H.NO : 2303A52243
NAME: SRI SAHITH . G

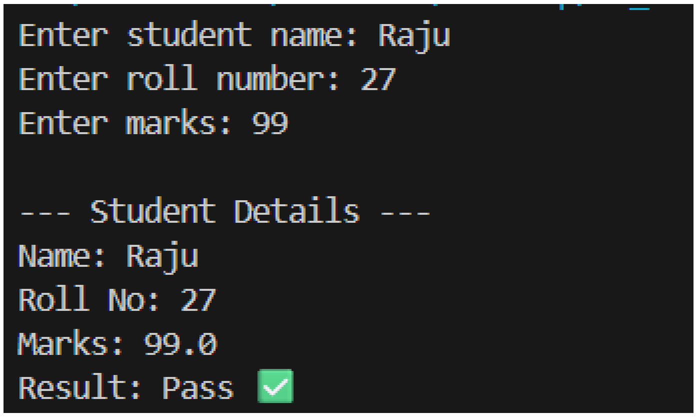
Lab 6: AI-Based Code Completion – Classes, Loops, and Conditionals
Lab Objectives:
• To explore AI-powered auto-completion features for core Python constructs.
• To analyze how AI suggests logic for class definitions, loops, and conditionals.
• To evaluate the completeness and correctness of code generated by AI assistants.
Lab Outcomes (LOs):
After completing this lab, students will be able to:
• Use AI tools to generate and complete class definitions and methods.
• Understand and assess AI-suggested loops for iterative tasks.
• Generate conditional statements through prompt-driven suggestions.
• Critically evaluate AI-assisted code for correctness and clarity.
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).
Expected Output-1
• Python class with constructor and validation logic
• Clear method implementation returning pass/fail status
• Students analyze correctness and clarity of class design

CODE:

```
class Student:
    def __init__(self, name, roll_no, marks):
        self.name = name
        self.roll_no = roll_no
        self.marks = marks

    def is_pass(self):
        return self.marks >= 40


# ---- Taking input from user ----
name = input("Enter student name: ")
```

```
roll_no = int(input("Enter roll number: "))
marks = float(input("Enter marks: "))

# Creating Student object
student1 = Student(name, roll_no, marks)

# Displaying student details
print("\n--- Student Details ---")
print("Name:", student1.name)
print("Roll No:", student1.roll_no)
print("Marks:", student1.marks)

# Checking pass/fail
if student1.is_pass():
    print("Result: Pass ✅")
else:
    print("Result: Fail ❌")
```

OUTPUT:



```
Enter student name: Raju
Enter roll number: 27
Enter marks: 99

--- Student Details ---
Name: Raju
Roll No: 27
Marks: 99.0
Result: Pass ✅
```

Prompt:

> **Generate a Python class named `Student` with attributes name, roll_no, and marks. Add a method `is_pass()` that returns True if marks ≥ 40, otherwise False.**

Explanation

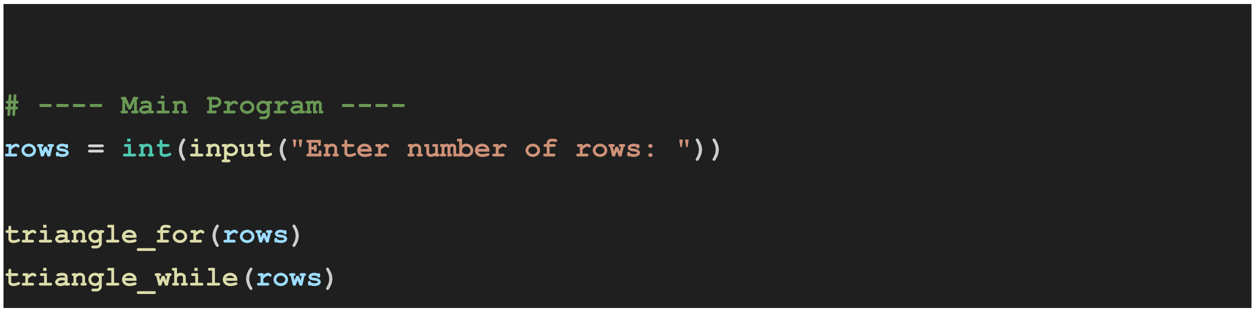
AI correctly:

- **Created a class**
- **Used a constructor to store data**
- **Added a method with a conditional check**

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.
Expected Output-2
• Correct pattern output using both loop types
• Logical loop structure with proper conditions

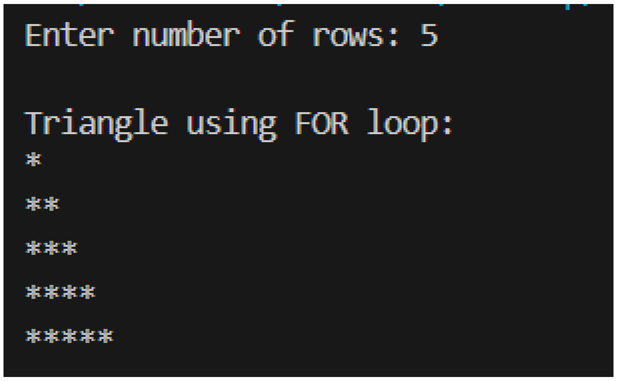
CODE:

```
# Function using FOR loop
def triangle_for(n):
    print("\nTriangle using FOR loop:")
    for i in range(1, n + 1):
        print("*" * i)


# Function using WHILE loop
def triangle_while(n):
    print("\nTriangle using WHILE loop:")
    i = 1
    while i <= n:
        print("*" * i)
        i += 1
```

