

AI ASSISTED CODING

LAB-7.5

Jadala Varshini

2303A51758

Batch-11

Task 1 (Mutable Default Argument – Function Bug)

Task: Analyze given code where a mutable default argument causes unexpected behavior. Use AI to fix it. # Bug: Mutable default

argument

def add_item(item, items=[]):

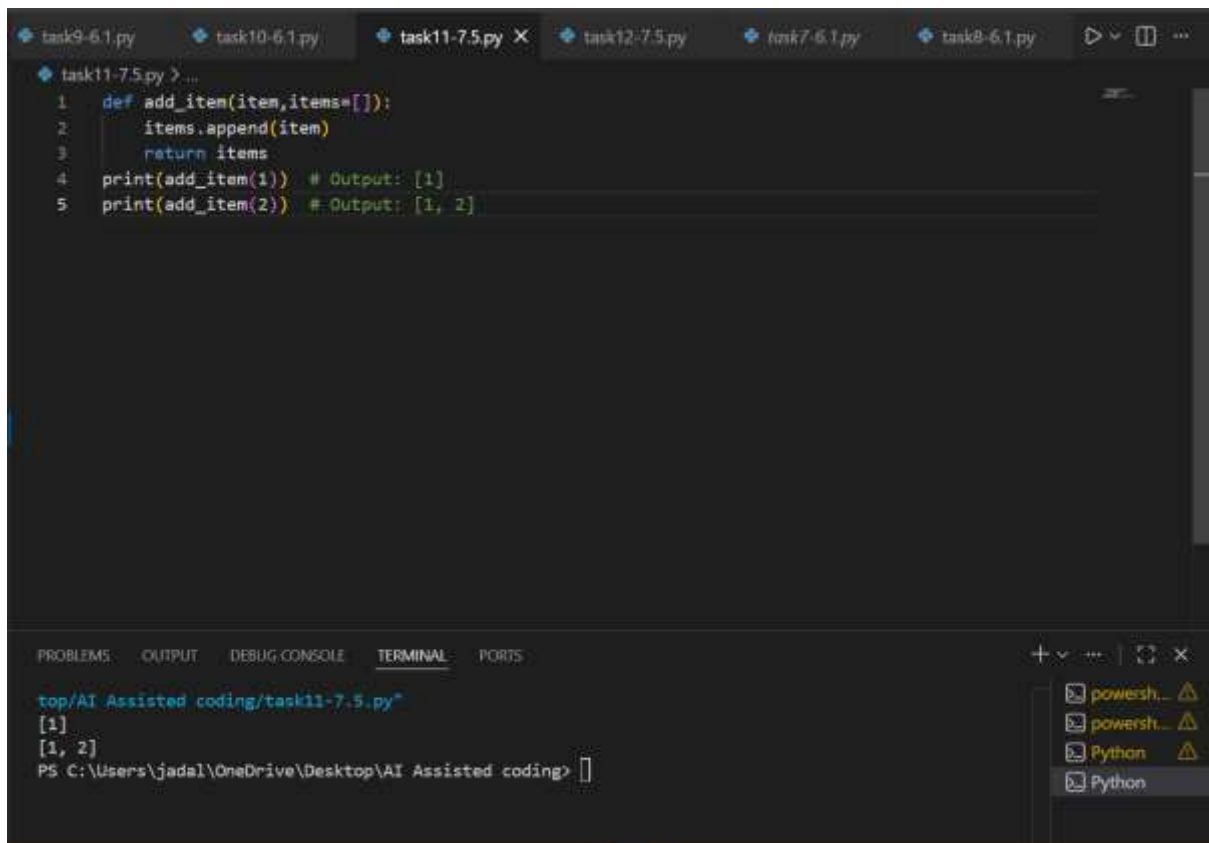
items.append(item) return

items print(add_item(1))

print(add_item(2))

Expected Output: Corrected function avoids shared list bug.

Given Code and Output:



```
task9-6.1.py task10-6.1.py task11-7.5.py X task12-7.5.py task7-6.1.py task8-6.1.py
task11-7.5.py > ...
1 def add_item(item, items=[]):
2     items.append(item)
3     return items
4 print(add_item(1)) # Output: [1]
5 print(add_item(2)) # Output: [1, 2]
```

```
top/AI Assisted coding/task11-7.5.py*
[1]
[1, 2]
PS C:\Users\jadal\OneDrive\Desktop\AI Assisted coding>
```

Task 2 (Floating-Point Precision Error)

Task: Analyze given code where floating-point comparison fails.

