code based on the ghost text.**
- **Provide an example of ghost text usage and include the generated code.**
- **Explain how this feature can improve your coding efficiency.**

Screenshots showing the GitHub Copilot generating ghost text on the basis of comments/placeholder:



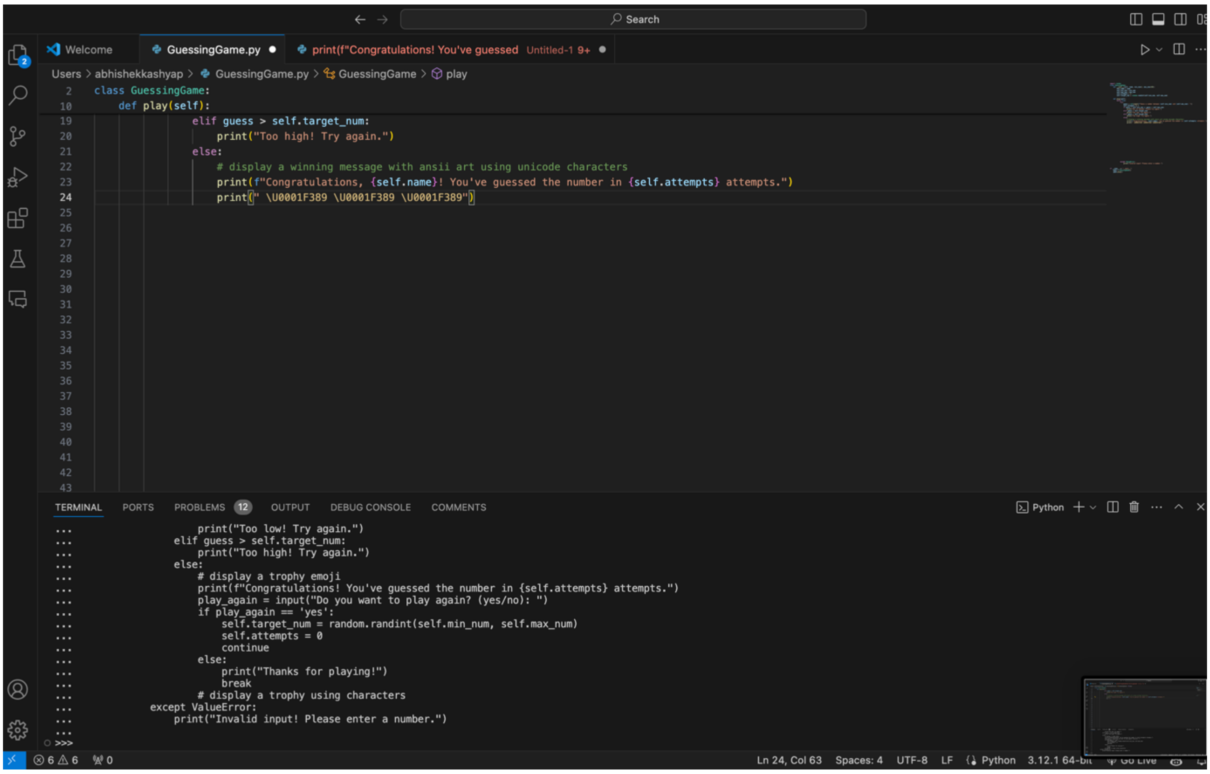
```
class GuessingGame:
    def play(self):
            elif guess > self.target_num:
                print("Too high! Try again.")
            else:
                # display a winning message with ansii art using unicode characters
                print(f"Congratulations, {self.name}! You've guessed the number in {self.attempts} attempts.")
```

```
...                 print("Too low! Try again.")
...             elif guess > self.target_num:
...                 print("Too high! Try again.")
...             else:
...                 # display a trophy emoji
...                 print(f"Congratulations! You've guessed the number in {self.attempts} attempts.")
...                 play_again = input("Do you want to play again? (yes/no): ")
...                 if play_again == 'yes':
...                     self.target_num = random.randint(self.min_num, self.max_num)
...                     self.attempts = 0
...                     continue
...                 else:
...                     print("Thanks for playing!")
...                     break
...                 # display a trophy using characters
...         except ValueError:
...             print("Invalid input! Please enter a number.")
...
>>>
```



```
class GuessingGame:
    def play(self):
            elif guess > self.target_num:
                print("Too high! Try again.")
            else:
                # display a winning message with ansii art using unicode characters
                print(f"Congratulations, {self.name}! You've guessed the number in {self.attempts} attempts.")
                print("
```

```
...                 print("Too low! Try again.")
...             elif guess > self.target_num:
...                 print("Too high! Try again.")
...             else:
...                 # display a trophy emoji
...                 print(f"Congratulations! You've guessed the number in {self.attempts} attempts.")
...                 play_again = input("Do you want to play again? (yes/no): ")
...                 if play_again == 'yes':
...                     self.target_num = random.randint(self.min_num, self.max_num)
...                     self.attempts = 0
...                     continue
...                 else:
...                     print("Thanks for playing!")
...                     break
...                 # display a trophy using characters
...         except ValueError:
...             print("Invalid input! Please enter a number.")
...
>>>
```



```python
class GuessingGame:
    def play(self):
                elif guess > self.target_num:
                    print("Too high! Try again.")
                else:
                    # display a winning message with ansii art using unicode characters
                    print(f"Congratulations, {self.name}! You've guessed the number in {self.attempts} attempts.")
                    print(" \U0001F389 \U0001F389 \U0001F389")
```

```
...                     print("Too low! Try again.")
...                 elif guess > self.target_num:
...                     print("Too high! Try again.")
...                 else:
...                     # display a trophy emoji
...                     print(f"Congratulations! You've guessed the number in {self.attempts} attempts.")
...                     play_again = input("Do you want to play again? (yes/no): ")
...                     if play_again == 'yes':
...                         self.target_num = random.randint(self.min_num, self.max_num)
...                         self.attempts = 0
...                         continue
...                     else:
...                         print("Thanks for playing!")
...                         break
...                     # display a trophy using characters
...             except ValueError:
...                 print("Invalid input! Please enter a number.")
...
>>>
```

