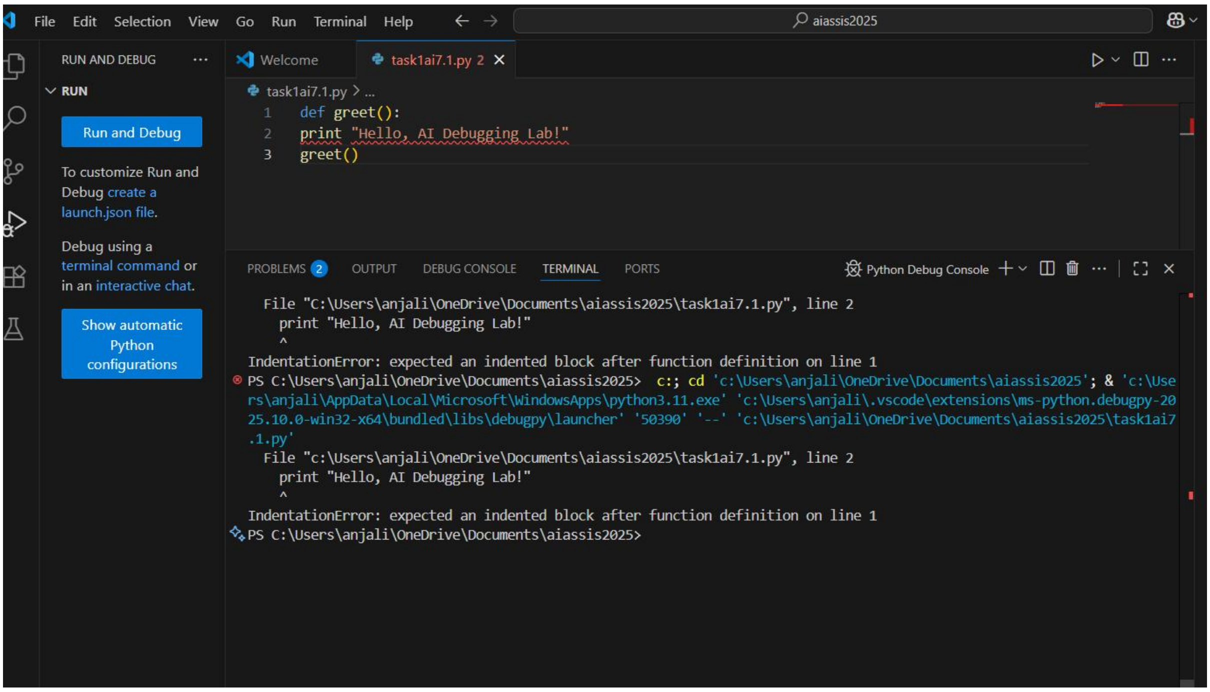
Assignment-7.1

2403A51276

Mohammed Ibrahim Hussain

TASK-1

Running given code:



File Edit Selection View Go Run Terminal Help
aiassis2025
RUN AND DEBUG
RUN
Run and Debug
To customize Run and Debug create a launch.json file.
Debug using a terminal command or in an interactive chat.
Show automatic Python configurations
Welcome
task1ai7.1.py 2
task1ai7.1.py > ...
PROBLEMS 2
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Python Debug Console

```python
def greet():
print "Hello, AI Debugging Lab!"
greet()
```

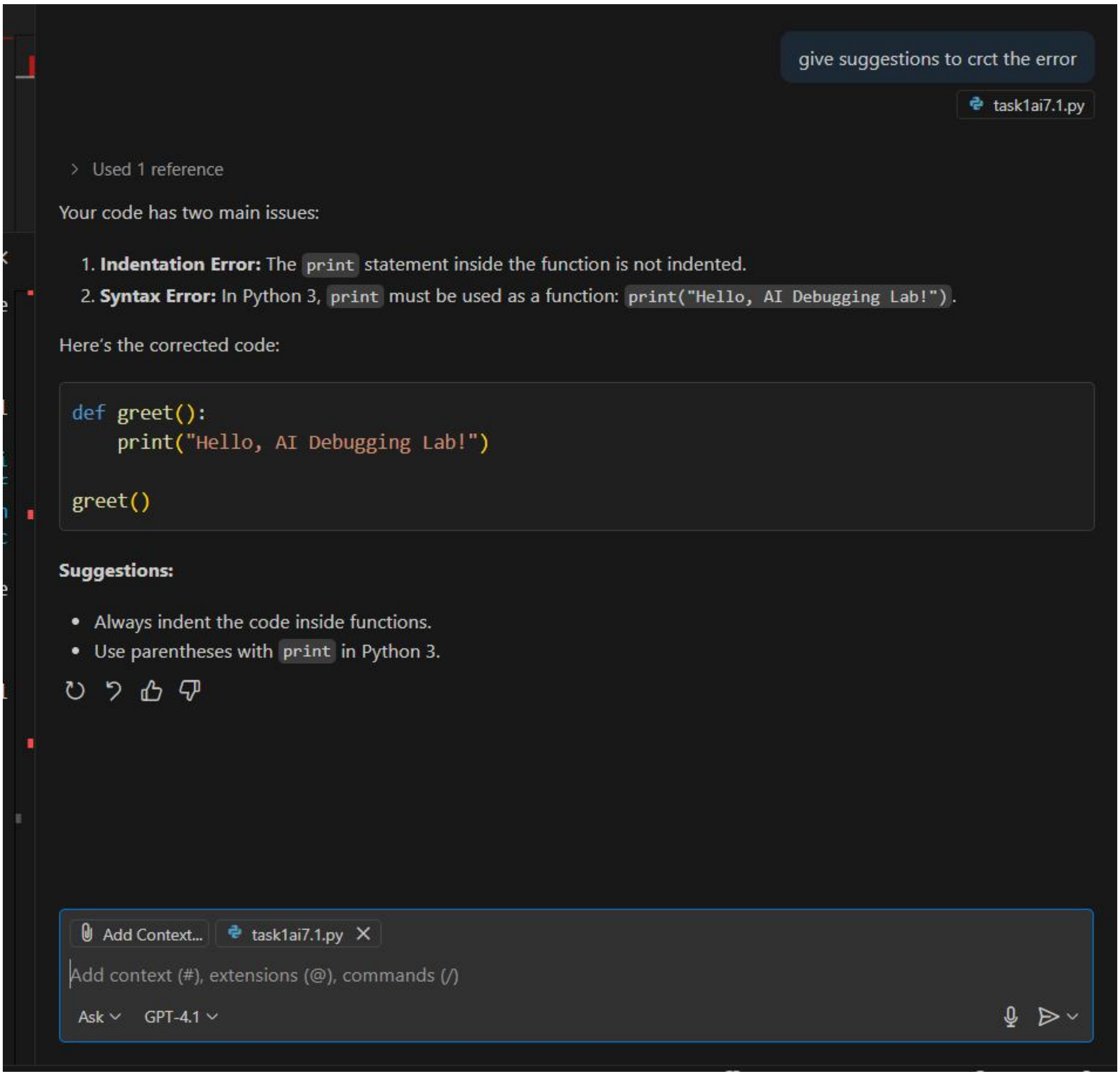
```
  File "C:\Users\anjali\OneDrive\Documents\aiassis2025\task1ai7.1.py", line 2
    print "Hello, AI Debugging Lab!"
    ^
IndentationError: expected an indented block after function definition on line 1
PS C:\Users\anjali\OneDrive\Documents\aiassis2025>  c:; cd 'c:\Users\anjali\OneDrive\Documents\aiassis2025'; & 'c:\Use
rs\anjali\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\anjali\.vscode\extensions\ms-python.debugpy-20
25.10.0-win32-x64\bundled\libs\debugpy\launcher' '50390' '--' 'c:\Users\anjali\OneDrive\Documents\aiassis2025\task1ai7
.1.py'
  File "c:\Users\anjali\OneDrive\Documents\aiassis2025\task1ai7.1.py", line 2
    print "Hello, AI Debugging Lab!"
    ^
IndentationError: expected an indented block after function definition on line 1
PS C:\Users\anjali\OneDrive\Documents\aiassis2025>
```