Use AI to correct with tolerance. #

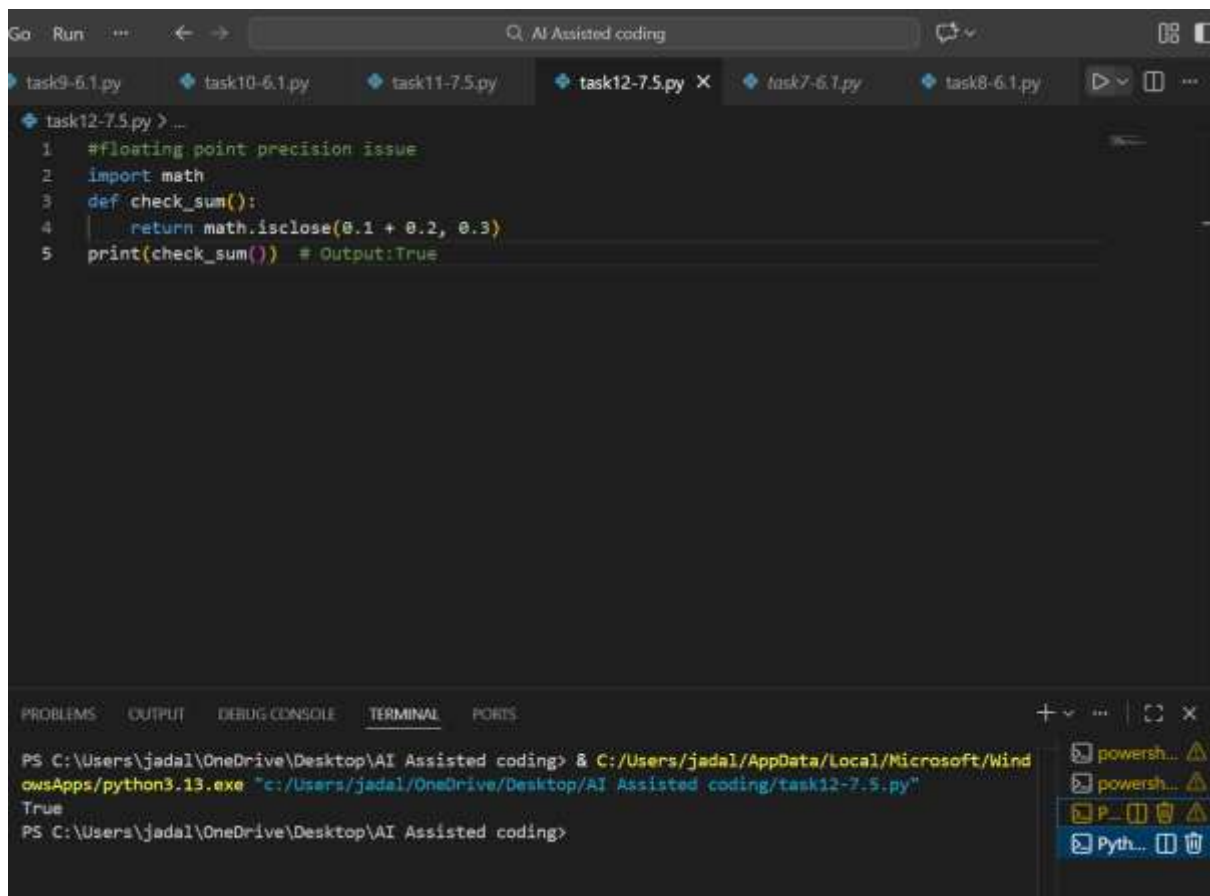
Bug: Floating point precision issue

```
def check_sum(): return (0.1 + 0.2)
```

```
== 0.3 print(check_sum())
```

Expected Output: Corrected function Given

Code and Output:



The screenshot shows a code editor with a dark theme. The top bar includes a search icon, the text 'AI Assisted coding', and a refresh icon. Below the top bar, there are several tabs for Python files: 'task9-6.1.py', 'task10-6.1.py', 'task11-7.5.py', 'task12-7.5.py' (which is active and has a close button), 'task7-6.1.py', and 'task8-6.1.py'. The active file 'task12-7.5.py' contains the following code:

```
1 #floating point precision issue
2 import math
3 def check_sum():
4     return math.isclose(0.1 + 0.2, 0.3)
5 print(check_sum()) # Output:True
```

At the bottom of the editor, there is a panel with tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL', and 'PORTS'. The 'TERMINAL' tab is selected, showing the command prompt output:

```
PS C:\Users\jadal\OneDrive\Desktop\AI Assisted coding> & C:/Users/jadal/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/jadal/OneDrive/Desktop/AI Assisted coding/task12-7.5.py"
True
PS C:\Users\jadal\OneDrive\Desktop\AI Assisted coding>
```

On the right side of the terminal panel, there are several icons for external tools, including 'powershell...', 'P...', and 'Pyth...'.

