

# Lab Assignment-7.5

Name: N.Paul Benjamin

Hallticket:2303A51116

Batch:03

## Task 1 (Mutable Default Argument – Function Bug)

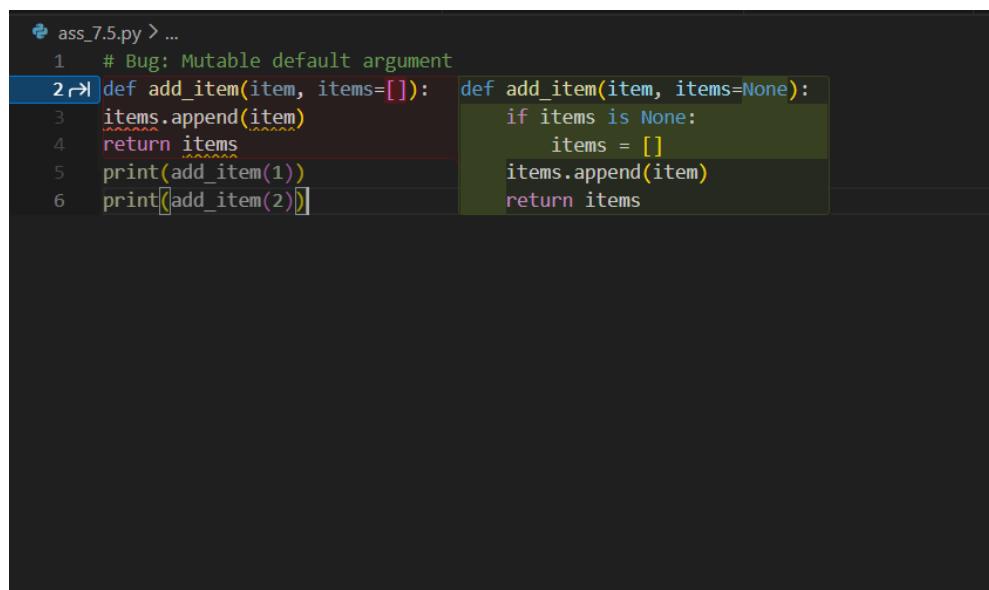
Task: Analyze given code where a mutable default argument causes unexpected behavior. Use AI to fix it.

```
# 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.

Screenshots:



```
ass_7.5.py > ...  
1 # Bug: Mutable default argument  
2 def add_item(item, items=[]):  
3     items.append(item)  
4     return items  
5 print(add_item(1))  
6 print(add_item(2))
```

The screenshot shows a terminal window with Python code. A green box highlights the line 'items.append(item)' in the original buggy code. A corrected version of the code is shown in a green box on the right, where the mutable list is replaced by a new list '[]' each time the function is called.

assg\_07.py > ...

```
1  def add_item(item, items=None):
2      if items is None:
3          items = []
4      items.append(item)
5      return items
6  print(add_item(1))
7  print(add_item(2))
```

output:

The screenshot shows the VS Code interface with the terminal tab active. The terminal window displays the command PS C:\Users\91938\OneDrive\Documents\Desktop\AI> & 'c:\Users\91938\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\91938\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '54845' '--' 'c:\Users\91938\OneDrive\Documents\Desktop\AI\AIAC7\_5.py'. It also shows two numbered brackets [1] and [2] indicating specific lines of output.

### 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

Screenshots:

The screenshot shows the VS Code interface with an AI suggestion overlay. The suggestion 'Fix the attached problems floating point pricision issue' is displayed at the top. Below it, there are three buttons: 'Add to chat (Alt+X)', 'Do not show again', 'Expected indented block', 'Code is structurally...', and a dropdown menu set to 'Auto'. The code editor shows the following code:

```
8  Fix the attached problems floating point pricision issue
9  def check_sum():
10     return (0.1 + 0.2) == 0.3
11     print(check_sum())
```

```
import math

def check_sum():
    return math.isclose(0.1 + 0.2, 0.3)
print(check_sum())
```

output:

```
PS C:\Users\91938\OneDrive\Documents\Desktop\AI> c:; cd 'c:\Users\91938\OneDrive\Documents\Desktop\AI'; & 'c:\Users\91938\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\91938\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '60525' '--' 'c:\Users\91938\OneDrive\Documents\Desktop\AI\AIACT_5.py'
True
PS C:\Users\91938\OneDrive\Documents\Desktop\AI>
```

### 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.

Screenshots:

```
ass_7.5.py > ...
3     print(n)
4     if n == 0:
5         return
4     return countdown(n-1)
5     countdown(5)
```

```
14
15     def countdown(n):
16         if n <= 0:
17             return
18         print(n)
19         return countdown(n-1)
20 countdown(5)
```

output:

```
PS C:\Users\91938\OneDrive\Documents\Desktop\AI> & 'c:\Users\91938\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\91938\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '57677' '--' 'c:\Users\91938\OneDrive\Documents\Desktop\AI\AIAC7_5.py'
5
4
3
2
1
PS C:\Users\91938\OneDrive\Documents\Desktop\AI>
```

## **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():
```

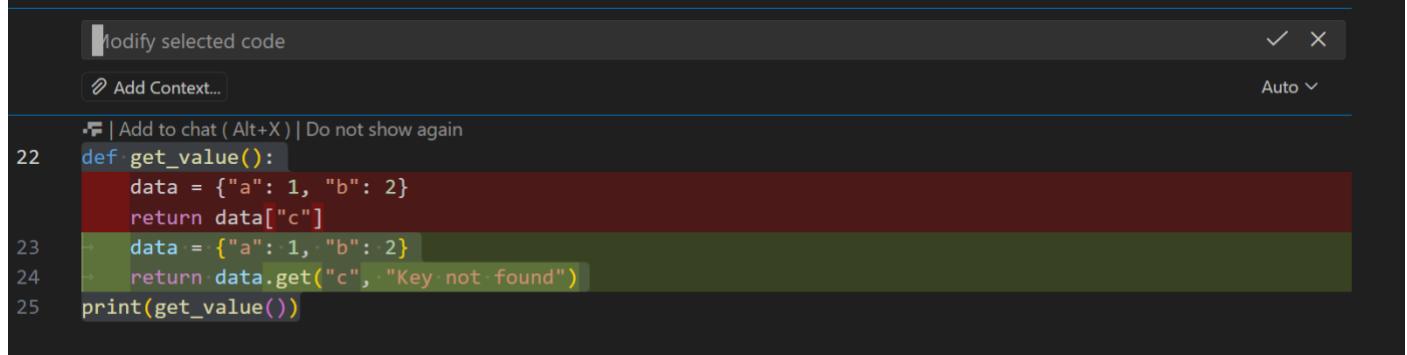
```
data = {"a": 1, "b": 2}
```

```
return data["c"]
```

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

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

## Screenshots:



```
21
22     def get_value():
23         data = {"a": 1, "b": 2}
24         return data.get("c", "Key not found")
25     print(get_value())
```

output:

```
1
● PS C:\Users\91938\OneDrive\Documents\Desktop\AI> c:; cd 'c:\Users\91938\OneDrive\Documents\Desktop\AI'; & 'c:\Users\91938\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\91938\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '52379' '--' 'c:\Users\91938\OneDrive\Documents\Desktop\AI\AIAC7_5.py'
Key not found
○ PS C:\Users\91938\OneDrive\Documents\Desktop\AI>
```

### 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.

Screenshots:

```
27
28     def loop_example():
29         i = 0
30         while i < 5:
31             print(i)
→|             i += 1
```

```
28
29     def loop_example():
30         i = 0
31         while i < 5:
32             print(i)
33             i += 1
34     loop_example()
```

output:

```
● PS C:\Users\91938\OneDrive\Documents\Desktop\AI> c;; cd 'c:\Users\91938\OneDrive\Documents\Desktop\AI'; & 'c:\Users\91938\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\91938\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '58963' '--' 'c:\Users\91938\OneDrive\Documents\Desktop\AI\AIAC7_5.py'
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.

Screenshots:

```
36
37     a, b, _ = (1, 2, 3)
38     print(a, b)
```

output:

```
● PS C:\Users\91938\OneDrive\Documents\Desktop\AI> c;; cd 'c:\Users\91938\OneDrive\Documents\Desktop\AI'; & 'c:\Users\91938\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\91938\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '55028' '--' 'c:\Users\91938\OneDrive\Documents\Desktop\AI\AIAC7_5.py'
1 2
○ PS C:\Users\91938\OneDrive\Documents\Desktop\AI>
```

## 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():
```

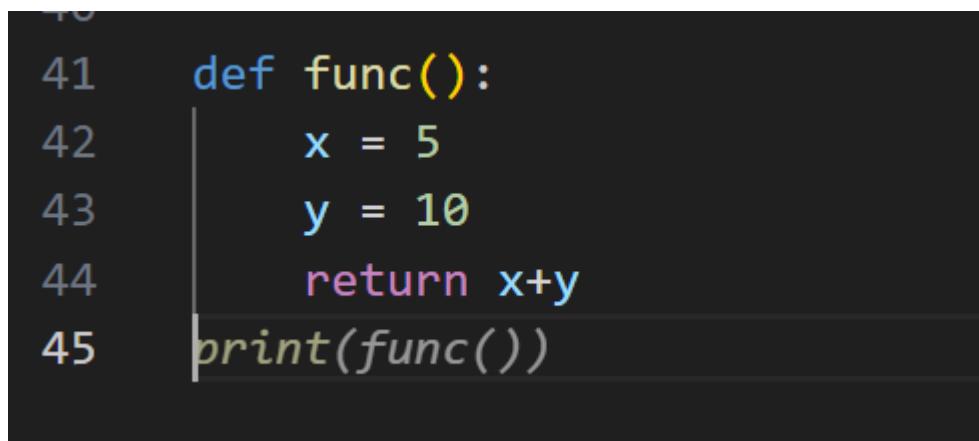
```
    x = 5
```

```
    y = 10
```

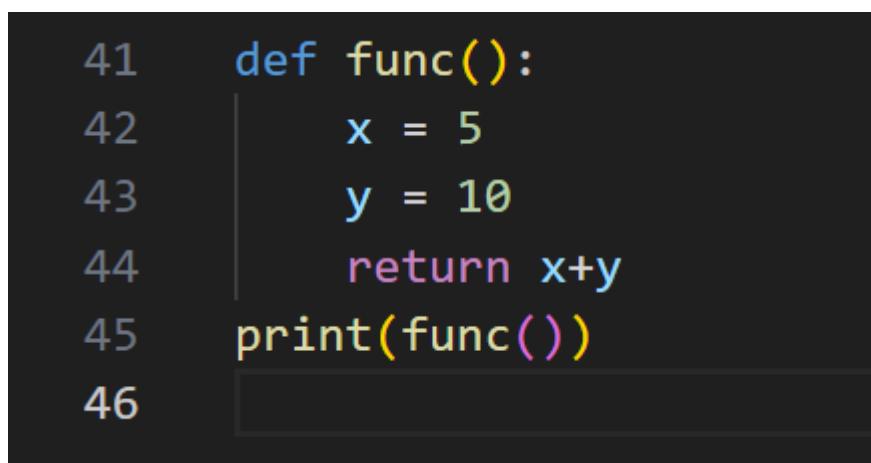
```
    return x+y
```

Expected Output : Consistent indentation applied.

Screenshots:



```
41     def func():
42         x = 5
43         y = 10
44         return x+y
45     print(func())
```



```
41     def func():
42         x = 5
43         y = 10
44         return x+y
45     print(func())
46
```

output:

```
● PS C:\Users\91938\OneDrive\Documents\Desktop\AI> & 'c:\Users\91938\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\91938\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '53419' '--' 'c:\Users\91938\OneDrive\Documents\Desktop\AI\AIAC7_5.py'
15
○ PS C:\Users\91938\OneDrive\Documents\Desktop\AI>
```

