Lab Assignment 7.4

Name : k.Nikshitha

Htno : 2303A51692

Batch : 24

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)

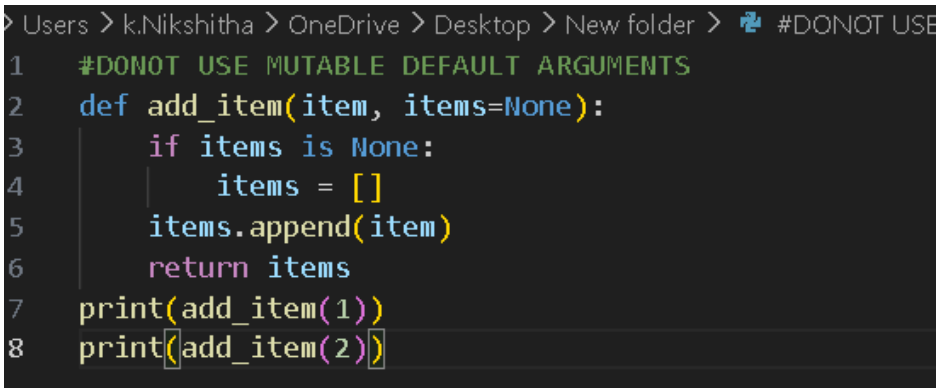
return items

print(add_item(1))

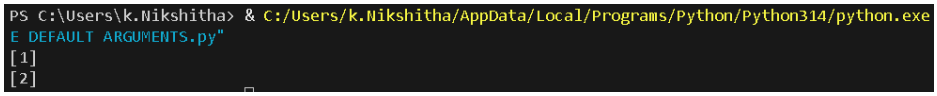
print(add_item(2))

Expected Output: Corrected function avoids shared list bug.

CODE :



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

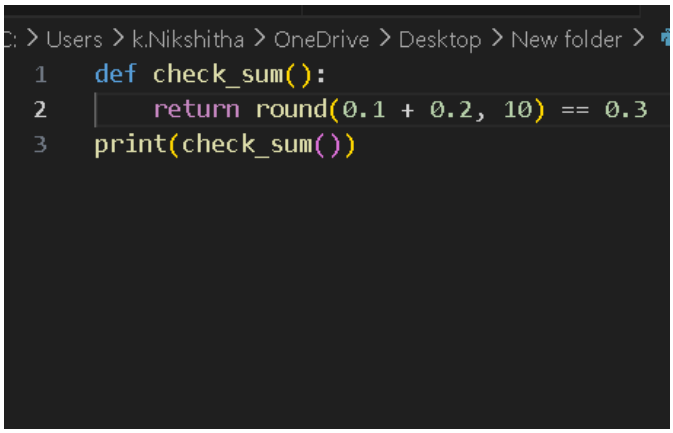
def check_sum():

return (0.1 + 0.2) == 0.3

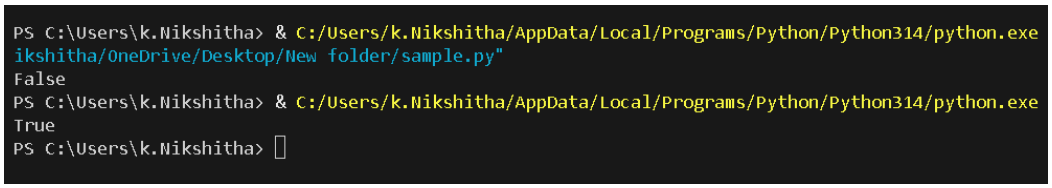
print(check_sum())

Expected Output: Corrected function

CODE :



OUTPUT :



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.

CODE :

```
def countdown(n):
print(n)
return countdown(n-1)
countdown(5)
```

OUTPUT :

```
IndentationError: expected an indented block after function definition on line 2
PS C:\Users\k.Nikshitha> & C:/Users/k.Nikshitha/AppData/Local/Programs/Python/Python314/python.ex
C:\Users\k.Nikshitha\AppData\Local\Programs\Python\Python314\python.exe: can't open file 'C:\\Use
PS C:\Users\k.Nikshitha> & C:/Users/k.Nikshitha/AppData/Local/Programs/Python/Python314/python.ex
  File "c:\Users\k.Nikshitha\OneDrive\Desktop\New folder\3_1.py", line 2
```

CORRECT CODE :

C: > Users > k.Nikshitha > OneDrive > Desktop > New folder > 3_1.py > ..

```
1  #fux the error in the below code
2  def countdown(n):
3      print(n)
4      if n == 0:
5          return
6      return countdown(n-1)
7  countdown(5)
```

OUTPUT :

```
IndentationError: expected an indented block after function definition on line 1
PS C:\Users\k.Nikshitha> & C:/Users/k.Nikshitha/AppData/Local/Programs/Python/Python314/python.exe "c
5
4
3
2
1
0
PS C:\Users\k.Nikshitha>
```

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.