Errors:

1. **Indentation Error:** The print statement inside the function is not indented.
2. **Syntax Error:** In Python 3, print must be used as a function

AI suggestions:

- Always indent the code inside functions.
- Use parentheses with print in Python 3.



give suggestions to crct the error

Your code has two main issues:

1. **Indentation Error:** The `print` statement inside the function is not indented.
2. **Syntax Error:** In Python 3, `print` must be used as a function: `print("Hello, AI Debugging Lab!")`.

Here's the corrected code:

```python
def greet():
    print("Hello, AI Debugging Lab!")

greet()
```

Suggestions:

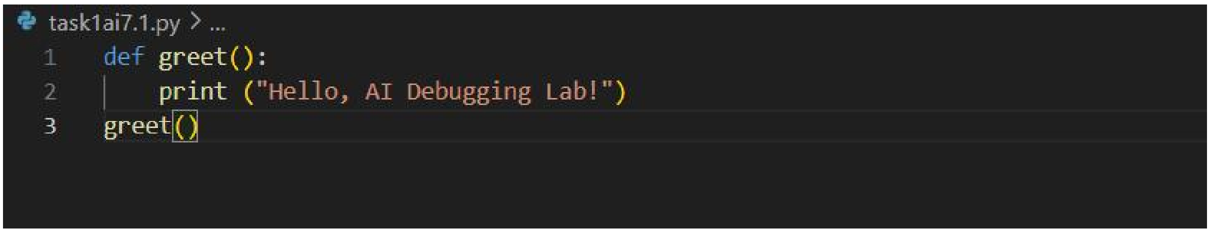
- Always indent the code inside functions.
- Use parentheses with `print` in Python 3.

Corrected Syntax:

def greet():

print ("Hello, AI Debugging Lab!")

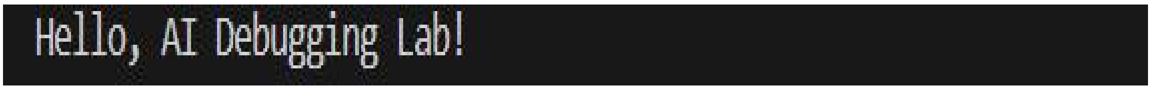
greet()



```python
def greet():
    print ("Hello, AI Debugging Lab!")
greet()
```

OUTPUT:

Hello, AI Debugging Lab!



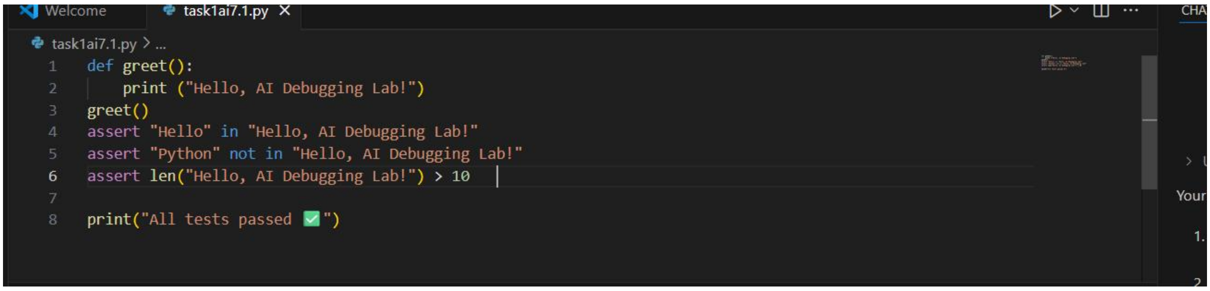
```
Hello, AI Debugging Lab!
```

Explanation of code:

- The function greet() prints the message "Hello, AI Debugging Lab!".
- The function is called, so the message is displayed.

Assert cases:

```
def greet():
    print ("Hello, AI Debugging Lab!")
greet()
assert "Hello" in "Hello, AI Debugging Lab!"
assert "Python" not in "Hello, AI Debugging Lab!"
assert len("Hello, AI Debugging Lab!") > 10
print("All tests passed ✅")
```

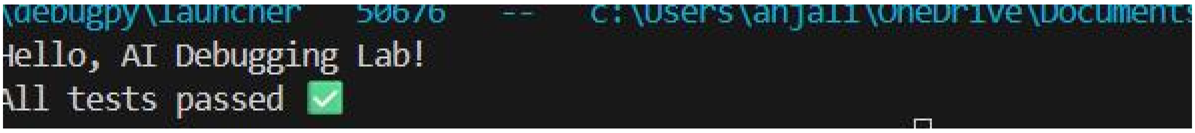


```
def greet():
    print ("Hello, AI Debugging Lab!")
greet()
assert "Hello" in "Hello, AI Debugging Lab!"
assert "Python" not in "Hello, AI Debugging Lab!"
assert len("Hello, AI Debugging Lab!") > 10

print("All tests passed ✅")
```

Output:

Hello, AI Debugging Lab!

All tests passed ✅

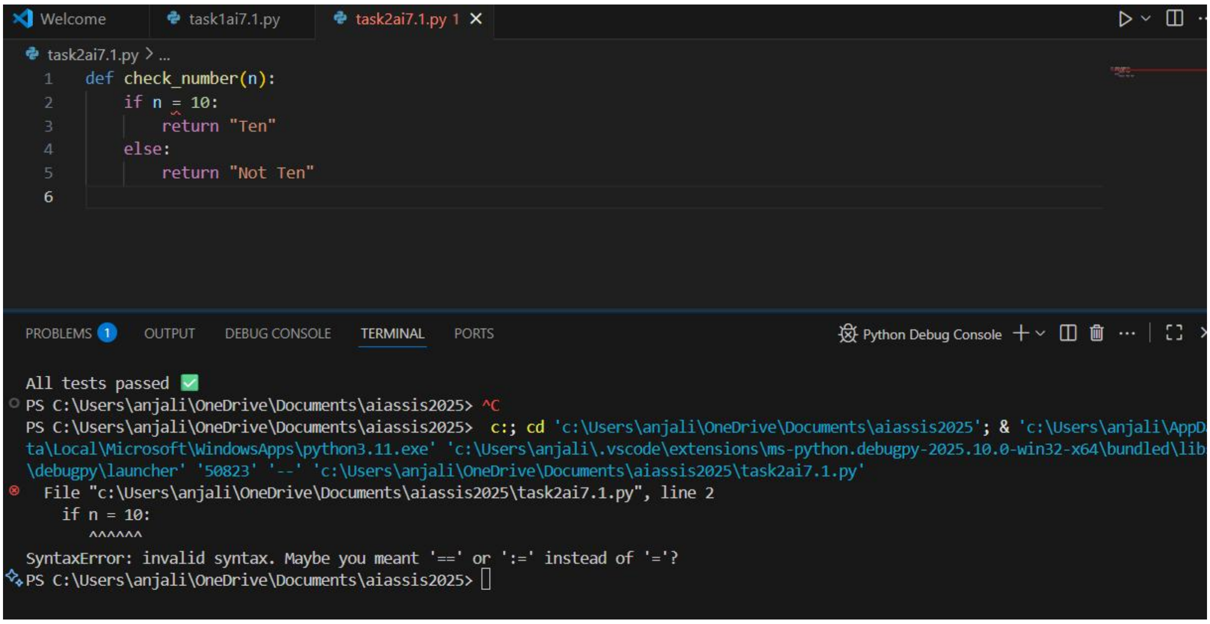


```
\debugpy\launcher  50676  --  c:\Users\anjali\OneDrive\Documents
Hello, AI Debugging Lab!
All tests passed ✅
```