- **Explain how this feature can improve your coding efficiency.**

  This feature was incredibly helpful while writing my code. As soon as I started coding and adding comments or placeholders to outline my strategy, GitHub Copilot quickly provided relevant suggestions. This significantly increased my development speed and decreased the likelihood of missing any critical scenarios while planning the solution.

4. Inline Chat and Collaboration:

- **Collaborate with a colleague or friend using GitHub Copilot's inline chat.**
- **Share code snippets, discuss solutions, and make real-time edits.**
- **Describe your experience with inline chat and its benefits.**

5. Best Practices and Dos and Don'ts:

- **Research best practices for using GitHub Copilot.**
- **List at least three dos and three don'ts when working with the tool.**
- **Explain why following these practices is essential.**

- **Research best practices for using GitHub Copilot.**

1. Don’t rely blindly on suggestions. Always review and understand the code Copilot generates.
2. Treat Copilot as a tool to enhance your productivity, not replace your coding skills.
3. Keep up with the latest updates and improvements to GitHub Copilot to make the most of new features and capabilities.

- **List at least three dos and three don’ts when working with the tool.**

Dos

1. Always review the code suggestions to ensure they meet your project’s requirements and follow best practices.
2. Use comments and placeholder text to guide Copilot in generating code that fits your specific needs.
3. Leverage Copilot for repetitive or boilerplate code to save time and reduce errors.

Don’ts

1. Don’t rely solely on Copilot for critical code without understanding and verifying it yourself.
2. Avoid using generated code that could introduce security vulnerabilities without thorough review.
3. Don’t use Copilot to generate code for sensitive or proprietary algorithms without careful consideration and review.

- **Explain why following these practices is essential.**

1. By reviewing and understanding the generated code, you ensure that it meets quality standards and is maintainable.
2. Following best practices helps prevent introducing security vulnerabilities through automatically generated code.
3. Using Copilot for appropriate tasks like boilerplate code can significantly enhance productivity, allowing developers to focus on more complex aspects of their projects.
4. By customizing suggestions and reviewing generated code, developers can learn and improve their coding skills, rather than becoming overly dependent on the tool.

