Lab 7.3

K.Sai Karthik

2303A51678

Batch-28

Task 1: Fixing Syntax Errors

Scenario

You are reviewing a Python program where a basic function definition contains a syntax error.

Requirements:

Typical Python error message:

javascript Copy code

```
SyntaxError: expected ':'
```

Output

3. Corrected Function (AI Fix)

The AI automatically corrects the issue by adding the missing colon.

python Copy code

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

4. AI Explanation of the Fix

> The syntax error occurred because Python function definitions must end with a colon.
> The colon tells Python that the indented block below belongs to the function.
> Adding the colon resolves the syntax error and allows the function to run correctly.

Task 2: Debugging Logic Errors in Loops

Scenario

You are debugging a loop that runs infinitely due to a logical mistake.

Requirements

1. Original Loop (With Logic Error)

This loop is intended to count **up to 5**, but it never stops.

python Copy code

```
i = 1
while i <= 5:
    print(i)
    i -= 1
```

---

2. AI Identifies the Cause of Infinite Iteration

What the AI notices:

- The loop condition is `i <= 5`
- The variable `i` starts at `1`
- Inside the loop, `i` is **decremented** (`i -= 1`)
- This causes `i` to become `0, -1, -2, ...`
- The condition `i <= 5` is **always true**

👉 Result: the loop runs forever.

3. AI Fixes the Loop Logic

Since the loop condition checks for `i <= 5`, the counter should **increase**, not decrease.

✅ **Corrected Loop**

python Copy code

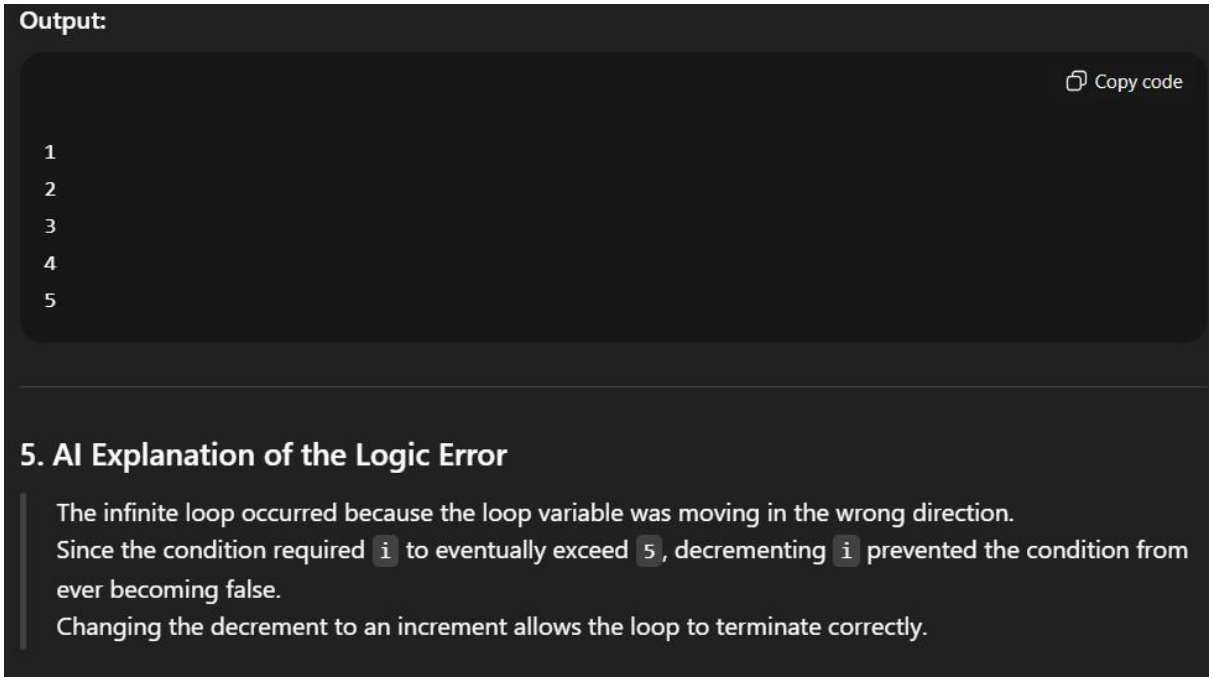
```
i = 1
while i <= 5:
    print(i)
    i += 1
```

---

4. Analysis of the Corrected Behavior

- `i` starts at `1`
- Each iteration increases `i` by `1`
- When `i` becomes `6`, the condition `i <= 5` becomes false
- The loop exits normally