CODE :

```
C: > Users > k.Nikshitha > OneDrive > Deskto
1   def get_value():
2   data = {"a": 1, "b": 2}
3   return data["c"]
4   print(get_value())
```

OUTPUT :

```
PS C:\Users\k.Nikshitha> & C:/Users/k.Nikshitha/AppData/Local/Programs/Python/Python314/python.exe
ror.py"
  File "c:\Users\k.Nikshitha\OneDrive\Desktop\New folder\dictionary key error.py", line 2
    data = {"a": 1, "b": 2}
    ^^^^
IndentationError: expected an indented block after function definition on line 1
```

CORRECT CODE :

```
C: > Users > k.Nikshitha > OneDrive > Desktop > New fol
1   def get_value():
2       data = {"a": 1, "b": 2}
3       return data.get("c", 0)
4   print(get_value())  # Output: 0
```

OUTPUT :

```
PS C:\Users\k.Nikshitha> & C:/Users/k.Nikshitha/AppData/Local/Programs/Python/Python314/python.exe
ror.py"
0
PS C:\Users\k.Nikshitha>
```

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.

CODE :

```
Users > k.Nikshitha > OneDrive > Deskt
def loop_example():
i = 0
while i < 5:
print(i)
```

OUTPUT :

```
PS C:\Users\k.Nikshitha> & C:/Users/k.Nikshitha/AppData/Local/Programs/Python/Python

  File "c:\Users\k.Nikshitha\OneDrive\Desktop\New folder\INFINITE_LOOP.py", line 2
    i = 0
    ^
IndentationError: expected an indented block after function definition on line 1
```

CORRECT CODE :

```
> Users > k.Nikshitha > OneDrive > Desktop > New folder >
1
2   #def loop_example():
3   i = 0
4   while i < 5:
5       print(i)
6       i += 1
7
```

OUTPUT :

```
PS C:\Users\k.Nikshitha> & C:/Users/k.Nikshitha/AppData/Local/Programs/Python/Python314/python.exe

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

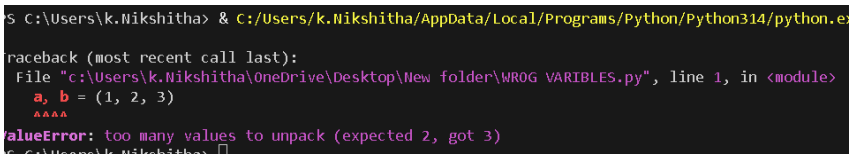
a, b = (1, 2, 3)

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

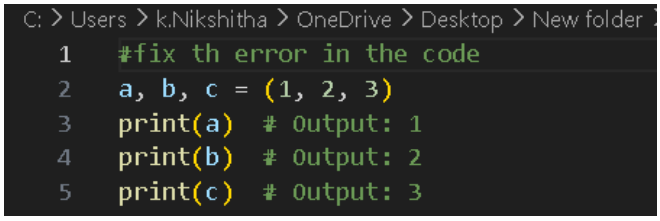
CODE :

```
C: > Users > k.Nikshitha > OneDrive > Desktop
1   a, b = (1, 2, 3)
```

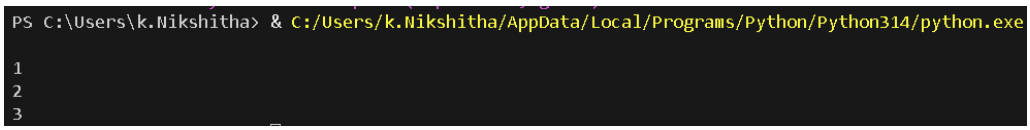
OUTPUT :



CORRECT CODE :



OUTPUT :



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():

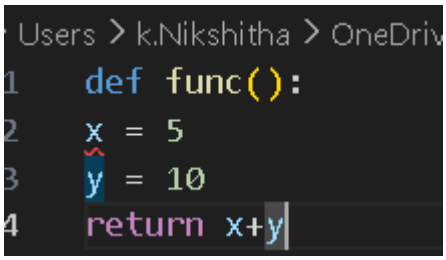
x = 5

y = 10

return x+y

Expected Output : Consistent indentation applied.

CODE :



OUTPUT :

