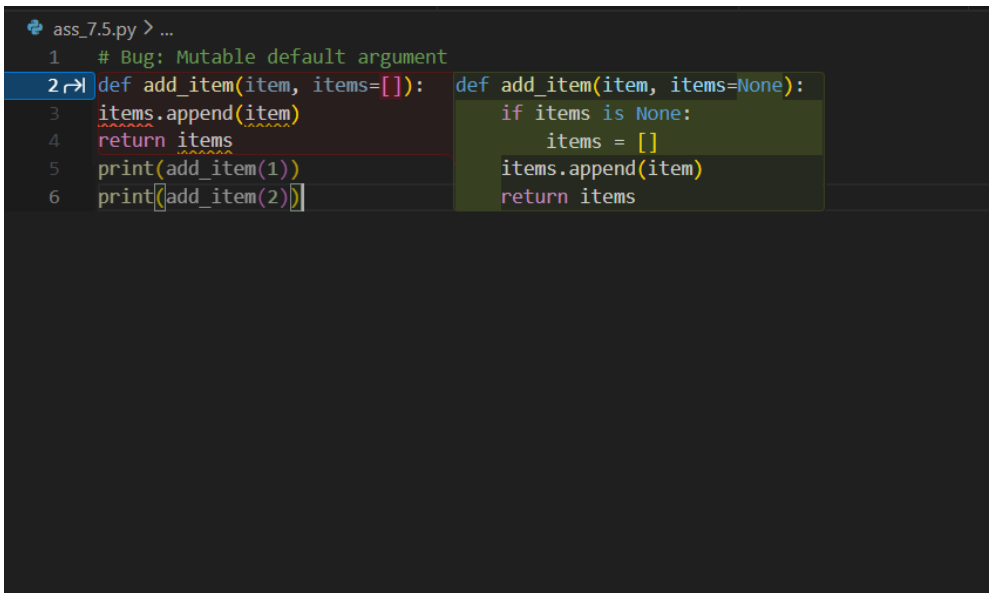
ASSIGNMENT-7.5

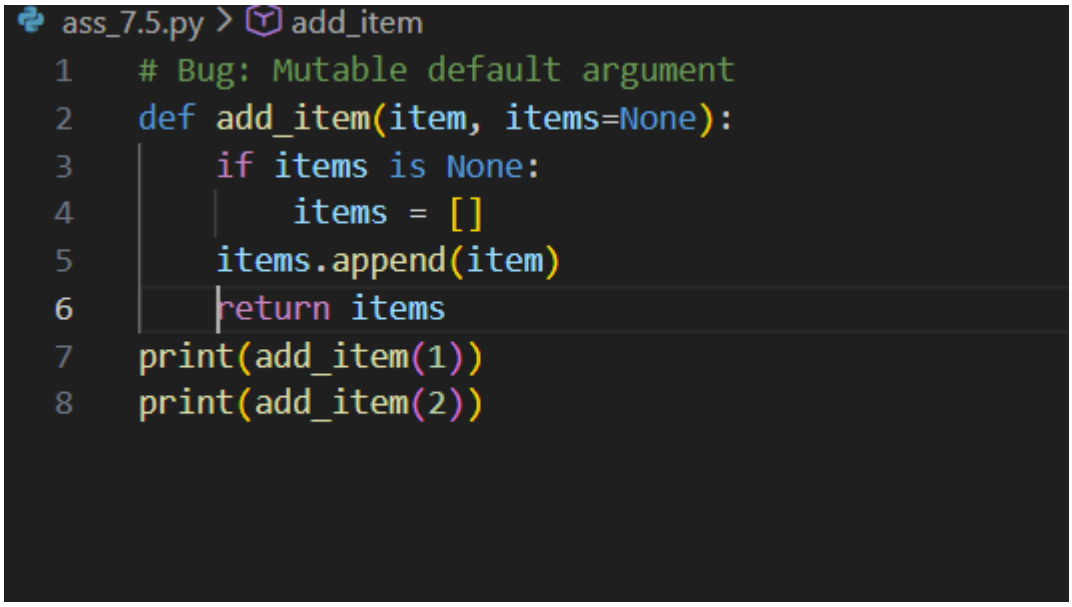
CH.Harshini

2303a51185

Batch:03

1)INPUT:





OUTPUT:

```
ai_coding/ass_7.5.py
[1]
[2]
```

2)INPUT:

```
1   # Bug: Floating point precision issue
2   def check_sum():
3   return (0.1 + 0.2) == 0.3
        return abs((0.1 + 0.2) - 0.3) < 1e-10  # Use a small tolerance for floating point comparison
4   print(check_sum())
5
```

```
# Bug: Floating point precision issue
def check_sum():
    return abs((0.1 + 0.2) - 0.3) < 1e-10  # Use a small tolerance for floating point comparison
print(check_sum())
```

OUTPUT:

```
True
```

3)INPUT:

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

```python
# Bug: No base case
def countdown(n):
    if n == 0:
        return
    print(n)
    countdown(n-1)
countdown(5)
```

OUTPUT:

```
5
4
3
2
1
```

4)INPUT:

```python
ass_7.5.py > ...
2   def get_value():
3   data = {"a": 1, "b": 2}
4   return data["c"]
    return data.get("c", "Key not found")
5   print(get_value())
```

```python
# Bug: Accessing non-existing key
def get_value():
    data = {"a": 1, "b": 2}
    return data.get("c", "Key not found")
print(get_value())
```

OUTPUT:

```
Key not found
```

5)INPUT:

```python
# Bug: Infinite loop
def loop_example():
i = 0
while i < 5:
print(i)
    i += 1
```

```
Go  Run  Terminal  Help
Welcome | # AI-Generated Logic Without Modularizat.py | ass_3.2.py
ass-7.5.py > ...
```

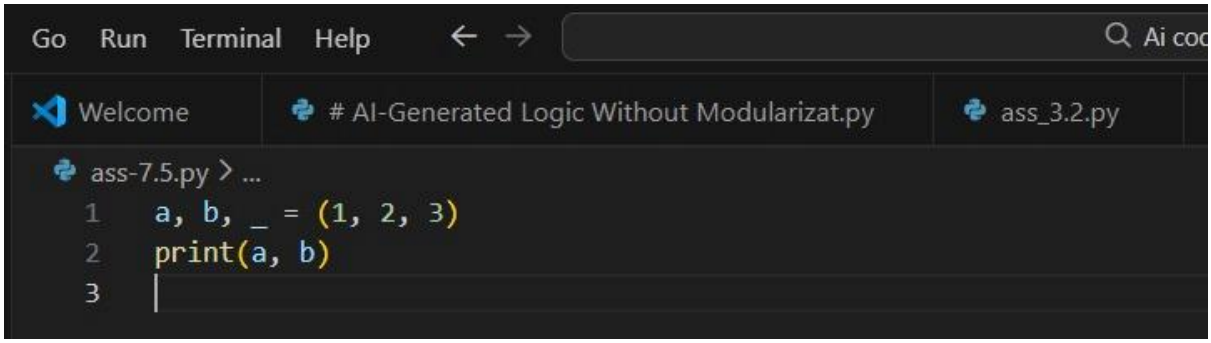
```python
def loop_example():
    i = 0
    while i < 5:
        print(i)
        i += 1   # Increment added

loop_example()
```

OUTPUT:

6)INPUT:

```
# Bug: Wrong unpacking
a, b = (1, 2, 3)
Expected Output: Correct unpacking or using _ for extra values.
```