TASK-2

Logic Error – Incorrect Condition in an If Statement

Running given code:



```python
def check_number(n):
    if n = 10:
        return "Ten"
    else:
        return "Not Ten"
```

```
All tests passed ✅
PS C:\Users\anjali\OneDrive\Documents\aiassis2025> ^C
PS C:\Users\anjali\OneDrive\Documents\aiassis2025>  c:; cd 'c:\Users\anjali\OneDrive\Documents\aiassis2025'; & 'c:\Users\anjali\AppD
ta\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\anjali\.vscode\extensions\ms-python.debugpy-2025.10.0-win32-x64\bundled\lib
\debugpy\launcher' '50823' '--' 'c:\Users\anjali\OneDrive\Documents\aiassis2025\task2ai7.1.py'
  File "c:\Users\anjali\OneDrive\Documents\aiassis2025\task2ai7.1.py", line 2
    if n = 10:
       ^^^^^^
SyntaxError: invalid syntax. Maybe you meant '==' or ':=' instead of '='?
PS C:\Users\anjali\OneDrive\Documents\aiassis2025> 
```

Errors:

```
if n = 10:

^^^^^^

SyntaxError: invalid syntax. Maybe you meant '==' or ':=' instead of '='?
```

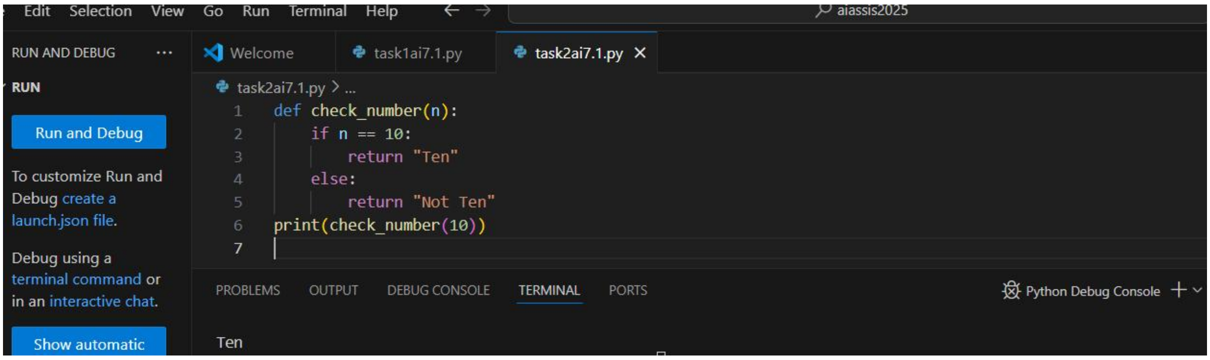
SyntaxError caused by using = instead of == in the if statement.

AI suggestions:

1. **Use == for comparison:**
   Replace if n = 10: with if n == 10:.
   = is for assignment, == is for comparison.

Corrected syntax:

```python
def check_number(n):

    if n == 10:

        return "Ten"

    else:

        return "Not Ten"

print(check_number(10))
```



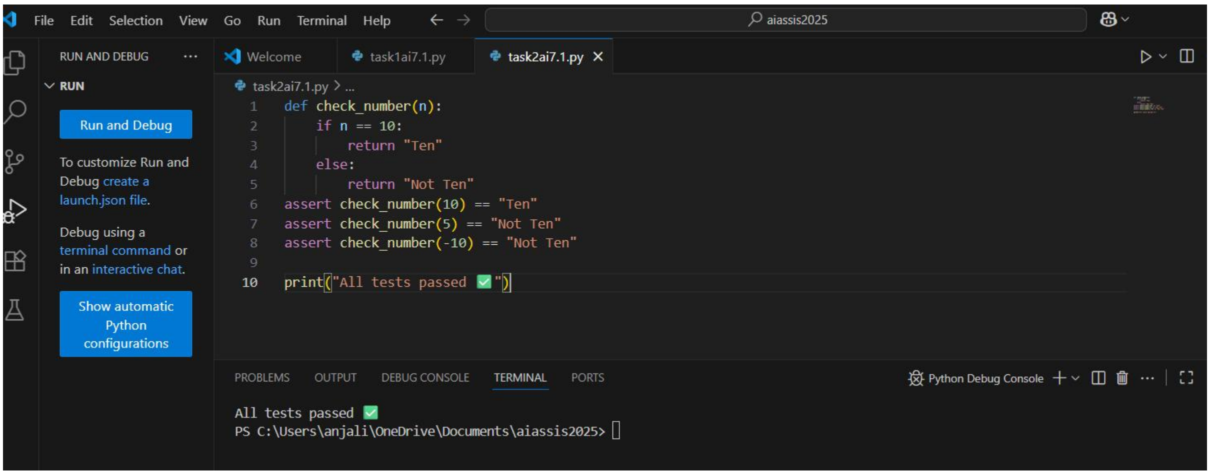
```python
def check_number(n):
    if n == 10:
        return "Ten"
    else:
        return "Not Ten"
print(check_number(10))
```

```
Ten
```

Output:

Ten

Assert cases:

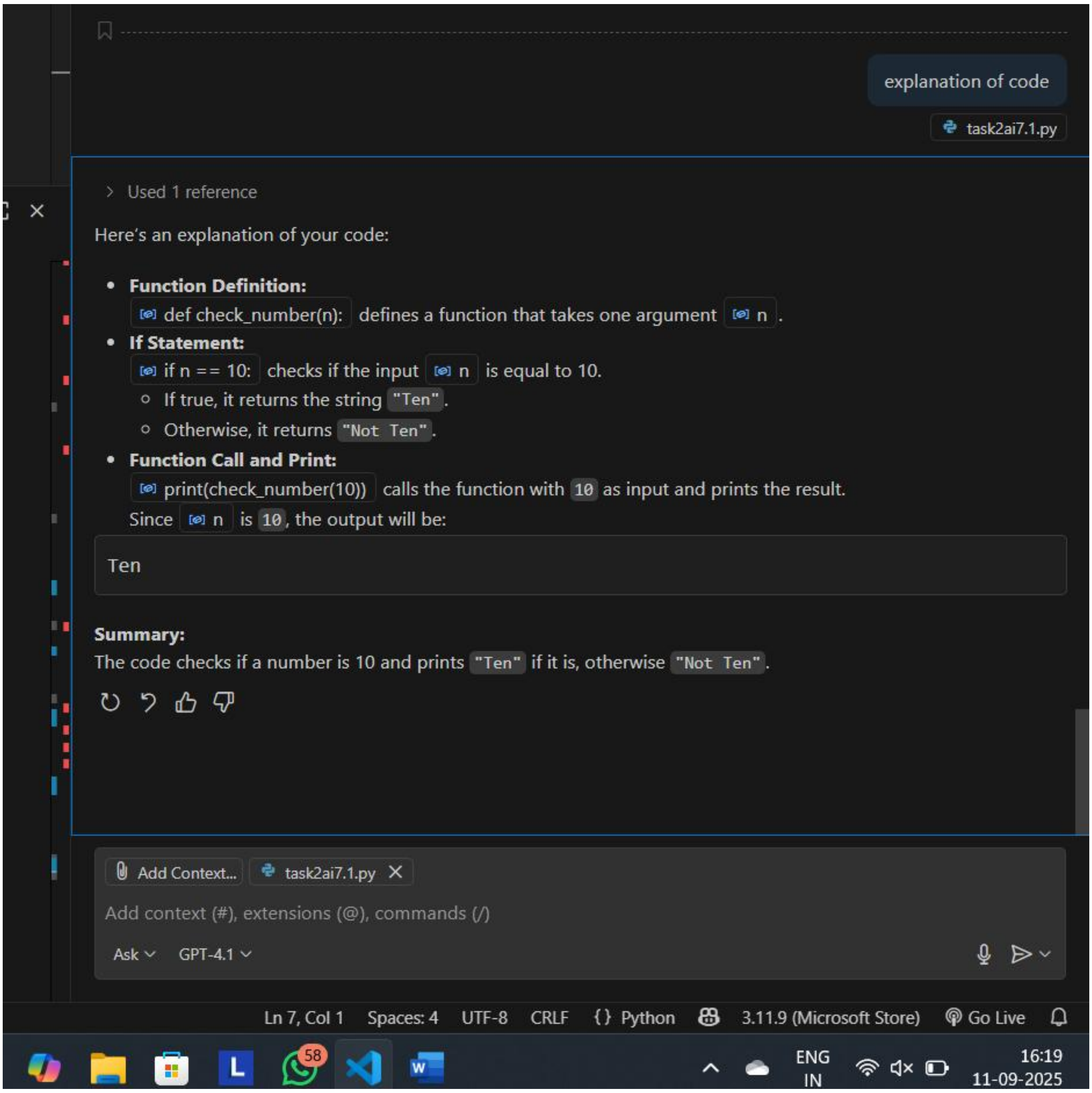


```python
def check_number(n):
    if n == 10:
        return "Ten"
    else:
        return "Not Ten"
assert check_number(10) == "Ten"
assert check_number(5) == "Not Ten"
assert check_number(-10) == "Not Ten"

print("All tests passed ✅")
```

```
All tests passed ✅
PS C:\Users\anjali\OneDrive\Documents\aiassis2025>
```

AI explanation:

- **Function Definition:**
  def check_number(n): defines a function that takes one argument n.
- **If Statement:**
  if n == 10: checks if the input n is equal to 10.
    - If true, it returns the string "Ten".
    - Otherwise, it returns "Not Ten".
- **Function Call and Print:**
  print(check_number(10)) calls the function with 10 as input and prints the result.



Here's an explanation of your code:

- **Function Definition:**
  `def check_number(n):` defines a function that takes one argument `n`.
- **If Statement:**
  `if n == 10:` checks if the input `n` is equal to 10.
  - If true, it returns the string `"Ten"`.
  - Otherwise, it returns `"Not Ten"`.
- **Function Call and Print:**
  `print(check_number(10))` calls the function with `10` as input and prints the result.
  Since `n` is `10`, the output will be:

```
Ten
```

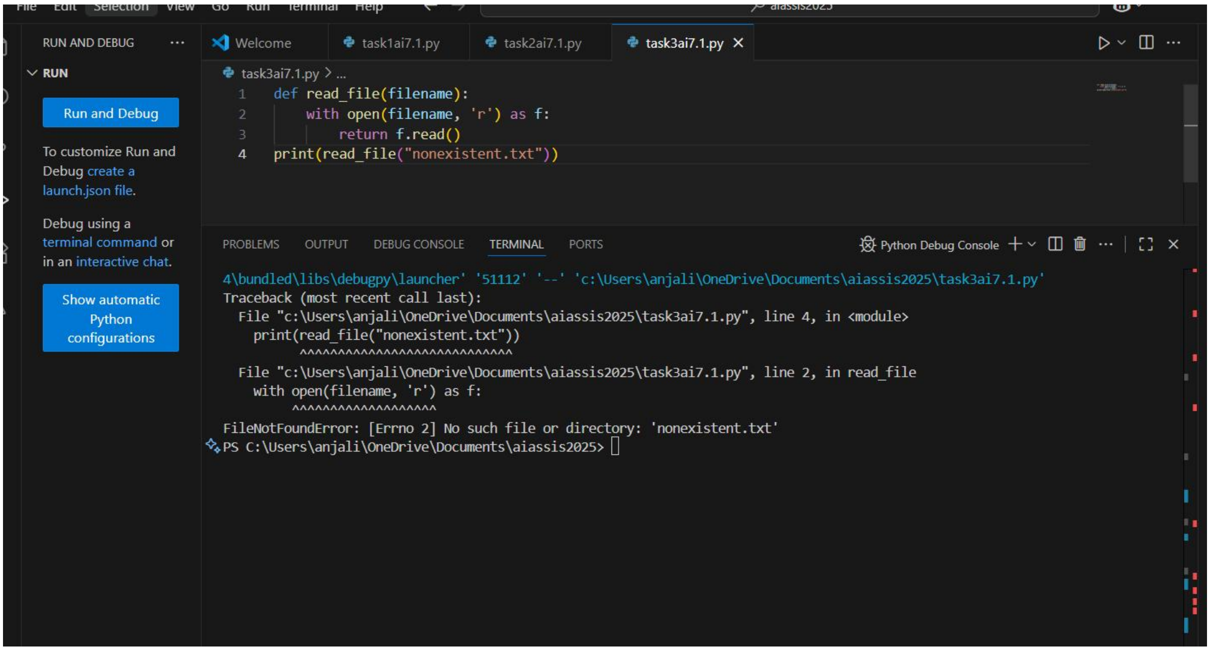
Summary:
The code checks if a number is 10 and prints `"Ten"` if it is, otherwise `"Not Ten"`.

TASK-3:

File Handling:

Running given code:

```
def read_file(filename):
    with open(filename, 'r') as f:
        return f.read()
print(read_file("nonexistent.txt"))
```



To customize Run and Debug create a launch.json file.