Output

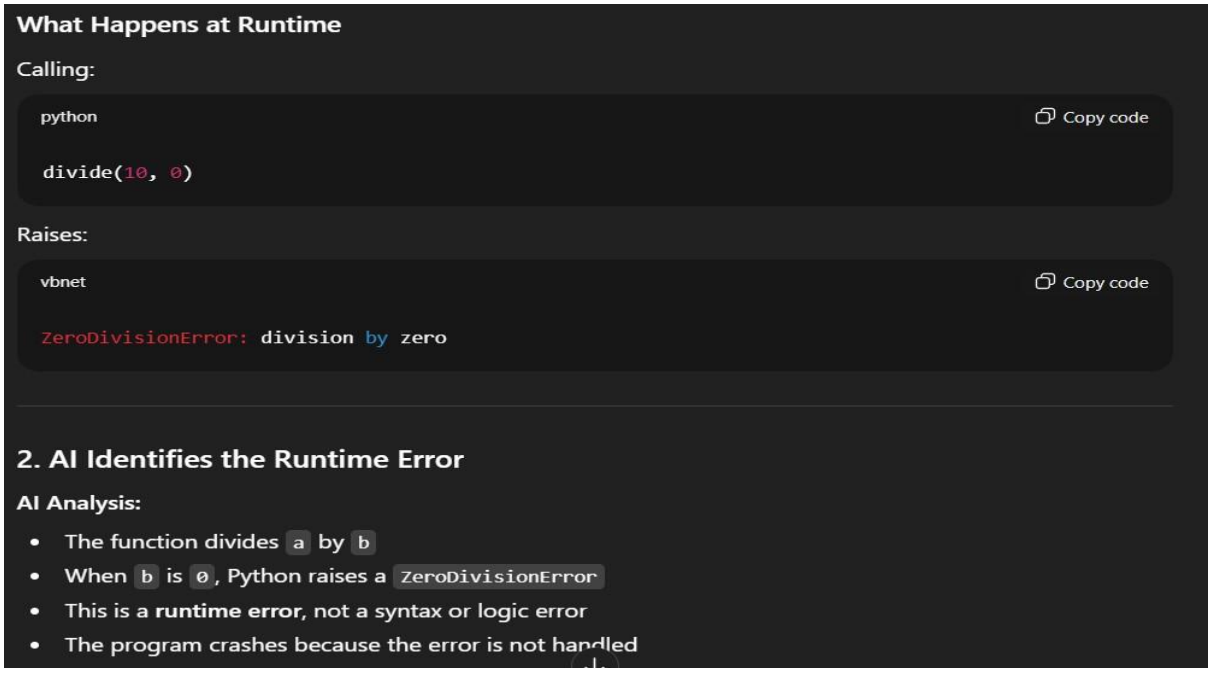


Task 3: Handling Runtime Errors (Division by Zero)

Scenario

A Python function crashes during execution due to a division by zero error.

Requriments



3. AI Adds try-except for Safe Execution

The AI fixes the issue by catching the exception using a `try-except` block.

✅ Corrected Function (With Error Handling)

python Copy code

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

---

4. Review of the Corrected Behavior

Normal case:

python Copy code

```python
divide(10, 2)
```

Output:

Output:

Copy code

```
5.0
```

Error case:

python Copy code

```python
divide(10, 0)
```

Output:

vbnet Copy code

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

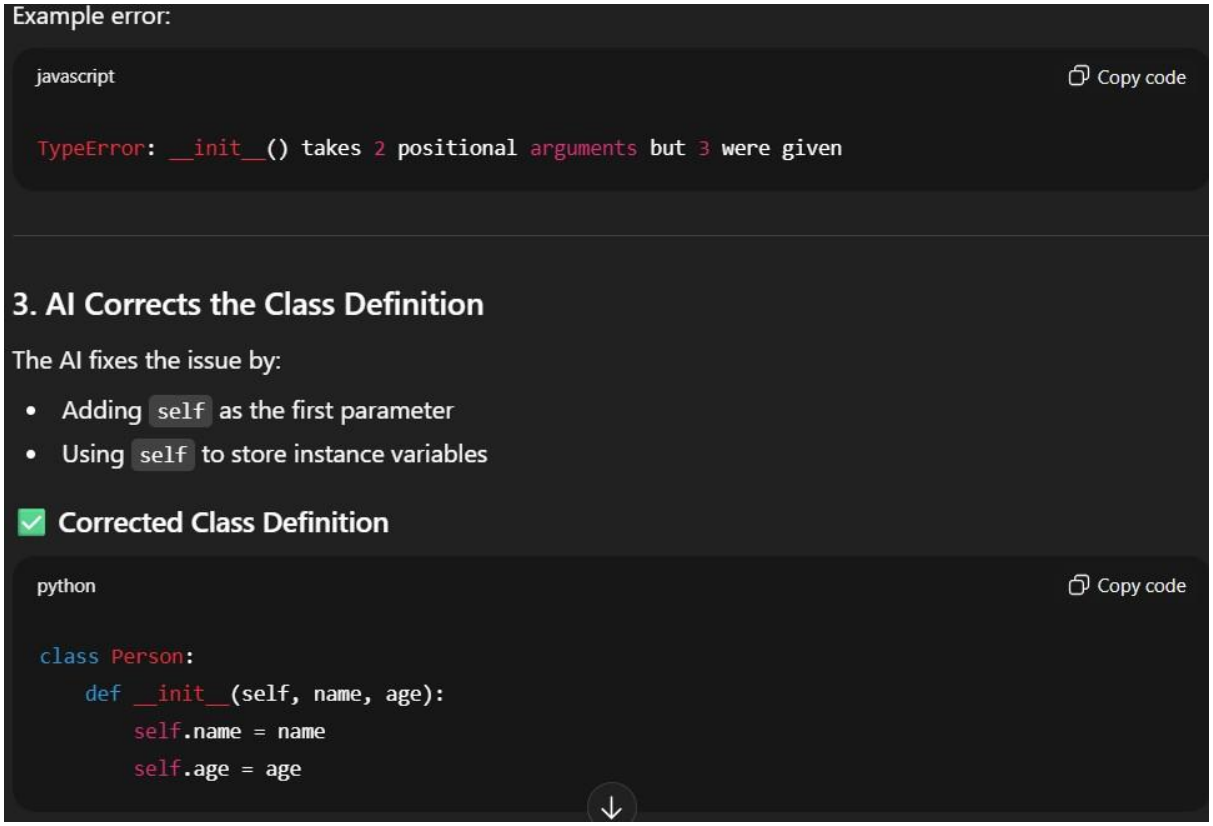
The function now:

- Executes safely
- Does not crash the program

Task 4: Debugging Class Definition Errors

Scenario

You are given a faulty Python class where the constructor is incorrectly defined.

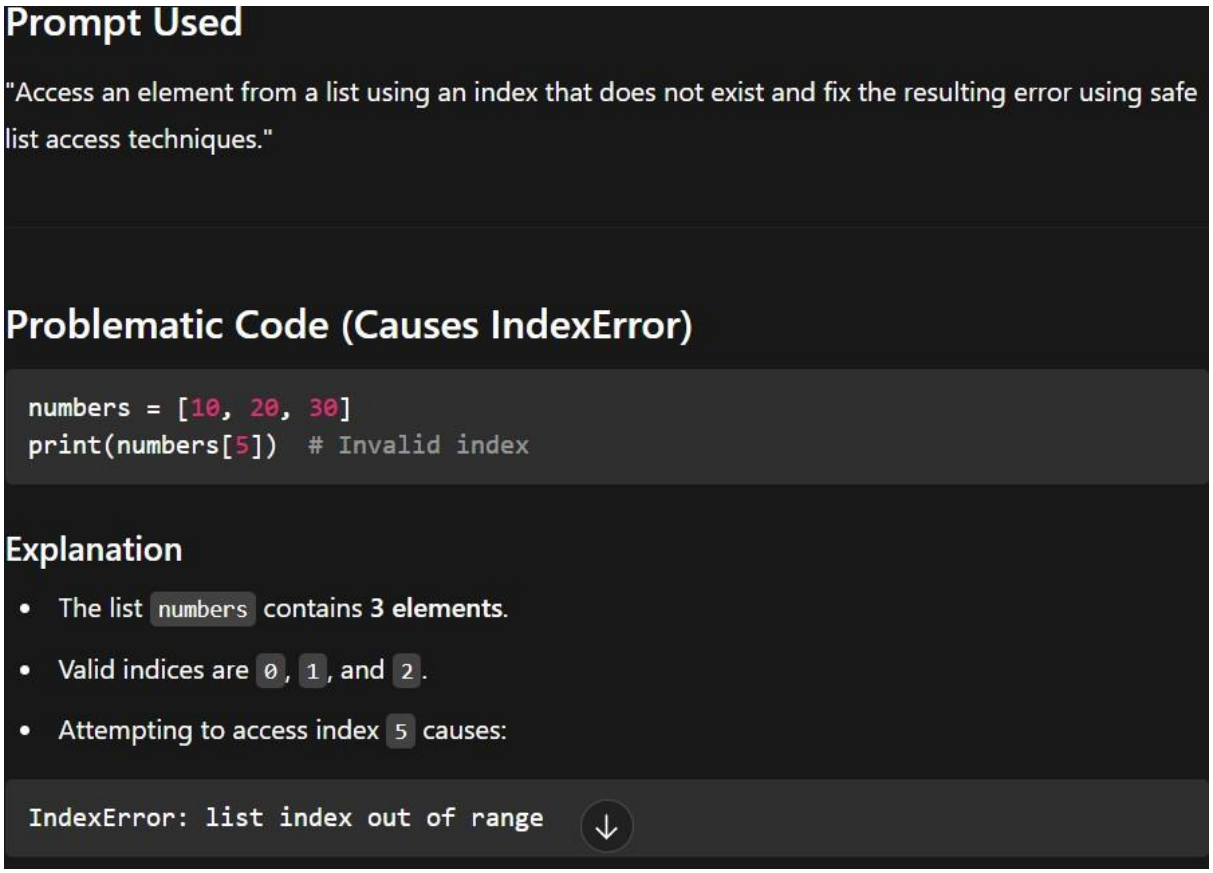


Output

Task 5: Resolving Index Errors in Lists

Scenario

A program crashes when accessing an invalid index in a list.



Output