6. Security and Privacy Considerations: Unauthorized Code Access

- **GitHub Copilot may inadvertently generate code that exposes sensitive information, such as authentication tokens or API keys.**
- **Discuss the risks associated with exposing such information in code comments.**
- **Explain the potential impact if an unauthorized person gains access to these tokens.**
- **Propose preventive measures to avoid accidental exposure.**

- **Discuss the risks associated with exposing such information in code comments.**

1. GitHub Copilot may inadvertently suggest code that includes sensitive information such as authentication tokens or API keys, especially if it learns from public code repositories that contain such data.
2. If these tokens or keys are exposed in code comments, anyone with access to the code can potentially see and misuse them, leading to security breaches.
3. Sensitive information exposed in comments can be inadvertently copied into other parts of the codebase or shared publicly, further increasing the risk of unauthorized access.

- **Explain the potential impact if an unauthorized person gains access to these tokens.**

1. If an unauthorized person gains access to authentication tokens or API keys, they could gain unauthorized access to your services, APIs, or databases, leading to data breaches.
2. Exposed tokens can be used to steal sensitive data, compromising user privacy and organizational security.
3. Data breaches can result in significant legal repercussions and non-compliance with data protection regulations, potentially leading to fines and legal action.

- **Propose preventive measures to avoid accidental exposure.**

    1. **Review and Filter Suggestions**:
        - 1.1. Always review the code suggestions provided by GitHub Copilot. Ensure no sensitive information is included before integrating the code into your project.
        - 1.2. Use specific comments and placeholders to guide Copilot’s suggestions, avoiding any references to actual sensitive data.

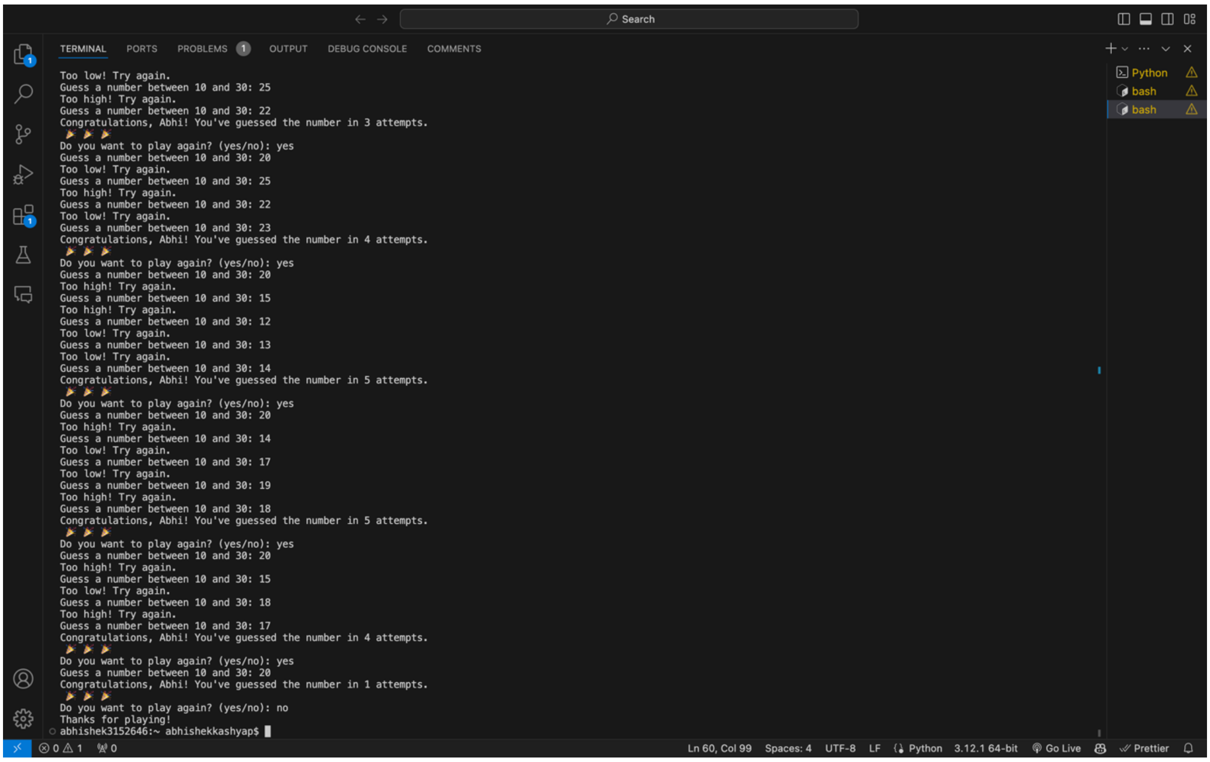
2. **Environment Variables**:
    2.1. Store sensitive information like API keys and tokens in environment variables rather than hard-coding them in the code.
    2.2. Use configuration management tools to handle environment variables securely.