```
PS C:\Users\k.Nikshitha> & C:/Users/k.Nikshitha/AppData/Local/Programs/Python/Pytho
  File "c:\Users\k.Nikshitha\OneDrive\Desktop\New folder\7_1.py", line 2
    x = 5
    ^
IndentationError: expected an indented block after function definition on line 1
```

CORRECT CODE :

```
C: > Users > k.Nikshitha > OneDrive > Desktop > N
1   def func():
2       x = 5
3       y = 10
4       return x + y
5   print(func())
```

OUTPUT :

```
PS C:\Users\k.Nikshitha> & C:/Users/k.Nikshitha/AppData/Local/Programs/Python/Python314/python.ex
15
```

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

CODE :

```
C: > Users > k.Nikshitha > OneDrive > Desktop > New folder > 
1   import maths
2   print(maths.sqrt(16))
```

OUTPUT :

```
PS C:\Users\k.Nikshitha> & C:/Users/k.Nikshitha/AppData/Local/Programs/Python/Python314/python.exe
Traceback (most recent call last):
  File "c:\Users\k.Nikshitha\OneDrive\Desktop\New folder\import maths.py", line 1, in <module>
    import maths
ModuleNotFoundError: No module named 'maths'
```

CORRECT CODE :

```
C: > Users > k.Nikshitha > OneDrive > Desktop > New folder > 
1   import math
2   print(math.sqrt(16))
```

OUTPUT :

```
PS C:\Users\k.Nikshitha> & C:/Users/k.Nikshitha/AppData/Local/Programs/Python/Python314/python.exe
4.0
```

Task 9 (Unreachable Code – Return Inside Loop)

Task: Analyze given code where a return inside a loop prevents full

iteration. Use AI to fix it.

Bug: Early return inside loop

def total(numbers):

for n in numbers:

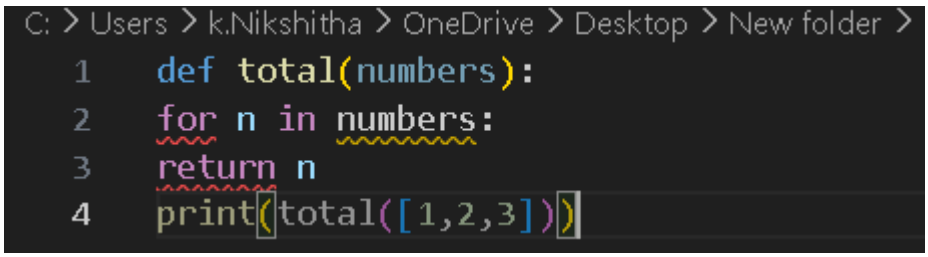
return n

print(total([1,2,3]))

Expected Output: Corrected code accumulates sum and returns

after loop.

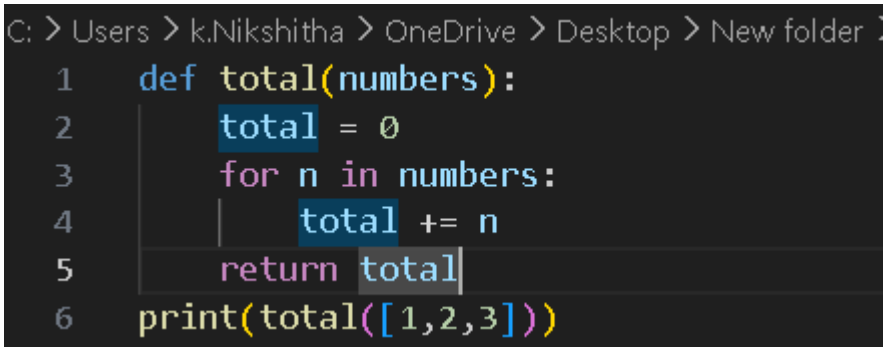
CODE :



OUTPUT :

```
PS C:\Users\k.Nikshitha> & C:/Users/k.Nikshitha/AppData/Local/Programs/Python/Python314/python.exe "c:/Use
  File "c:\Users\k.Nikshitha\OneDrive\Desktop\New folder\9_1.py", line 2
    for n in numbers:
    ^^^
IndentationError: expected an indented block after function definition on line 1
```

CORRECT CODE :



OUTPUT :