Task 3: (Recursion Error – Missing Base Case)

Task: Analyze given code where recursion runs infinitely due to missing base case. Use AI to fix.

Bug: No base case def

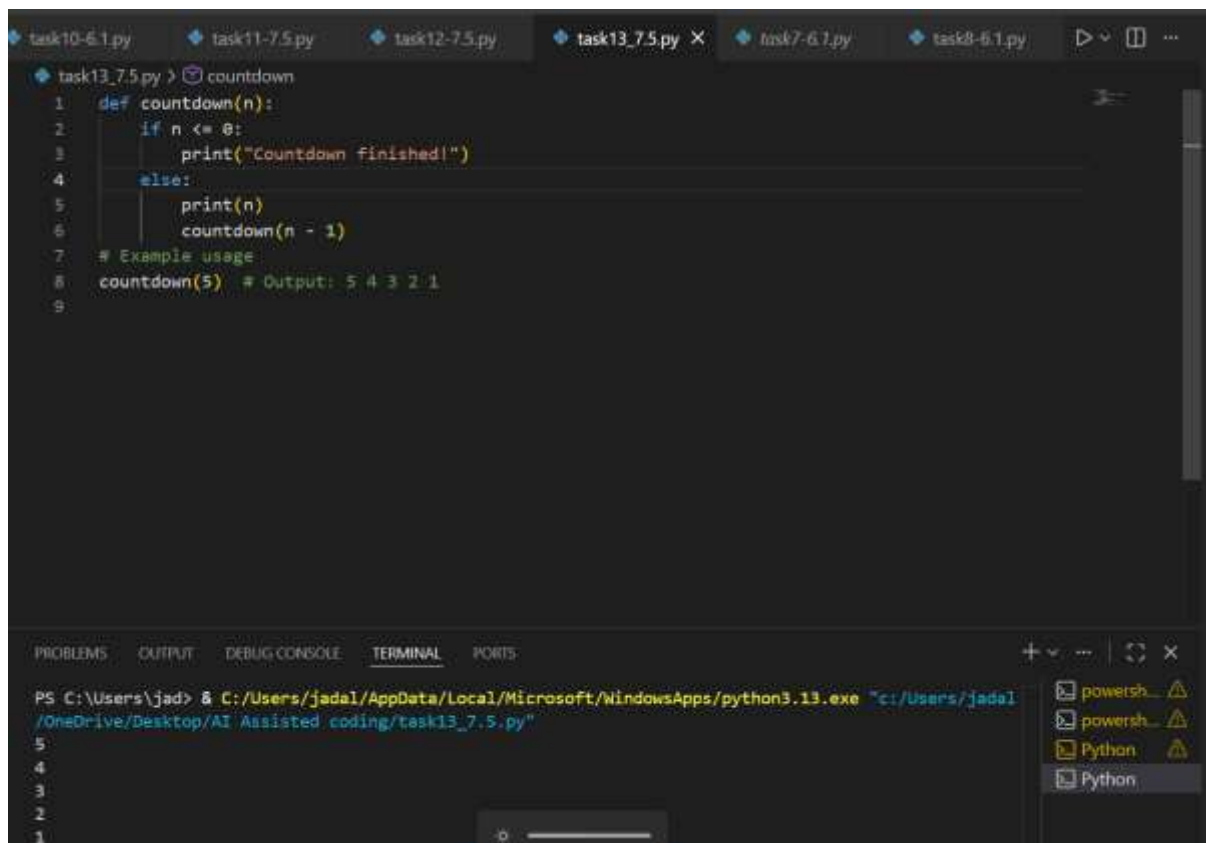
```
countdown(n):
```

```
print(n)
```

```
return countdown(n-1) countdown(5)
```

Expected Output : Correct recursion with stopping condition.

