ASSIGNMENT-6.2

Name: Akshaya Nemalipuri

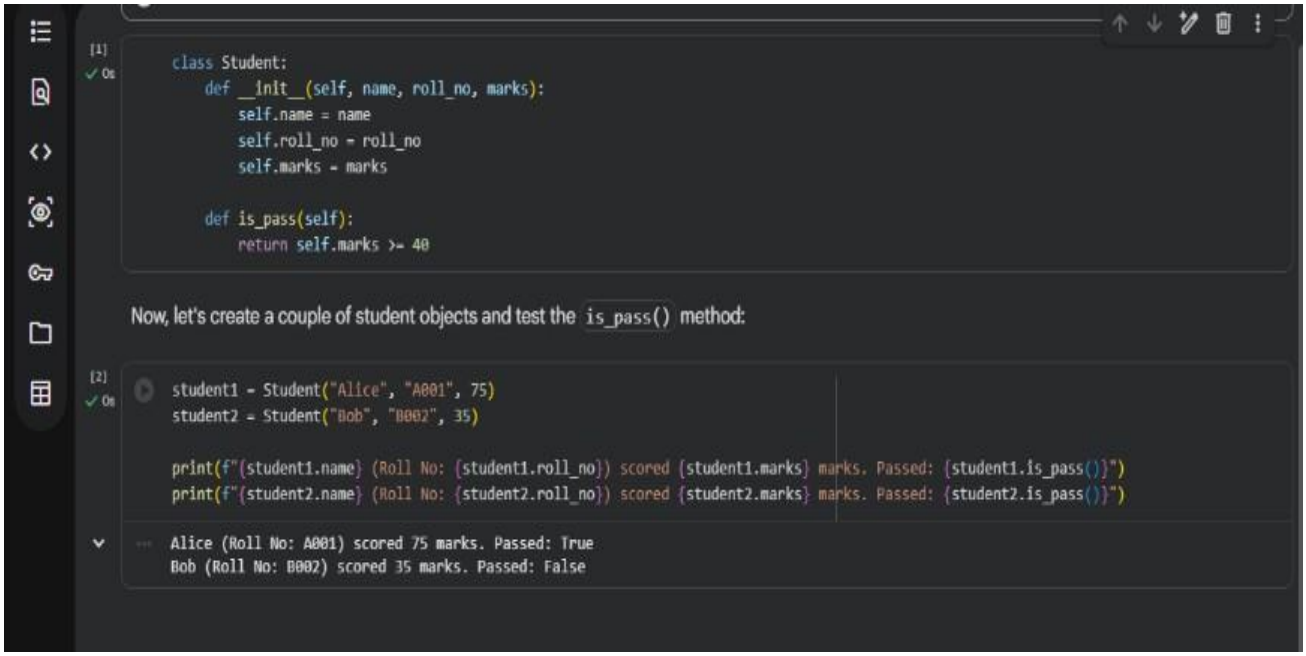
Batch: 37

Roll No: 2303A2441

Lab: 06

Task Description-1 (Classes – Data Validation) :

- Prompt AI to generate a Student class with attributes: name, roll_no, and marks. Add a method is_pass() that returns whether the student has passed (marks ≥ 40).



Explanation :

- The __init__ constructor initializes student details.
- The is_pass() method checks if marks are greater than or equal to 40.

- Returns True for pass and False for fail.

Analysis

- Code is simple, readable, and correct.
- Validation logic is clearly implemented.
- AI-generated class design follows Python best practices.

Task Description-2 (Loops – Pattern Generation)

- Ask AI to generate a function that prints a right-angled triangle star pattern using a for loop. Then regenerate the same pattern using a while loop.

Explanation

- The loop counter controls the number of stars printed.
- Both loops produce identical output with different logic structures.