```
PS C:\Users\k.Nikshitha> & C:/Users/k.Nikshitha/AppData/Local/Programs/Python/Python314/python.exe
6
```

Task 10 (Name Error – Undefined Variable)

Task: Analyze given code where a variable is used before being

defined. Let AI detect and fix the error.

Bug: Using undefined variable

def calculate_area():

return length * width

print(calculate_area())

Requirements:

• Run the code to observe the error.

• Ask AI to identify the missing variable definition.

• Fix the bug by defining length and width as parameters.

• Add 3 assert test cases for correctness.

Expected Output :

• Corrected code with parameters.

• AI explanation of the bug.

Successful execution of assertions.

CODE :

```
C: > Users > k.Nikshitha > OneDrive > Desktop > New folder
1   def calculate_area():
2   return length * width
3   print(calculate_area())
```

OUTPUT:

```
PS C:\Users\k.Nikshitha> & C:/Users/k.Nikshitha/AppData/Local/Programs/Python/Python314/python.exe
  File "c:\Users\k.Nikshitha\OneDrive\Desktop\New folder\NAME ERROR.py", line 2
    return length * width
    ^^^^^^
IndentationError: expected an indented block after function definition on line 1
```

CORRECT CODE :

```
> Users > k.Nikshitha > OneDrive > Desktop > New folder >
1   def calculate_area():
2       length = 5
3       width = 10
4       return length * width
5   print(calculate_area())
```

OUTPUT :

```
PS C:\Users\k.Nikshitha> & C:/Users/k.Nikshitha/AppData/Local/Programs/Python/Python314/python.exe
50
```

Task 11 (Type Error – Mixing Data Types Incorrectly)

Task: Analyze given code where integers and strings are added

incorrectly. Let AI detect and fix the error.

Bug: Adding integer and string

def add_values():

return 5 + "10"

print(add_values())

Requirements:

• Run the code to observe the error.

• AI should explain why int + str is invalid.

• Fix the code by type conversion (e.g., int("10") or str(5)).

• Verify with 3 assert cases.

Expected Output #6:

• Corrected code with type handling.

• AI explanation of the fix.

Successful test validation.

CODE :

```
C: > Users > k.Nikshitha > OneDrive > Desktop > New folder >
1   def add_values():
2   return 5 + "10"
3   print(add_values())
```

OUTPUT :

```
PS C:\Users\k.Nikshitha> & C:/Users/k.Nikshitha/AppData/Local/Programs/Python/Python314/python.exe
  File "c:\Users\k.Nikshitha\OneDrive\Desktop\New folder\TYPE ERROR.py", line 2
    return 5 + "10"
    ^^^^^^
IndentationError: expected an indented block after function definition on line 1
```

CORRECT CODE :

```
> Users > k.Nikshitha > OneDrive > Desktop > New folder >
1   def add_values():
2       return 5 + int("10")
3   print(add_values())
```

OUTPUT :

```
PS C:\Users\k.Nikshitha> & C:/Users/k.Nikshitha/AppData/Local/Programs/Python/Python314/python.exe
15
PS C:\Users\k.Nikshitha>
```

Task 12

(Type Error – String + List Concatenation)

Task: Analyze code where a string is incorrectly added to a list.

Bug: Adding string and list

def combine():

return "Numbers: " + [1, 2, 3]

print(combine())

Requirements:

• Run the code to observe the error.

• Explain why str + list is invalid.

• Fix using conversion (str([1,2,3]) or " ".join()).

• Verify with 3 assert cases.

Expected Output:

• Corrected code

• Explanation

• Successful test validation

CODE :

```
C: > Users > k.Nikshitha > OneDrive > Desktop > New folder >
1   def combine():
2   return "Numbers: " + [1, 2, 3]
3   print(combine())
```

OUTPUT :

```
PS C:\Users\k.Nikshitha> & C:/Users/k.Nikshitha/AppData/Local/Programs/Python/Python314/python.exe
  File "c:\Users\k.Nikshitha\OneDrive\Desktop\New folder\string list+.py", line 2
    return "Numbers: " + [1, 2, 3]
    ^^^^^^
IndentationError: expected an indented block after function definition on line 1
```

CORRECT CODE :

```
C: > Users > k.Nikshitha > OneDrive > Desktop > New folder >
1   def combine():
2       numbers = [1, 2, 3]
3       return "Numbers: " + str(numbers)
4   print(combine())
```

OUTPUT :