### Task 8 (Import Error – Wrong Module Usage)

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

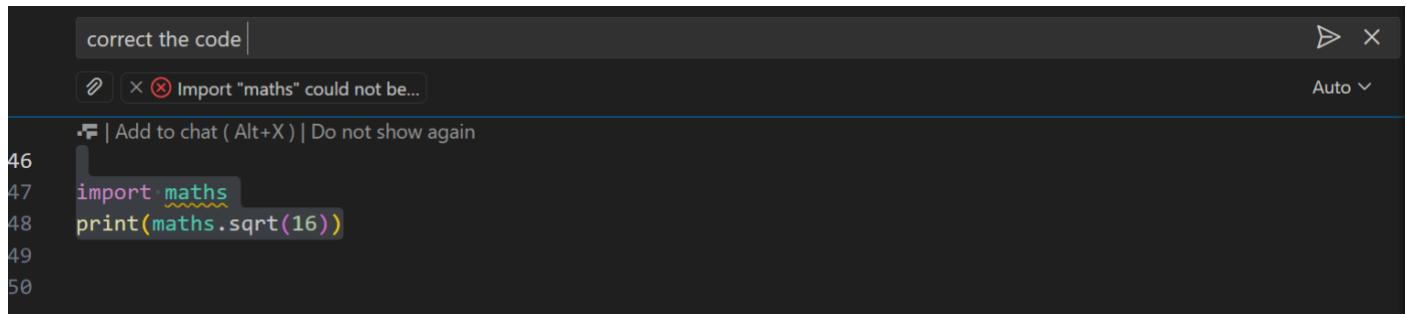
# Bug: Wrong import

```
import maths
```

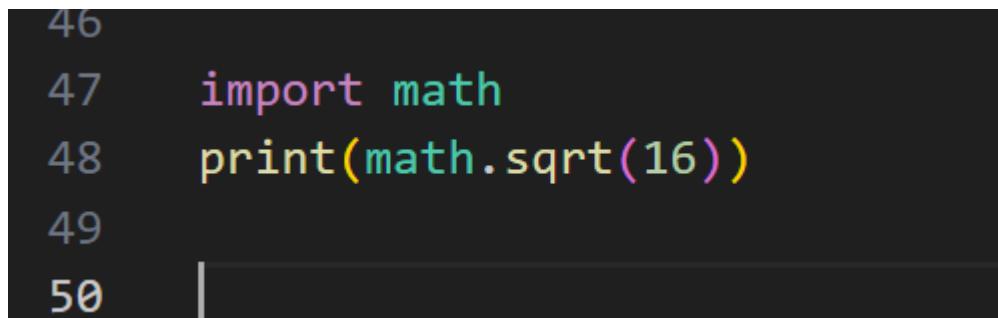
```
print(maths.sqrt(16))
```

Expected Output: Corrected to import math

Screenshots:



A screenshot of the Visual Studio Code interface. The title bar says "correct the code". In the top right, there are icons for file, edit, and close. Below the title bar, there's a toolbar with a search icon, a refresh icon, and a dropdown labeled "Auto". A message box displays an error: "Import 'maths' could not be..." with a link to "Add to chat (Alt+X)" and a "Do not show again" option. The main editor area shows code with line numbers 46 through 50. Line 47 contains the incorrect import statement "import maths". Line 48 shows the usage of "maths.sqrt(16)".



A screenshot of a terminal window. The code has been corrected to "import math" and "print(math.sqrt(16))". The terminal shows the command "PS C:\Users\91938\OneDrive\Documents\Desktop\AI> c:; cd 'c:\Users\91938\OneDrive\Documents\Desktop\AI'; & 'c:\Users\91938\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\91938\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '53439' '--' 'c:\Users\91938\OneDrive\Documents\Desktop\AI\AIAC7\_5.py'" followed by the output "4.0".

output:

```
PS C:\Users\91938\OneDrive\Documents\Desktop\AI> c:; cd 'c:\Users\91938\OneDrive\Documents\Desktop\AI'; & 'c:\Users\91938\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\91938\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '53439' '--' 'c:\Users\91938\OneDrive\Documents\Desktop\AI\AIAC7_5.py'
4.0
PS C:\Users\91938\OneDrive\Documents\Desktop\AI>
```