Debug using a terminal command or in an interactive chat.

```python
def read_file(filename):
    with open(filename, 'r') as f:
        return f.read()
print(read_file("nonexistent.txt"))
```

```
4\bundled\libs\debugpy\launcher' '51112' '--' 'c:\Users\anjali\OneDrive\Documents\aiassis2025\task3ai7.1.py'
Traceback (most recent call last):
  File "c:\Users\anjali\OneDrive\Documents\aiassis2025\task3ai7.1.py", line 4, in <module>
    print(read_file("nonexistent.txt"))
          ^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "c:\Users\anjali\OneDrive\Documents\aiassis2025\task3ai7.1.py", line 2, in read_file
    with open(filename, 'r') as f:
         ^^^^^^^^^^^^^^^^^^^
FileNotFoundError: [Errno 2] No such file or directory: 'nonexistent.txt'
PS C:\Users\anjali\OneDrive\Documents\aiassis2025> 
```

Errors:

- The function tries to open "nonexistent.txt" for reading.
- Since the file does **not exist**, Python raises a FileNotFoundError.
- There is **no try-except block** to handle this error.
- As a result, the program will crash and display an error message

Suggestions:

Add a try-except block for user-friendly error handling

Prompt used:

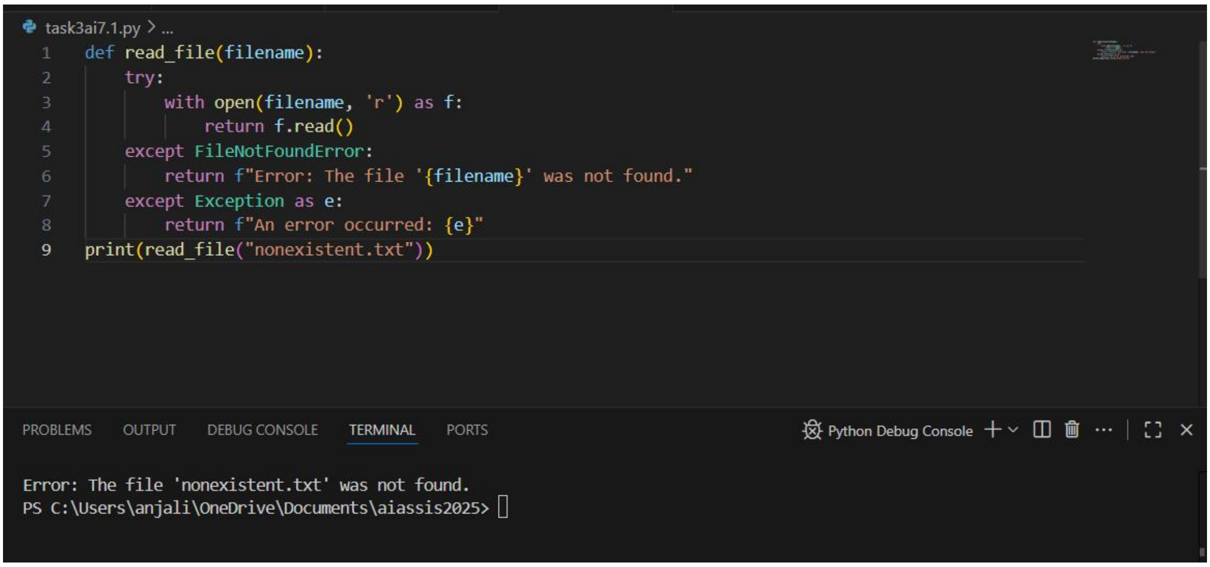
Implement a try-except block and Add a user-friendly error message on

```
def read_file(filename):
    with open(filename, 'r') as f:
        return f.read()
print(read_file("nonexistent.txt"))
```

Code given by AI:

```python
def read_file(filename):
    try:
        with open(filename, 'r') as f:
            return f.read()
    except FileNotFoundError:
        return f"Error: The file '{filename}' was not found."
    except Exception as e:
        return f"An error occurred: {e}"
```

Execution:



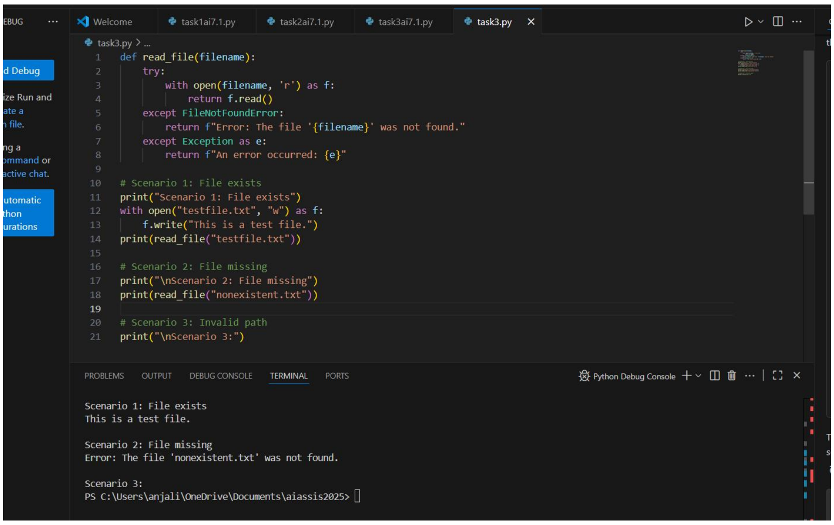
```python
def read_file(filename):
    try:
        with open(filename, 'r') as f:
            return f.read()
    except FileNotFoundError:
        return f"Error: The file '{filename}' was not found."
    except Exception as e:
        return f"An error occurred: {e}"
print(read_file("nonexistent.txt"))
```

```
Error: The file 'nonexistent.txt' was not found.
PS C:\Users\anjali\OneDrive\Documents\aiassis2025> 
```

Output:

Error: The file 'nonexistent.txt' was not found.

Scenarios:



```python
def read_file(filename):
    try:
        with open(filename, 'r') as f:
            return f.read()
    except FileNotFoundError:
        return f"Error: The file '{filename}' was not found."
    except Exception as e:
        return f"An error occurred: {e}"

# Scenario 1: File exists
print("Scenario 1: File exists")
with open("testfile.txt", "w") as f:
    f.write("This is a test file.")
print(read_file("testfile.txt"))

# Scenario 2: File missing
print("\nScenario 2: File missing")
print(read_file("nonexistent.txt"))

# Scenario 3: Invalid path
print("\nScenario 3:")
```

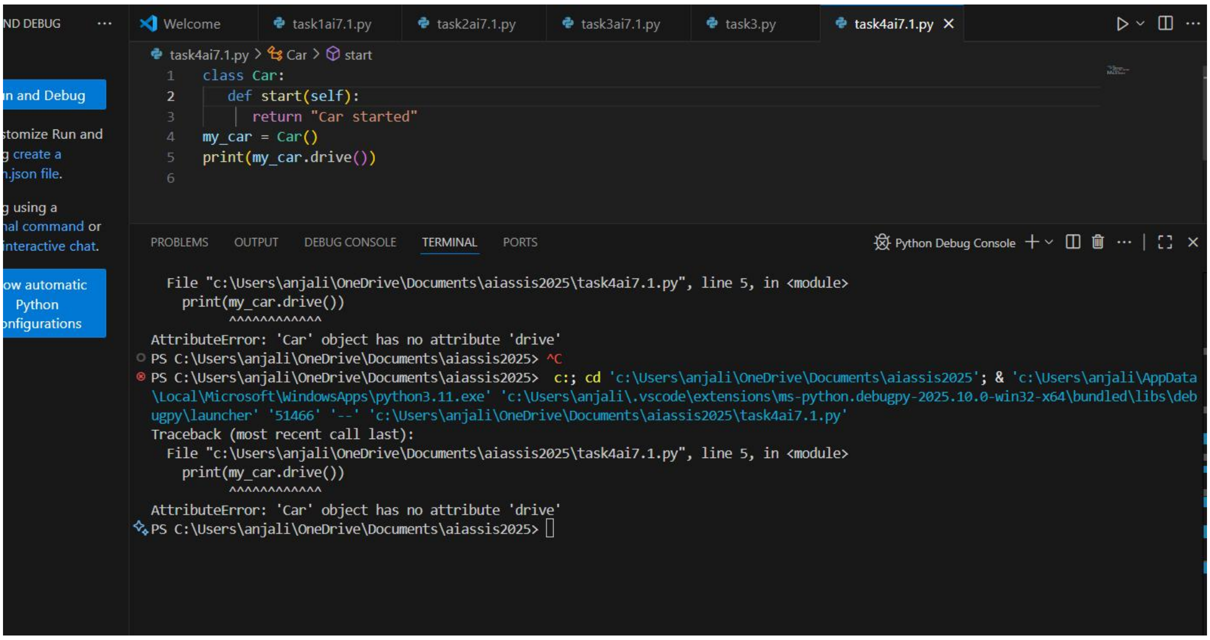
```
Scenario 1: File exists
This is a test file.

Scenario 2: File missing
Error: The file 'nonexistent.txt' was not found.

Scenario 3:
PS C:\Users\anjali\OneDrive\Documents\aiassis2025> 
```