```
PS C:\Users\k.Nikshitha> & C:/Users/k.Nikshitha/AppData/Local/Programs/Python/Python314/python.exe
Numbers: [1, 2, 3]
PS C:\Users\k.Nikshitha>
```

Task 13

(Type Error – Multiplying String by Float)

Task: Detect and fix code where a string is multiplied by a float.

Bug: Multiplying string by float

def repeat_text():

return "Hello" * 2.5

print(repeat_text())

Requirements:

• Observe the error.

• Explain why float multiplication is invalid for strings.

• Fix by converting float to int.

• Add 3 assert test cases.

CODE :

```
: > Users > k.Nikshitha > OneDrive > Desktop > New folder >
1   def repeat_text():
2   return "Hello" * 2.5
3   print(repeat_text())
```

OUTPUT :

```
PS C:\Users\k.Nikshitha> & C:/Users/k.Nikshitha/AppData/Local/Programs/Python/Python314/python.exe
y"
  File "c:\Users\k.Nikshitha\OneDrive\Desktop\New folder\MULTIPLY STRING.py", line 2
    return "Hello" * 2.5
    ^^^^^^
IndentationError: expected an indented block after function definition on line 1
```

CORRECT CODE :

```
C: > Users > k.Nikshitha > OneDrive > Desktop > New folder >
1   def repeat_text():
2       return "Hello" * 2
3   print(repeat_text())
```

OUTPUT :

```
PS C:\Users\k.Nikshitha> & C:/Users/k.Nikshitha/AppData/Local/Programs/Python/Python314/python.exe
y"
HelloHello
```

Task 14

(Type Error – Adding None to Integer)

Task: Analyze code where None is added to an integer.

Bug: Adding None and integer

def compute():

value = None

return value + 10

print(compute())

Requirements:

• Run and identify the error.

• Explain why NoneType cannot be added.

• Fix by assigning a default value.

• Validate using asserts.

CODE :

```
C: > Users > k.Nikshitha > OneDrive > Desktop > New folder >
1   def compute():
2   value = None
3   return value + 10
4   print(compute())
```

OUTPUT :

```
PS C:\Users\k.Nikshitha> & C:/Users/k.Nikshitha/AppData/Local/Programs/Python/Python314
py"
  File "c:\Users\k.Nikshitha\OneDrive\Desktop\New folder\add none integer.py", line 2
    value = None
    ^^^^^
IndentationError: expected an indented block after function definition on line 1
```

CORRECT CODE :

```
C: > Users > k.Nikshitha > OneDrive > Desktop > New folder >
def compute():
    value = None
    if value is not None:
        return value + 10
    else:
        return "Error: value is None"
print(compute())
```

OUTPUT :

```
PS C:\Users\k.Nikshitha> & C:/Users/k.Nikshitha/AppData/Local/Programs/Python/Python314/python.exe
py"
Error: value is None
```

Task 15

(Type Error – Input Treated as String Instead of

Number)

Task: Fix code where user input is not converted properly.

Bug: Input remains string

def sum_two_numbers():

a = input("Enter first number: ")

b = input("Enter second number: ")

return a + b

print(sum_two_numbers())

Requirements:

• Explain why input is always string.

• Fix using int() conversion.

• Verify with assert test cases.

CODE :

```
: > Users > k.Nikshitha > OneDrive > Desktop > New folder >
def sum_two_numbers():
a = input("Enter first number: ")
b = input("Enter second number: ")
return a + b
print(sum_two_numbers())
```

OUTPUT :

```
PS C:\Users\k.Nikshitha> & C:/Users/k.Nikshitha/AppData/Local/Programs/Python/Python314/python.exe
py"
  File "c:\Users\k.Nikshitha\OneDrive\Desktop\New folder\STRING AS NUMBER.py", line 2
    a = input("Enter first number: ")
    ^
IndentationError: expected an indented block after function definition on line 1
PS C:\Users\k.Nikshitha>
```

CORRECT CODE :

```
> Users > k.Nikshitha > OneDrive > Desktop > New folder >
1   def sum_two_numbers():
2       a = input("Enter first number: ")
3       b = input("Enter second number: ")
4       return int(a) + int(b)
5   print(sum_two_numbers())
```

OUTPUT :

```
PS C:\Users\k.Nikshitha> & C:/Users/k.Nikshitha/AppData/Local/Programs/Python/Python314/python.exe
py"
Enter first number: 6
Enter second number: 9
15
PS C:\Users\k.Nikshitha>
```