3. **Code Scanning and Auditing**:
    3.1. Implement automated code scanning tools to detect and alert for any hard-coded sensitive information in your codebase.
    3.2. Conduct regular code audits to ensure that no sensitive information is exposed.

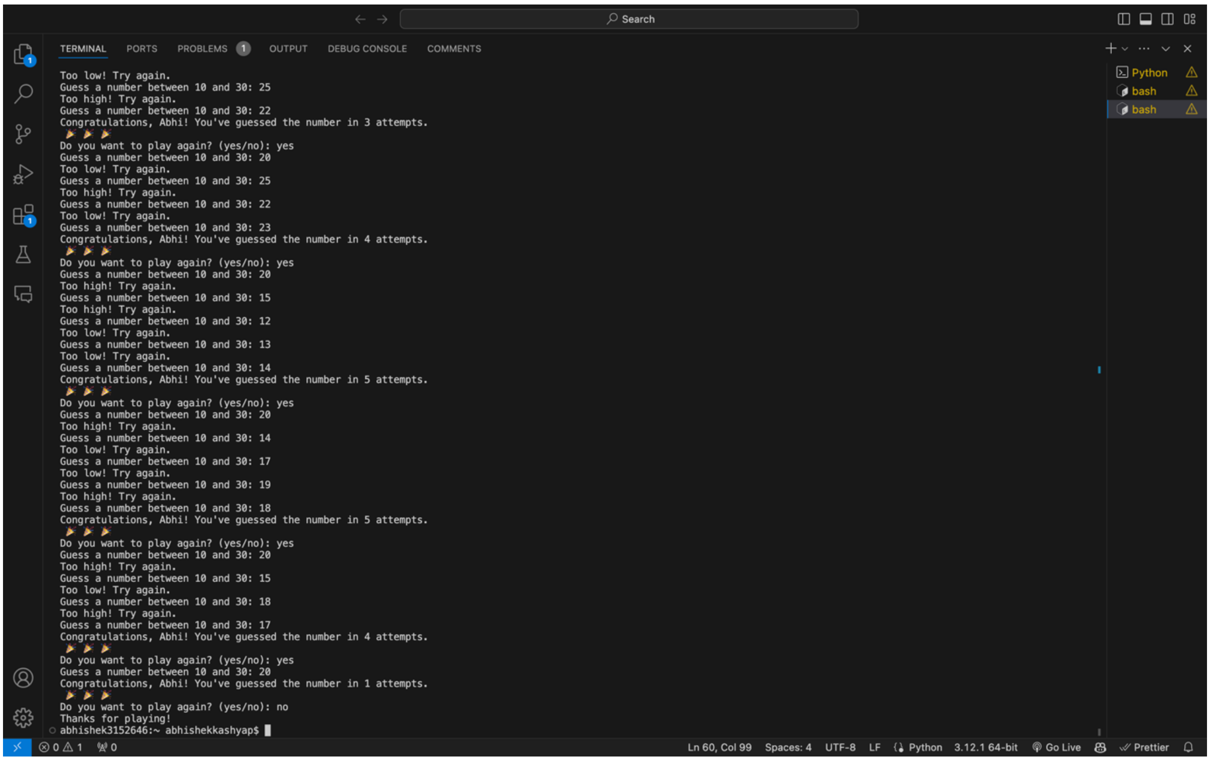
4. **Access Control**:
    4.1. Restrict access to the code repository to only those who need it. Use role-based access control to minimize the risk of unauthorized access.
    4.2. Monitor access logs and set up alerts for any unusual access patterns to the repository.

5. **Education and Awareness**:
    5.1. Educate developers about the risks of exposing sensitive information and best practices for handling it securely.
    5.2. Provide clear documentation on how to manage and store sensitive data properly within the organization’s coding standards.