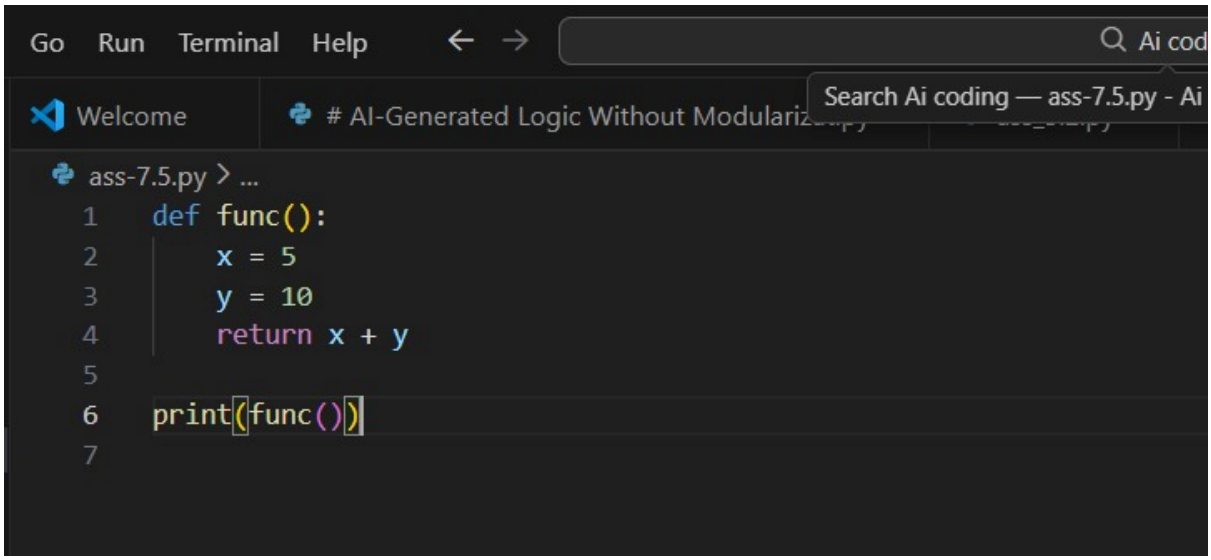


OUTPUT:

7)INPUT:

```
# Bug: Mixed indentation
def func():
    x = 5
      y = 10
    return x+y
```

Expected Output : Consistent indentation applied.

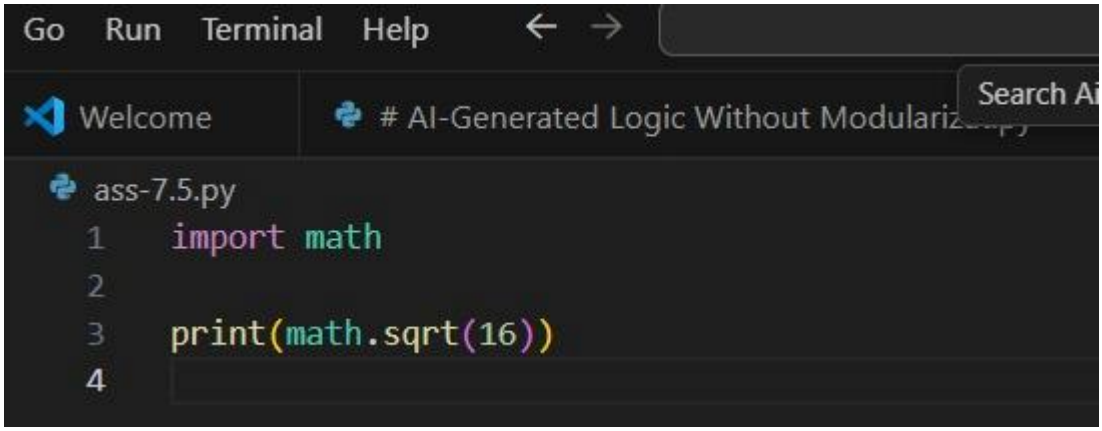


OUTPUT:

8)INPUT:

```
# Bug: Wrong import
import maths
print(maths.sqrt(16))
```

Expected Output: Corrected to import math


Go Run Terminal Help
Welcome
AI-Generated Logic Without Modulariz
Search Ai
ass-7.5.py
1 import math
2
3 print(math.sqrt(16))
4

OUTPUT:

4.0