TASK-4:

AttributeError – Calling a Non-Existent Method

Running given code:

```python
class Car:

    def start(self):
        return "Car started"

my_car = Car()
print(my_car.drive())
```



```python
class Car:
    def start(self):
        return "Car started"
my_car = Car()
print(my_car.drive())
```

```
  File "c:\Users\anjali\OneDrive\Documents\aiassis2025\task4ai7.1.py", line 5, in <module>
    print(my_car.drive())
          ^^^^^^^^^^^^
AttributeError: 'Car' object has no attribute 'drive'
PS C:\Users\anjali\OneDrive\Documents\aiassis2025> ^C
PS C:\Users\anjali\OneDrive\Documents\aiassis2025>  c:; cd 'c:\Users\anjali\OneDrive\Documents\aiassis2025'; & 'c:\Users\anjali\AppData
\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\anjali\.vscode\extensions\ms-python.debugpy-2025.10.0-win32-x64\bundled\libs\deb
ugpy\launcher' '51466' '--' 'c:\Users\anjali\OneDrive\Documents\aiassis2025\task4ai7.1.py'
Traceback (most recent call last):
  File "c:\Users\anjali\OneDrive\Documents\aiassis2025\task4ai7.1.py", line 5, in <module>
    print(my_car.drive())
          ^^^^^^^^^^^^
AttributeError: 'Car' object has no attribute 'drive'
PS C:\Users\anjali\OneDrive\Documents\aiassis2025> 
```

Error:

AttributeError because the Car class does **not** have a drive method. You are trying to call my_car.drive, but only start is defined

Analyzation:

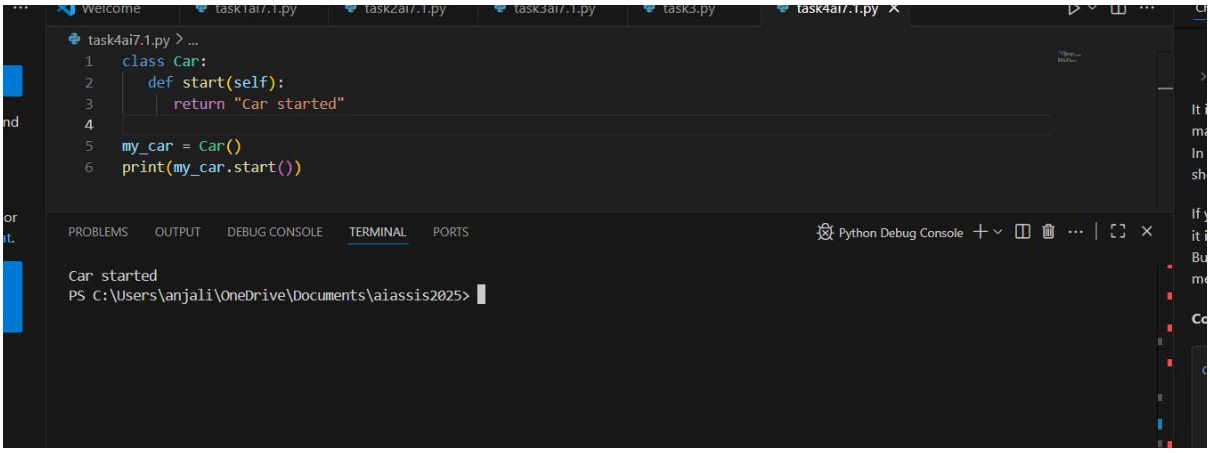
It is better to **correct the method call** to match the methods defined in your class.

Prompt used:

debug and fix.# Bug: Calling an undefined method
class Car:
def start(self):
return "Car started"
my_car = Car()
print(my_car.drive())

Corrected code:

```
        class Car:

  def start(self):

    return "Car started"

my_car = Car()

result =my_car.start()

print(result)
```



```python
class Car:
    def start(self):
        return "Car started"

my_car = Car()
print(my_car.start())
```

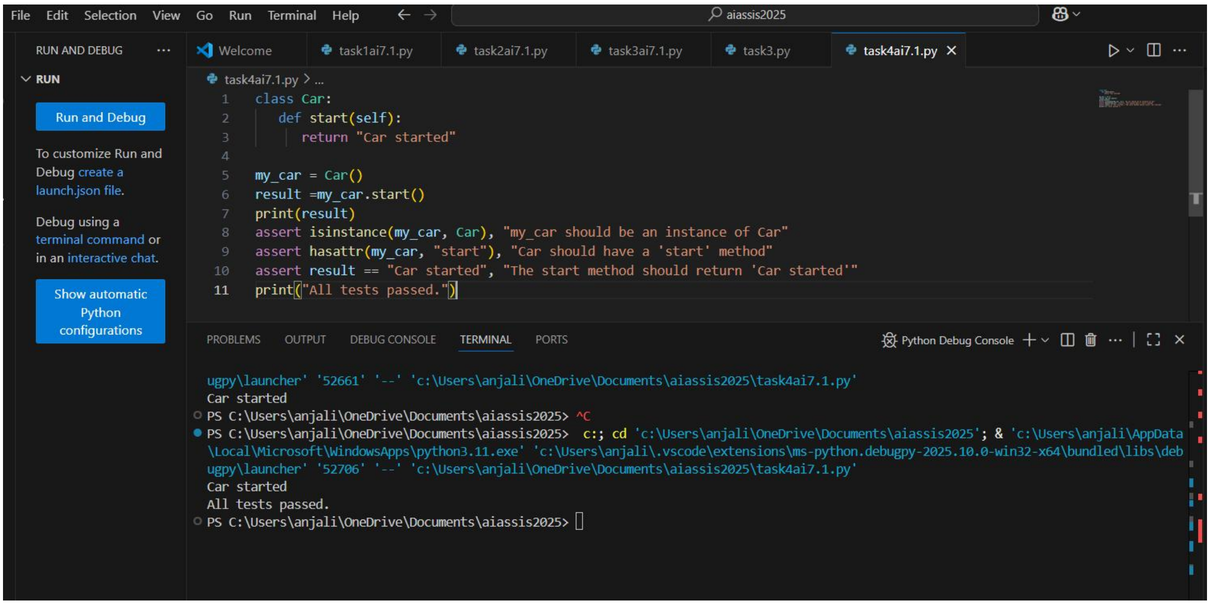
```
Car started
PS C:\Users\anjali\OneDrive\Documents\aiassis2025>
```

Output: Car started

Explanation:
This code defines a Car class with a start method that returns the string "Car started". An instance of Car is created and the start method is called, printing the result.