Given Code and Output:



```
task10-6.1.py task11-7.5.py task12-7.5.py task13_7.5.py X task7-6.1.py task8-6.1.py
task13_7.5.py > countdown
1 def countdown(n):
2     if n <= 0:
3         print("Countdown finished!")
4     else:
5         print(n)
6         countdown(n - 1)
7 # Example usage
8 countdown(5) # Output: 5 4 3 2 1
9

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\jad> & C:/Users/jadal/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/jadal/OneDrive/Desktop/AI Assisted coding/task13_7.5.py"
5
4
3
2
1
```

Task 4 (Dictionary Key Error)

Task: Analyze given code where a missing dictionary key causes error. Use AI to fix it.

Bug: Accessing non-existing key

```
def get_value(): data = {"a": 1,
```

```
"b": 2} return data["c"]
```

```
print(get_value())
```

Expected Output: Corrected with .get() or error handling.

Given Code and output:

```
Go Run ... < -> AI Assisted coding task11-7.5.py task12-7.5.py task13_7.5.py task14_7.5.py X task7-6.1.py task8-6.1.py
```

```
task14_7.5.py > ...
1 def get_value():
2     data={"a":1,"b":2}
3     return data.get("c","key not found")
4 print(get_value()) # Output: key not found
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
jadal/OneDrive/Desktop/AI Assisted coding/task14_7.5.py
key not found
PS C:\Users\jadal\OneDrive\Desktop\AI Assisted coding>
```

powerh... ⚠
powerh... ⚠
Python ⚠
Python

Task 5 (Infinite Loop – Wrong Condition)

Task: Analyze given code where loop never ends. Use AI to detect

and fix it. # Bug:

Infinite loop def

loop_example():

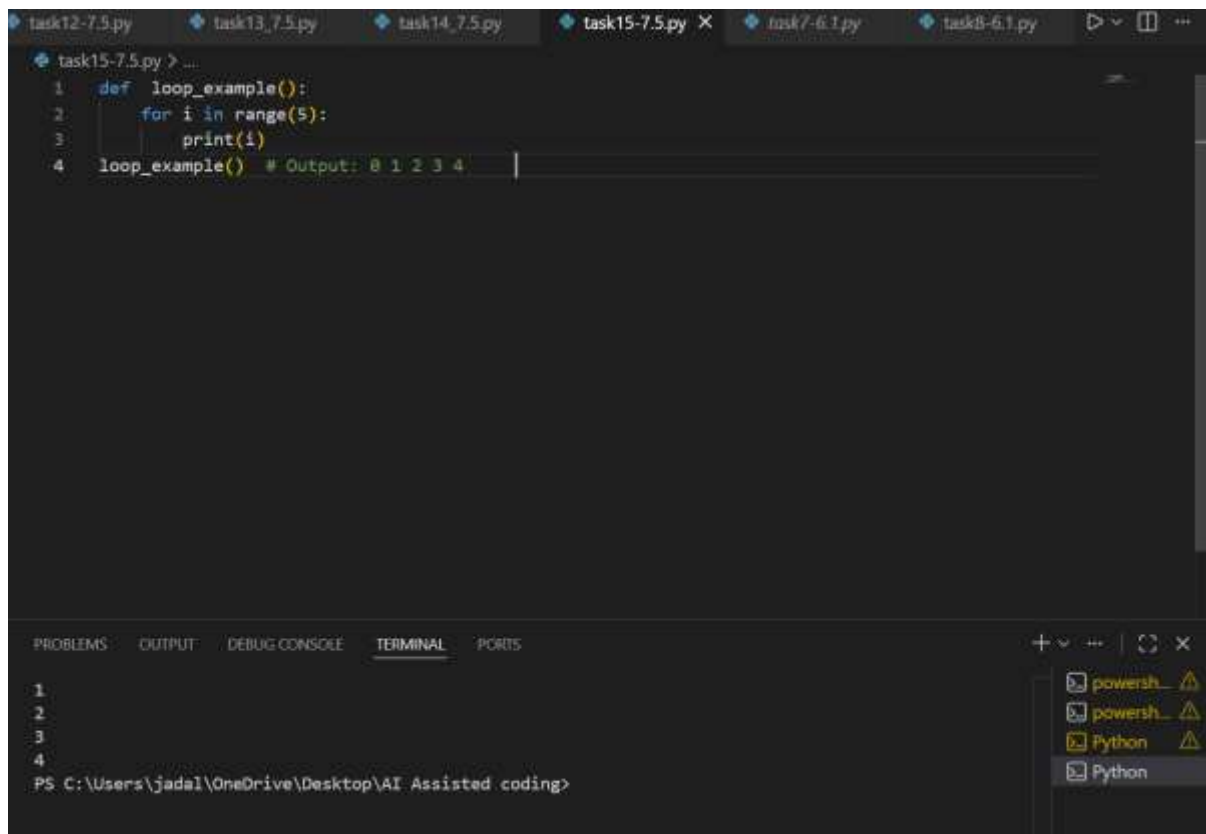
i = 0 while

i < 5:

print(i)

