AI Assisted Coding

Name : Vineeth-Chidurala **Date** : 06-02-2026

Ht.No. : 2303A52447

Task - 1 : Mutable Default Argument – Function Bug
Analyze given code where a mutable default argument causes unexpected behavior. Use AI to fix it

Prompt : Solve this error.

Screenshots :

```
def add_item(item, items=[]):
    items.append(item)
    return items
print(add_item(1))
print(add_item(2))
```

Output :

```
    print(add_item(2))
[2]  ✓ 0.0s
...  [1]
     [1, 2]
```

Task – 2 : Floating-Point Precision Error
Analyze given code where floating-point comparison fails. Use AI to correct with tolerance.

Prompt : Solve this error.

ScreenShots :

```
def check_sum():
    return (0.1 + 0.2) == 0.3
print(check_sum())
```

Output :

```
    print(check_sum())

[3]  ✓ 0.0s

...  False
```

Task-3 : Recursion Error – Missing Base Case

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

Prompt : Solve this error.

ScreenShots :

```
def countdown(n):
    print(n)
    if n == 0:
        return
    return countdown(n-1)
countdown(5)
```

Output :

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

[5]  ✓ 0.0s

...  5
     4
     3
     2
     1
     0
```

Task-4 : Dictionary Key Error.

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

Prompt : Solve this error.

Screenshots :

```
def get_value():
    data = {"a": 1, "b": 2}
    return data.get("c")
print(get_value())
```

Output :

```
    return data.get("c")
print(get_value())

[7]  ✓ 0.0s

...  None
```

Task-5 : Infinite Loop – Wrong Condition

Analyze given code where loop never ends. Use AI to detect and fix it.

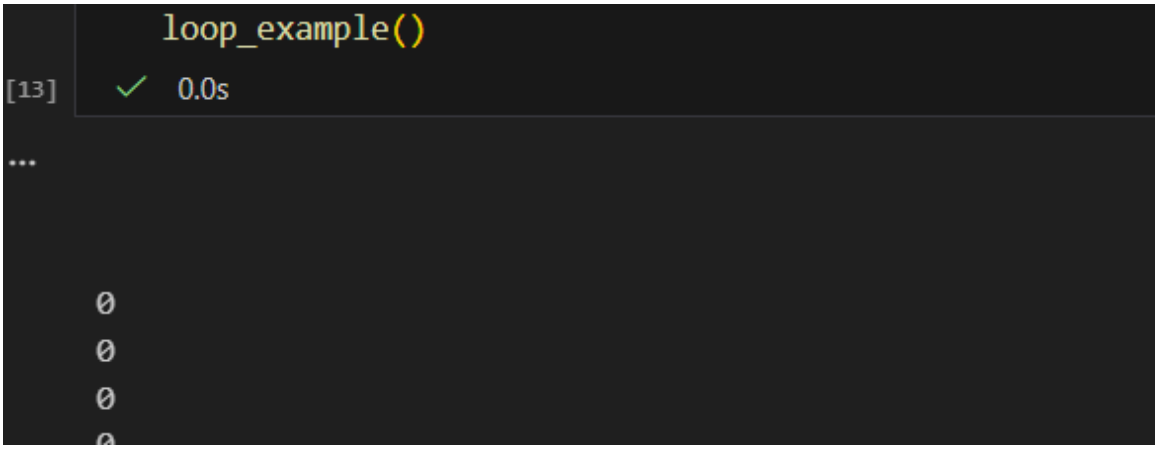
Prompt :

Screenshots :

```
def loop_example():
    i = 0
    while i < 5:
        print(i)
        def loop_example():
            i = 0
            while i < 5:
                print(i)
                i += 1

loop_example()
```

Output :

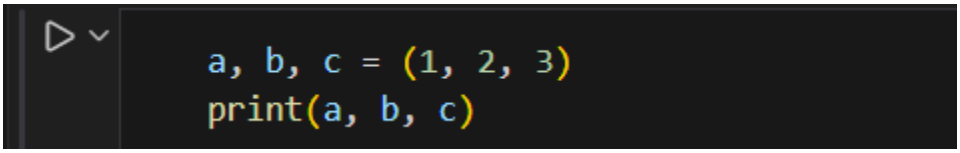


Task- 6 : Unpacking Error – Wrong Variables

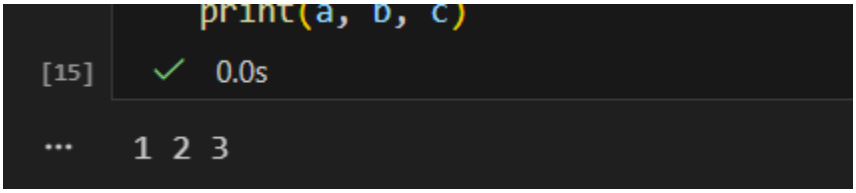
Analyze given code where tuple unpacking fails. Use AI to fix it.

Prompt : Solve the unpacking error.

Screenshots :



Output :



Task- 7 : Mixed Indentation – Tabs vs Spaces

Analyze given code where mixed indentation breaks execution. Use AI to fix it.

Prompt : Solve mix indentation error.

Screenshots :

```
def func():
    x = 5
    y = 10
    def inner_func():
        x = 5
        y = 10
        return x + y
    return inner_func()
print(func())
```

Output :

