

BATCH:06 Name :Chaithra Alluri

id:2403A51105

Task Description#1

- Paste a function with a missing colon (add(a, b)), and let AI fix the syntax error.

```
python

def add(a, b)
    return a + b
```

CODE:

```
def add(a, b):
    return a + b

print(add(3, 5))
```

FIX OF ERROR:

It must end with a colon (:).

Expected Output#1

- Corrected function with syntax fix

8



Task Description#2 (Loops)

- Identify and fix a logic error in a loop that causes infinite iteration.

```
python

def count_down(n):
    while n >= 0:
        print(n)
        n += 1 # Should be n -= 1
```

CODE:

```
def count_down(n):  
    while n >= 0:  
        print(n)  
        n -= 1    #
```

FIXERROR:

The loop condition is while n >= 0.

Inside the loop, n += 1 makes n increase forever, so it never becomes less than 0

→ infinite loop

Expected Output#2

- AI fixes increment/decrement error

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

Task Description#3

- Debug a runtime error caused by division by zero. Let AI insert try-except.

```
# Debug the following code  
def divide(a, b):  
    return a / b  
  
print(divide(10, 0))
```

CODE:

```
def divide(a, b):
    try:
        return a / b
    except ZeroDivisionError:
        return "Error: Division by zero is not allowed"

print(divide(10, 0))
```

FIXERROR:

This will raise a `ZeroDivisionError` at runtime because `b = 0`.
In Python, division by zero is not allowed

Expected Output#3

- Corrected function with safe error handling

```
Error: Division by zero is not allowed
```

Task Description#4

- Provide a faulty class definition (missing `self` in parameters). Let AI fix it

```
python

class Rectangle:
    def __init__(length, width):
        self.length = length
        self.width = width
```

CODE:

```
class Rectangle:
    def __init__(self, length, width):
        self.length = length
        self.width = width
```

FIXERROR:

In Python, instance methods (including `__init__`) must include `self` as the first parameter. Without `self`, Python doesn't know which object's attributes (`length` and `width`) to assign.

Expected Output#4

- Correct `__init__()` method and explanation

```
rect = Rectangle(10, 5)
print(rect.length)  # Output: 10
print(rect.width)   # Output: 5
```

Task Description#5

- Access an invalid list index and use AI to resolve the Index Error.

```
python

numbers = [1, 2, 3]
print(numbers[5])
```

CODE:

Fix 1: Check index before accessing

```
numbers = [1, 2, 3]
index = 5

if index < len(numbers):
    print(numbers[index])
else:
    print("Index out of range")
```

Fix 2: Use try-except

```
numbers = [1, 2, 3]

try:
    print(numbers[5])
except IndexError:
    print("Error: Invalid index")
```

Fix 3: Safe access with default value

```
numbers = [1, 2, 3]
index = 5

value = numbers[index] if index < len(numbers) else None
print(value)    # Output: None
```

FIXERROR:

IndexError: list index out of range

Expected Output#5

- AI suggests checking length or using safe access logic

Fix 1: Check index before accessing:

Index out of range

Fix 2: Use try-except:

Error: Invalid index

Fix 3: Safe access with default value:

None