Expected Output: Corrected loop increments i.

Given Code and Output:



The screenshot shows a code editor with several tabs at the top: task12-7.5.py, task13-7.5.py, task14-7.5.py, task15-7.5.py (active), task7-6.1.py, and task8-6.1.py. The active tab displays a Python function named `loop_example()` that uses a `for` loop to print numbers from 0 to 4. Below the function definition, the function is called, and the output is shown as `# Output: 0 1 2 3 4`. The bottom panel of the editor shows the `TERMINAL` tab with a command prompt at `PS C:\Users\jadal\OneDrive\Desktop\AI Assisted coding>`. On the right side of the bottom panel, there is a list of installed extensions: `powershell`, `powershell`, `Python`, and `Python`.

```
1 def loop_example():
2     for i in range(5):
3         print(i)
4 loop_example() # Output: 0 1 2 3 4
```

Task 6 (Unpacking Error – Wrong Variables)

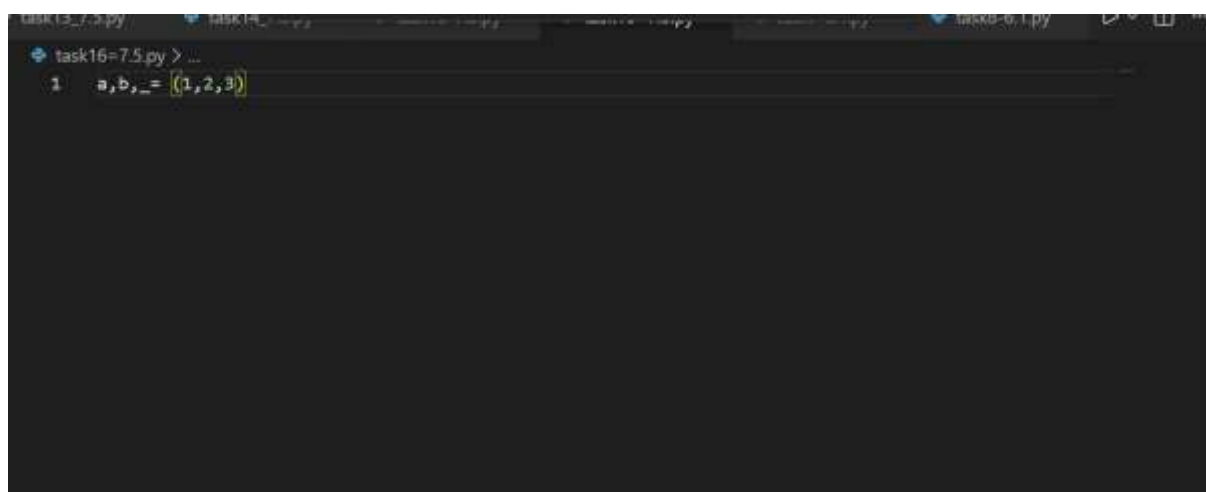
Task: Analyze given code where tuple unpacking fails. Use AI to fix it.

Bug: Wrong unpacking

a, b = (1, 2, 3)

Expected Output: Correct unpacking or using `_` for extra values.

Given Code:



The screenshot shows a code editor with several tabs at the top: task13-7.5.py, task14-7.5.py, task15-7.5.py, task16-7.5.py (active), task7-6.1.py, and task8-6.1.py. The active tab displays a single line of Python code: `a, b, _ = (1, 2, 3)`. The code is highlighted in yellow.