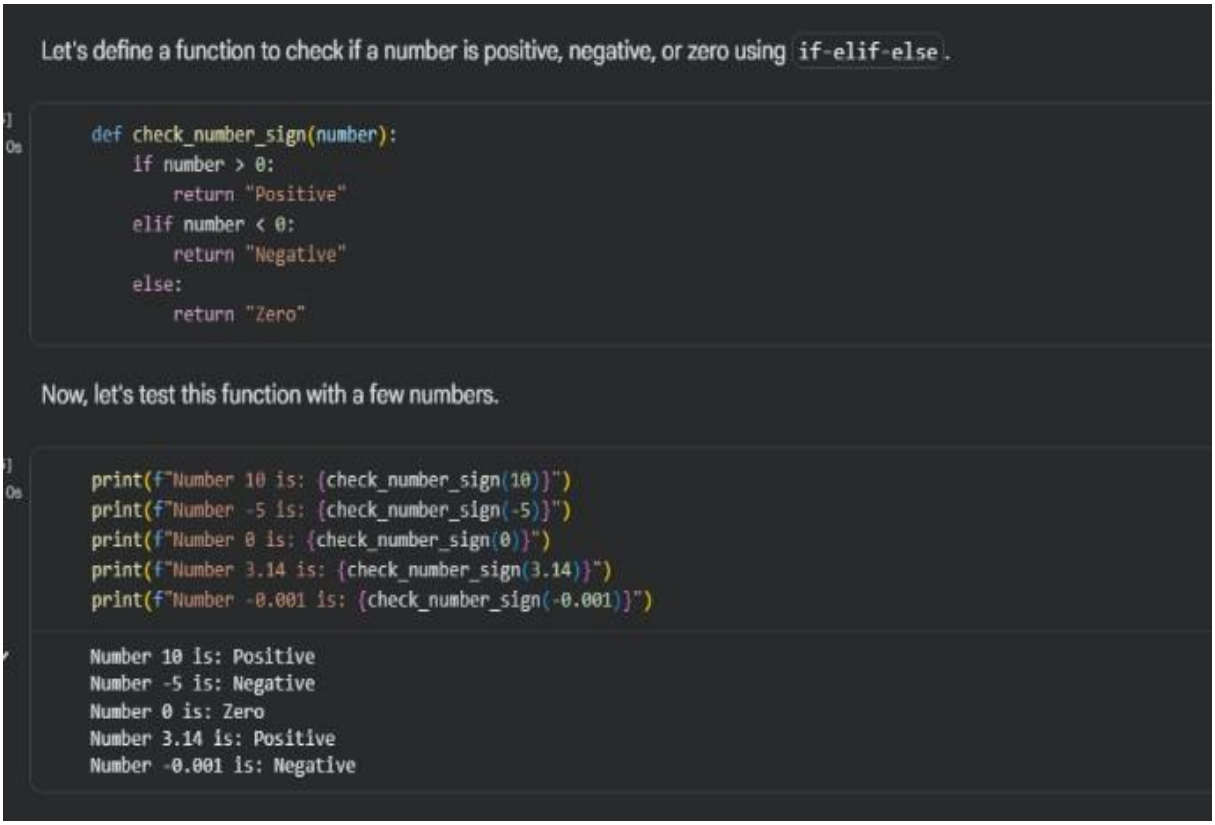
Analysis

- AI correctly used loop conditions.

- Logical flow is easy to understand.
- Demonstrates equivalent solutions using different looping constructs.

Task Description-3 (Conditional Statements – Number Analysis) :

- Ask AI to write a function that checks whether a given number is positive, negative, or zero using if-elif-else. Test the function with multiple inputs.



Explanation

- Uses if, elif, and else to classify numbers.
- Covers all possible conditions.