```
        x = 5
        y = 10
        return x + y
    return inner_func()
print(func())
```

0.0s

```
15
```

Task- 8 : Import Error – Wrong Module Usage

Analyze given code with incorrect import. Use AI to fix

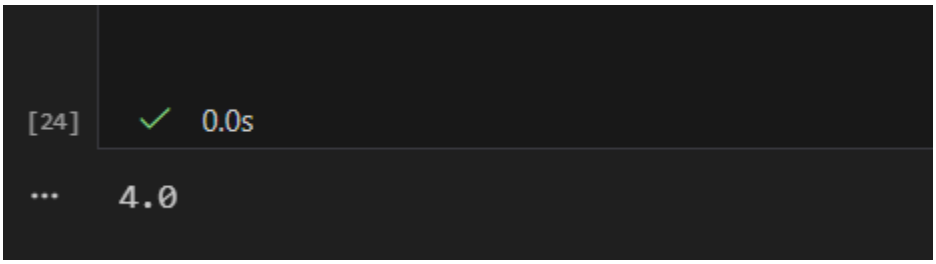
Prompt : Solve the input error.

Screenshots :

```
import math

print(math.sqrt(16))
```

Output :

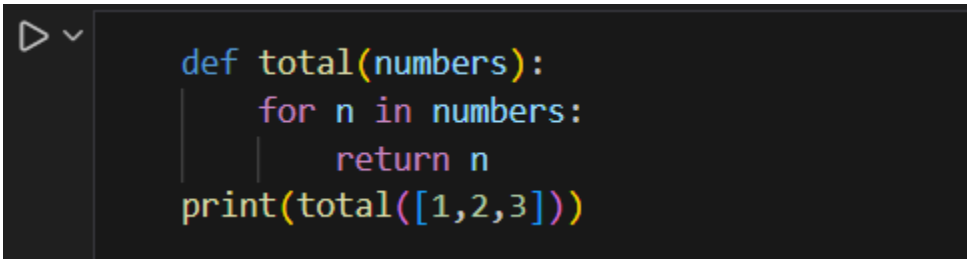


Task- 9 : Unreachable Code – Return Inside Loop

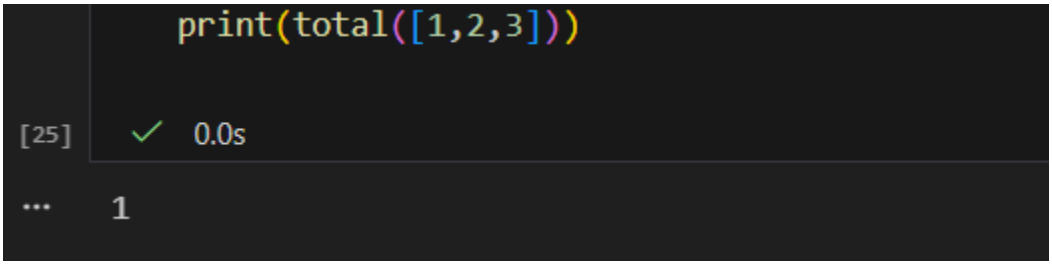
Analyze given code where a return inside a loop prevents full iteration. Use AI to fix it.

Prompt : Solve the early return error.

Screenshots :



Output :



Task- 10 : Name Error – Undefined Variable

Analyze given code where a variable is used before being defined. Let AI detect and fix the error.

Prompt : Solve the missing variable error.

Screenshots :

```
def calculate_area(length, width):
    return length * width

print(calculate_area(5, 4))
```

Output :

```
print(calculate_area(5, 4))

[27]  ✓ 0.0s

...  20
```

Task- 11 : Mixing Data Types Incorrectly

Analyze given code where integers and strings are added incorrectly. Let AI detect and fix the error.

Prompt : Solve this type error

Screenshots :

```
def add_values():
    return 5 + int("10")
print(add_values())
```

Output :

```
print(add_values())

[29]  ✓ 0.0s

...  15
```

Task- 12 : Type Error – String + List Concatenation

Analyze code where a string is incorrectly added to a list

Prompt : Solve the error inside the code.

Screenshots :

```
def combine():
    return "Numbers: " + [1, 2, 3]
print("Numbers: " + str([a, b, c]))
```

Output :

```
print( Numbers:    + str([a, b, c]))

31]   ✓ 0.0s

·   Numbers: [1, 2, 3]
```

Task- 13 : Type Error – Multiplying String by Float

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

Prompt : Solve the multiplication error of the string and integer.

Screenshots :

```
def repeat_text():
    return "Hello" * 2.5
def repeat_text():
    return "Hello" * int(2.5)
print(repeat_text())
```

Output :

```
print(repeat_text())

[33]  ✓ 0.0s

...   HelloHello
```

Task- 14 : Type Error – Adding None to Integer

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

Prompt : Solve this error

Screenshots :

```
def compute():
    value = 10
    return value + 10

print(compute())
```

Output :

```
print(compute())

35]  ✓ 0.0s

..   20
```

Task- 15 : Type Error – Input Treated as String Instead of Number

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

Prompt : Solve this error

Screenshots :

```python
def sum_two_numbers():
    a = int(input("Enter first number: "))
    b = int(input("Enter second number: "))
    return a + b

print(sum_two_numbers())
```

Output :

```
    return a + b

print(sum_two_numbers())
```

[36] ✓ 5.1s

```
25
```