7. Game Screenshots



```
Too low! Try again.
Guess a number between 10 and 30: 25
Too high! Try again.
Guess a number between 10 and 30: 22
Congratulations, Abhi! You've guessed the number in 3 attempts.
 🎉 🎉 🎉
Do you want to play again? (yes/no): yes
Guess a number between 10 and 30: 20
Too low! Try again.
Guess a number between 10 and 30: 25
Too high! Try again.
Guess a number between 10 and 30: 22
Too low! Try again.
Guess a number between 10 and 30: 23
Congratulations, Abhi! You've guessed the number in 4 attempts.
 🎉 🎉 🎉
Do you want to play again? (yes/no): yes
Guess a number between 10 and 30: 20
Too high! Try again.
Guess a number between 10 and 30: 15
Too high! Try again.
Guess a number between 10 and 30: 12
Too low! Try again.
Guess a number between 10 and 30: 13
Too low! Try again.
Guess a number between 10 and 30: 14
Congratulations, Abhi! You've guessed the number in 5 attempts.
 🎉 🎉 🎉
Do you want to play again? (yes/no): yes
Guess a number between 10 and 30: 20
Too high! Try again.
Guess a number between 10 and 30: 14
Too low! Try again.
Guess a number between 10 and 30: 17
Too low! Try again.
Guess a number between 10 and 30: 19
Too high! Try again.
Guess a number between 10 and 30: 18
Congratulations, Abhi! You've guessed the number in 5 attempts.
 🎉 🎉 🎉
Do you want to play again? (yes/no): yes
Guess a number between 10 and 30: 20
Too high! Try again.
Guess a number between 10 and 30: 15
Too low! Try again.
Guess a number between 10 and 30: 18
Too high! Try again.
Guess a number between 10 and 30: 17
Congratulations, Abhi! You've guessed the number in 4 attempts.
 🎉 🎉 🎉
Do you want to play again? (yes/no): yes
Guess a number between 10 and 30: 20
Congratulations, Abhi! You've guessed the number in 1 attempts.
 🎉 🎉 🎉
Do you want to play again? (yes/no): no
Thanks for playing!
○ abhishek3152646:~ abhishekkashyap$ █
```



```
Too low! Try again.
Guess a number between 10 and 30: 25
Too high! Try again.
Guess a number between 10 and 30: 22
Congratulations, Abhi! You've guessed the number in 3 attempts.
 🎉 🎉 🎉
Do you want to play again? (yes/no): yes
Guess a number between 10 and 30: 20
Too low! Try again.
Guess a number between 10 and 30: 25
Too high! Try again.
Guess a number between 10 and 30: 22
Too low! Try again.
Guess a number between 10 and 30: 23
Congratulations, Abhi! You've guessed the number in 4 attempts.
 🎉 🎉 🎉
Do you want to play again? (yes/no): yes
Guess a number between 10 and 30: 20
Too high! Try again.
Guess a number between 10 and 30: 15
Too high! Try again.
Guess a number between 10 and 30: 12
Too low! Try again.
Guess a number between 10 and 30: 13
Too low! Try again.
Guess a number between 10 and 30: 14
Congratulations, Abhi! You've guessed the number in 5 attempts.
 🎉 🎉 🎉
Do you want to play again? (yes/no): yes
Guess a number between 10 and 30: 20
Too high! Try again.
Guess a number between 10 and 30: 14
Too low! Try again.
Guess a number between 10 and 30: 17
Too low! Try again.
Guess a number between 10 and 30: 19
Too high! Try again.
Guess a number between 10 and 30: 18
Congratulations, Abhi! You've guessed the number in 5 attempts.
 🎉 🎉 🎉
Do you want to play again? (yes/no): yes
Guess a number between 10 and 30: 20
Too high! Try again.
Guess a number between 10 and 30: 15
Too low! Try again.
Guess a number between 10 and 30: 18
Too high! Try again.
Guess a number between 10 and 30: 17
Congratulations, Abhi! You've guessed the number in 4 attempts.
 🎉 🎉 🎉
Do you want to play again? (yes/no): yes
Guess a number between 10 and 30: 20
Congratulations, Abhi! You've guessed the number in 1 attempts.
 🎉 🎉 🎉
Do you want to play again? (yes/no): no
Thanks for playing!
○ abhishek3152646:~ abhishekkashyap$ █
```

Sensitivity: Internal