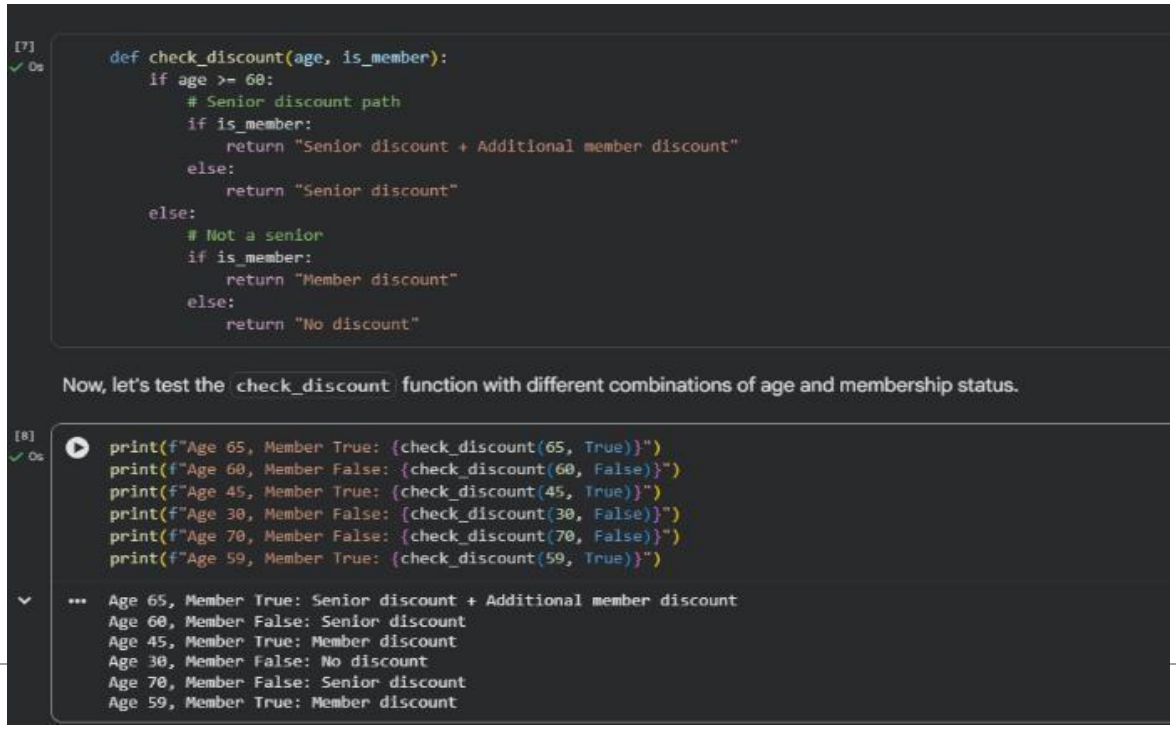
Analysis

- Decision logic is complete and correct.
- AI handled edge case (zero) properly.
- Code is concise and readable.

Task Description-4 (Nested Conditionals)

- Generate a function check_discount(age, is_member) that determines discount eligibility: • Age ≥ 60 → Senior discount • Member → Additional discount

Use nested if statements.



```
[7]
✓ 0s
def check_discount(age, is_member):
    if age >= 60:
        # Senior discount path
        if is_member:
            return "Senior discount + Additional member discount"
        else:
            return "Senior discount"
    else:
        # Not a senior
        if is_member:
            return "Member discount"
        else:
            return "No discount"
```

Now, let's test the `check_discount` function with different combinations of age and membership status.

```
[8]
✓ 0s
print(f"Age 65, Member True: {check_discount(65, True)}")
print(f"Age 60, Member False: {check_discount(60, False)}")
print(f"Age 45, Member True: {check_discount(45, True)}")
print(f"Age 30, Member False: {check_discount(30, False)}")
print(f"Age 70, Member False: {check_discount(70, False)}")
print(f"Age 59, Member True: {check_discount(59, True)}")

...
Age 65, Member True: Senior discount + Additional member discount
Age 60, Member False: Senior discount
Age 45, Member True: Member discount
Age 30, Member False: No discount
Age 70, Member False: Senior discount
Age 59, Member True: Member discount
```

Explanation

- Nested if statements check age first, then membership.
- Clear hierarchical decision flow.

Analysis

- AI correctly implemented nested conditionals.
- Logic is easy to trace and debug.
- Matches real-world discount scenarios.

Task Description-5 (Class – Mathematical Opera)

- Ask AI to create a Circle class with methods to calculate area () and circumference () given the radius.

Explanation

- Uses math.pi for accurate calculations.
- Area and circumference formulas are correctly implemented.
- Class design is reusable.

Analysis

- Mathematical logic is accurate.
- Code follows object-oriented principles.
- AI-generated solution is efficient and well-structured.