Assert tests:

```
class Car:
  def start(self):
    return "Car started"


my_car = Car()
result =my_car.start()
print(result)
assert isinstance(my_car, Car), "my_car should be an instance of Car"
assert hasattr(my_car, "start"), "Car should have a 'start' method"
assert result == "Car started", "The start method should return 'Car started'"
print("All tests passed.")
```


File Edit Selection View Go Run Terminal Help
aiassis2025
RUN AND DEBUG
RUN
Run and Debug
To customize Run and Debug create a launch.json file.
Debug using a terminal command or in an interactive chat.
Show automatic Python configurations
Welcome
task1ai7.1.py
task2ai7.1.py
task3ai7.1.py
task3.py
task4ai7.1.py
task4ai7.1.py > ...
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Python Debug Console

```python
class Car:
    def start(self):
        return "Car started"

my_car = Car()
result =my_car.start()
print(result)
assert isinstance(my_car, Car), "my_car should be an instance of Car"
assert hasattr(my_car, "start"), "Car should have a 'start' method"
assert result == "Car started", "The start method should return 'Car started'"
print("All tests passed.")
```

```
ugpy\launcher' '52661' '--' 'c:\Users\anjali\OneDrive\Documents\aiassis2025\task4ai7.1.py'
Car started
PS C:\Users\anjali\OneDrive\Documents\aiassis2025> ^C
PS C:\Users\anjali\OneDrive\Documents\aiassis2025>  c:; cd 'c:\Users\anjali\OneDrive\Documents\aiassis2025'; & 'c:\Users\anjali\AppData
\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\anjali\.vscode\extensions\ms-python.debugpy-2025.10.0-win32-x64\bundled\libs\deb
ugpy\launcher' '52706' '--' 'c:\Users\anjali\OneDrive\Documents\aiassis2025\task4ai7.1.py'
Car started
All tests passed.
PS C:\Users\anjali\OneDrive\Documents\aiassis2025> 
```

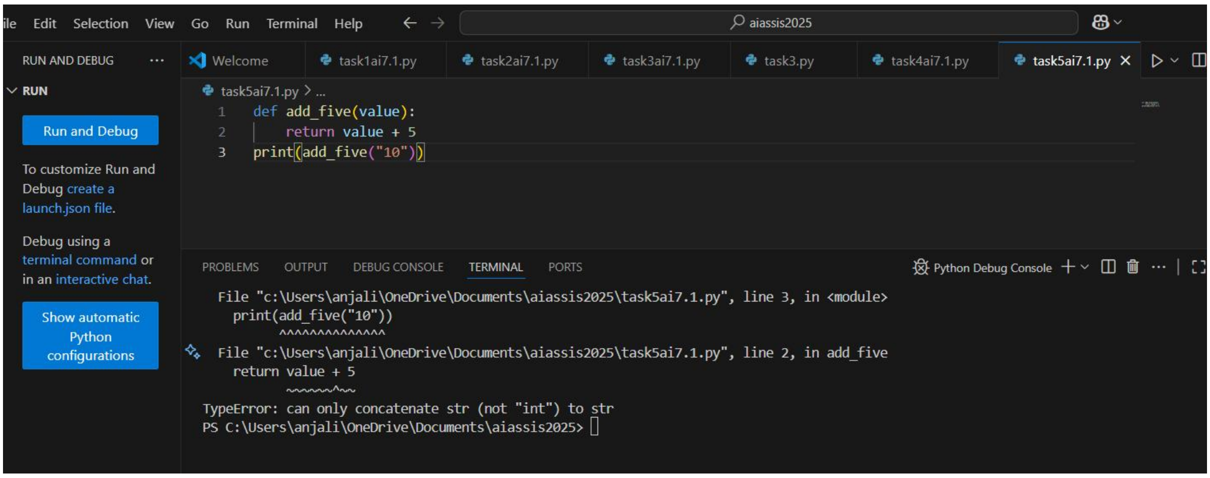
TASK-5

TypeError – Mixing Strings and Integers in Addition

Prompt used:

Resolve the given code and suggest the two correct codes using typecasting and string concatenation that takes multiple inputs in both cases.

```
def add_five(value):
return value + 5
print(add_five("10"))
```



```python
def add_five(value):
    return value + 5
print(add_five("10"))
```

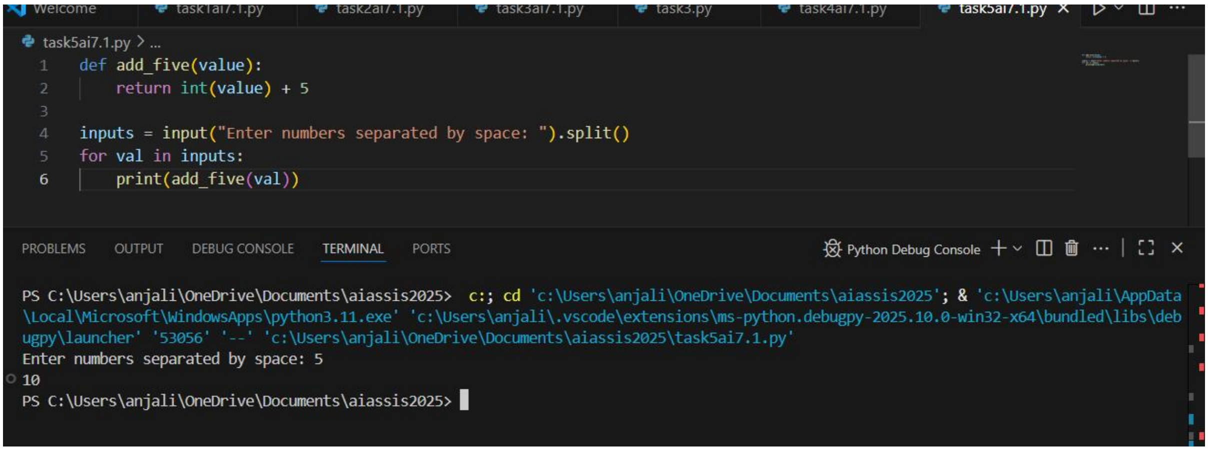
```
  File "c:\Users\anjali\OneDrive\Documents\aiassis2025\task5ai7.1.py", line 3, in <module>
    print(add_five("10"))
          ^^^^^^^^^^^^^^^
  File "c:\Users\anjali\OneDrive\Documents\aiassis2025\task5ai7.1.py", line 2, in add_five
    return value + 5
           ~~~~~~^~~
TypeError: can only concatenate str (not "int") to str
PS C:\Users\anjali\OneDrive\Documents\aiassis2025> 
```

Error:

TypeError: Trying to add an integer (5) to a string ("10")

Correction of code:

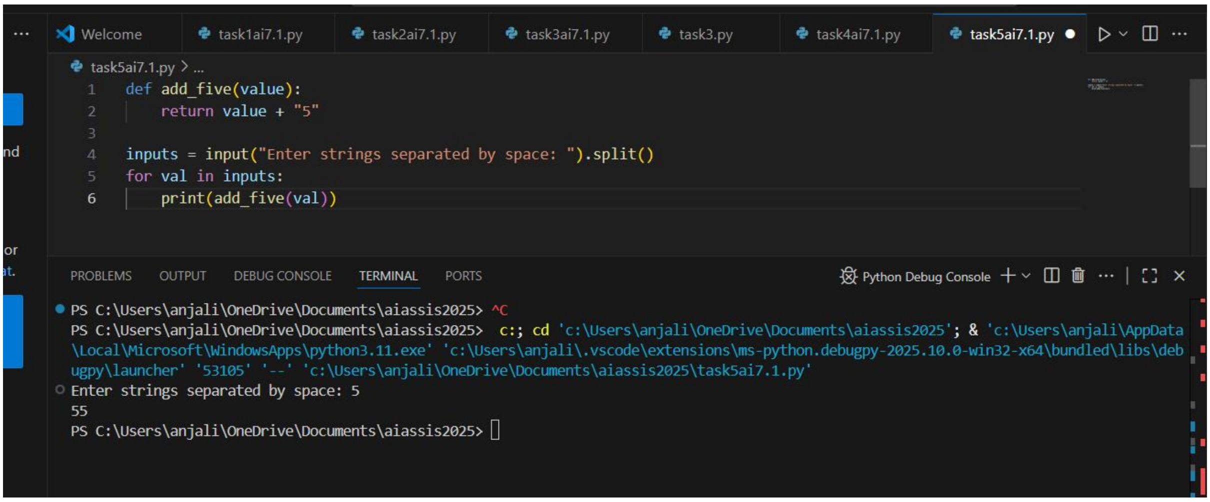
1.Using typecasting (convert input to int and add 5):



```python
def add_five(value):
    return int(value) + 5

inputs = input("Enter numbers separated by space: ").split()
for val in inputs:
    print(add_five(val))
```

```
PS C:\Users\anjali\OneDrive\Documents\aiassis2025>  c:; cd 'c:\Users\anjali\OneDrive\Documents\aiassis2025'; & 'c:\Users\anjali\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\anjali\.vscode\extensions\ms-python.debugpy-2025.10.0-win32-x64\bundled\libs\debugpy\launcher' '53056' '--' 'c:\Users\anjali\OneDrive\Documents\aiassis2025\task5ai7.1.py'
Enter numbers separated by space: 5
10
PS C:\Users\anjali\OneDrive\Documents\aiassis2025>
```

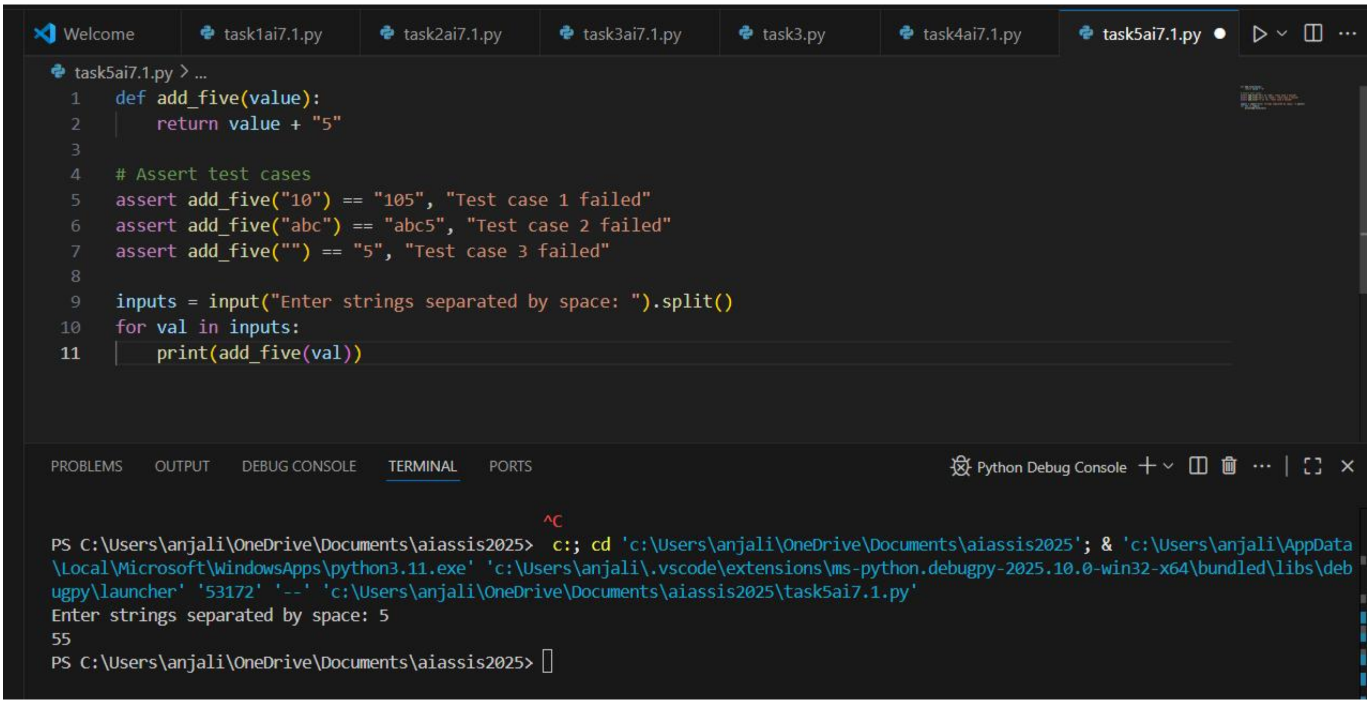
2.Using string concatenation (append "5" to each input string):



```python
def add_five(value):
    return value + "5"

inputs = input("Enter strings separated by space: ").split()
for val in inputs:
    print(add_five(val))
```

```
PS C:\Users\anjali\OneDrive\Documents\aiassis2025> ^C
PS C:\Users\anjali\OneDrive\Documents\aiassis2025>  c:; cd 'c:\Users\anjali\OneDrive\Documents\aiassis2025'; & 'c:\Users\anjali\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\anjali\.vscode\extensions\ms-python.debugpy-2025.10.0-win32-x64\bundled\libs\debugpy\launcher' '53105' '--' 'c:\Users\anjali\OneDrive\Documents\aiassis2025\task5ai7.1.py'
Enter strings separated by space: 5
55
PS C:\Users\anjali\OneDrive\Documents\aiassis2025>
```

Assert Cases:



```python
def add_five(value):
    return value + "5"

# Assert test cases
assert add_five("10") == "105", "Test case 1 failed"
assert add_five("abc") == "abc5", "Test case 2 failed"
assert add_five("") == "5", "Test case 3 failed"

inputs = input("Enter strings separated by space: ").split()
for val in inputs:
    print(add_five(val))
```

```
                                                 ^C
PS C:\Users\anjali\OneDrive\Documents\aiassis2025>  c:; cd 'c:\Users\anjali\OneDrive\Documents\aiassis2025'; & 'c:\Users\anjali\AppData
\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\anjali\.vscode\extensions\ms-python.debugpy-2025.10.0-win32-x64\bundled\libs\deb
ugpy\launcher' '53172' '--' 'c:\Users\anjali\OneDrive\Documents\aiassis2025\task5ai7.1.py'
Enter strings separated by space: 5
55
PS C:\Users\anjali\OneDrive\Documents\aiassis2025> 
```

AI explanation :

- The function add_five takes one argument, value.
- Inside the function, value is converted from a string to an integer using int(value).
- It then adds 5 to this integer and returns the result.
- print(add_five("10")) calls the function with the string "10", so it becomes 10 + 5, which is 15.