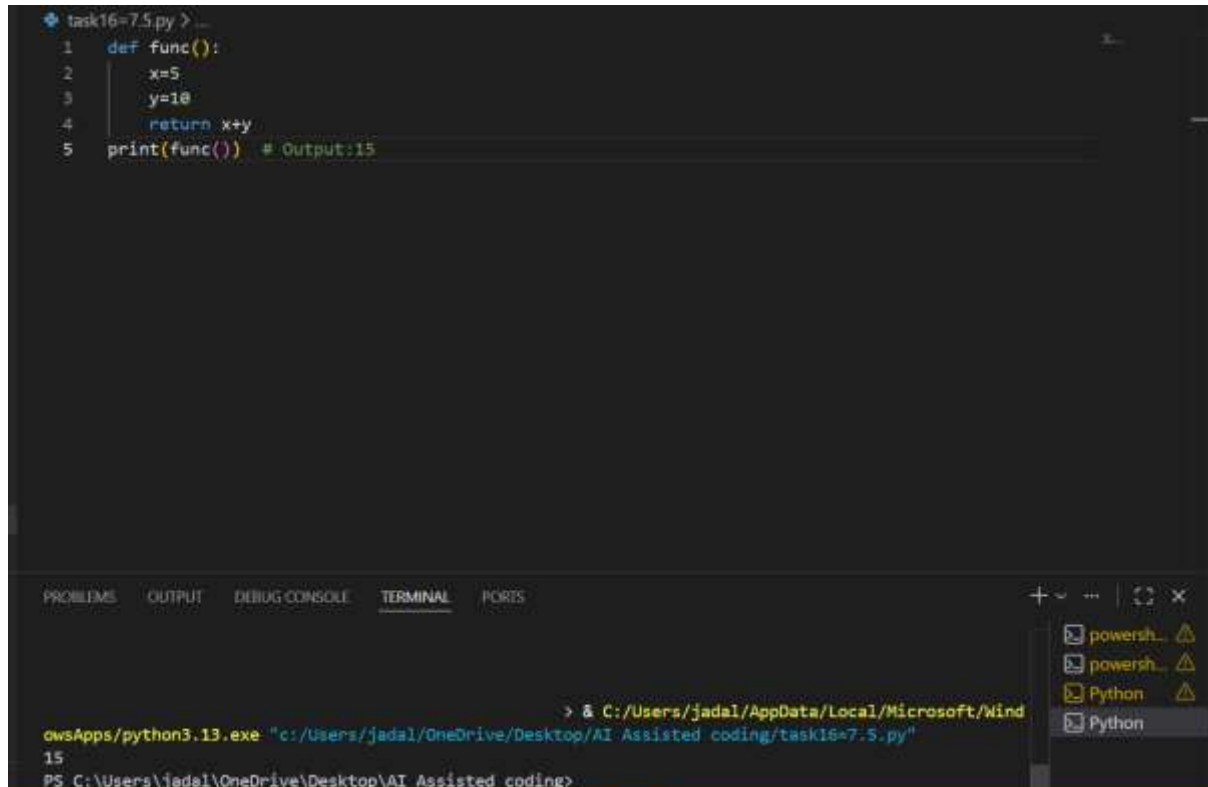
```
1 a, b, _ = (1, 2, 3)
```

Task 7 (Mixed Indentation – Tabs vs Spaces)

Task: Analyze given code where mixed indentation breaks execution. Use AI to fix it. # Bug: Mixed indentation def func(): x = 5 y = 10 return x+y print(func()) # Output:15

Expected Output : Consistent indentation applied.

Given Code and Output:



The screenshot shows a code editor with a Python file named 'task16=7.5.py'. The code is as follows:

```
1 def func():
2     x=5
3     y=10
4     return x+y
5 print(func()) # Output:15
```

The code is executed, and the output is shown in the terminal:

```
> & C:/Users/jadal/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/jadal/OneDrive/Desktop/AI Assisted coding/task16=7.5.py"
15
PS C:\Users\jadal\OneDrive\Desktop\AI Assisted coding>
```

Task 8 (Import Error – Wrong Module Usage)

Task: Analyze given code with incorrect import. Use AI to fix.

Bug: Wrong import

```
import      maths
print(maths.sqrt(16))
```

Expected Output: Corrected to import math Given

Code and Output:

```
task17-7.5.py
1 import math
2 print(math.sqrt(16)) # Output: 4.0
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

> & C:/Users/jadal/AppData/Local/Microsoft/WindowsApps/python3.13.exe ~c:/Users/jadal/OneDrive/Desktop/AI Assisted coding/task17-7.5.py
4.0
PS C:\Users\jadal\OneDrive\Desktop\AI Assisted coding>

powerh...
powerh...
Python
Python