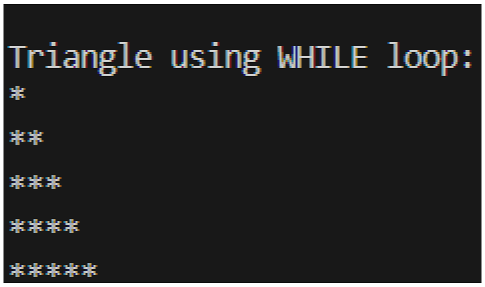


```python
# ---- Main Program ----
rows = int(input("Enter number of rows: "))

triangle_for(rows)
triangle_while(rows)
```

OUTPUT:



```
Enter number of rows: 5

Triangle using FOR loop:
*
**
***
****
*****
```



```
Triangle using WHILE loop:
*
**
***
****
*****
```

Prompt:

> **Write Python functions to print a right-angled triangle star pattern using a for loop and a while loop.**

EXPLANATION:

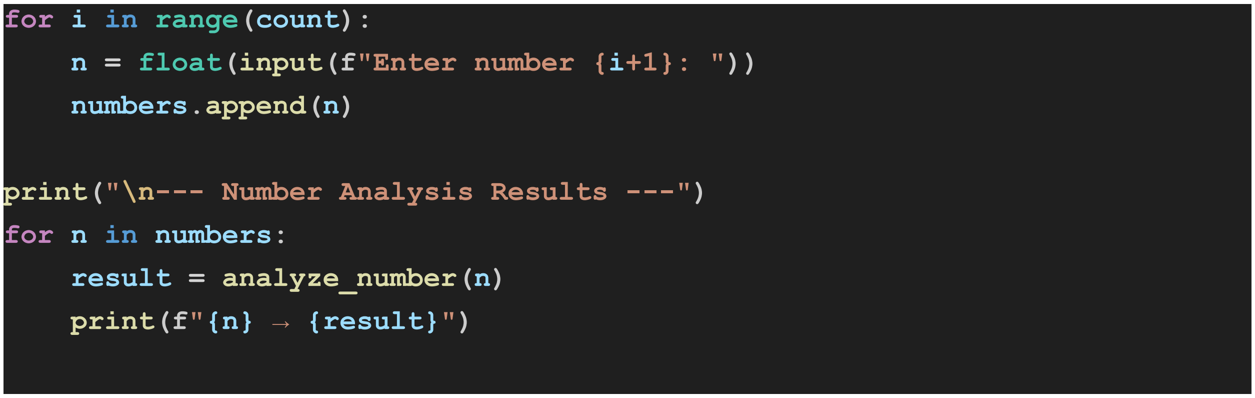
AI demonstrated:

- **Iteration using `for`**
- **Same logic recreated using `while`**
- **Proper loop conditions**

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.
Expected Output-3
• Function correctly classifies numbers
• Proper handling of all conditions
• Students analyze decision logic

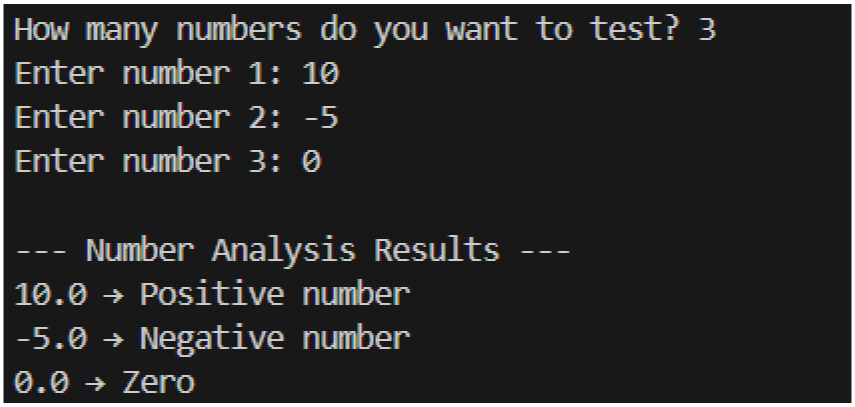
CODE:

```
def analyze_number(num):
    if num > 0:
        return "Positive number"
    elif num < 0:
        return "Negative number"
    else:
        return "Zero"


# ---- Main Program ----
numbers = []

count = int(input("How many numbers do you want to test? "))
```



```
for i in range(count):
    n = float(input(f"Enter number {i+1}: "))
    numbers.append(n)

print("\n--- Number Analysis Results ---")
for n in numbers:
    result = analyze_number(n)
    print(f"{n} → {result}")
```

OUTPUT:



```
How many numbers do you want to test? 3
Enter number 1: 10
Enter number 2: -5
Enter number 3: 0

--- Number Analysis Results ---
10.0 → Positive number
-5.0 → Negative number
0.0 → Zero
```

Prompt:

Create a function that determines whether a number is positive, negative, or zero using if-elif-else. Test with multiple inputs.

Explanation

AI properly used:

- `if-elif-else`
- **Covered all possible number categories**

Task Description-4 (Nested Conditionals)

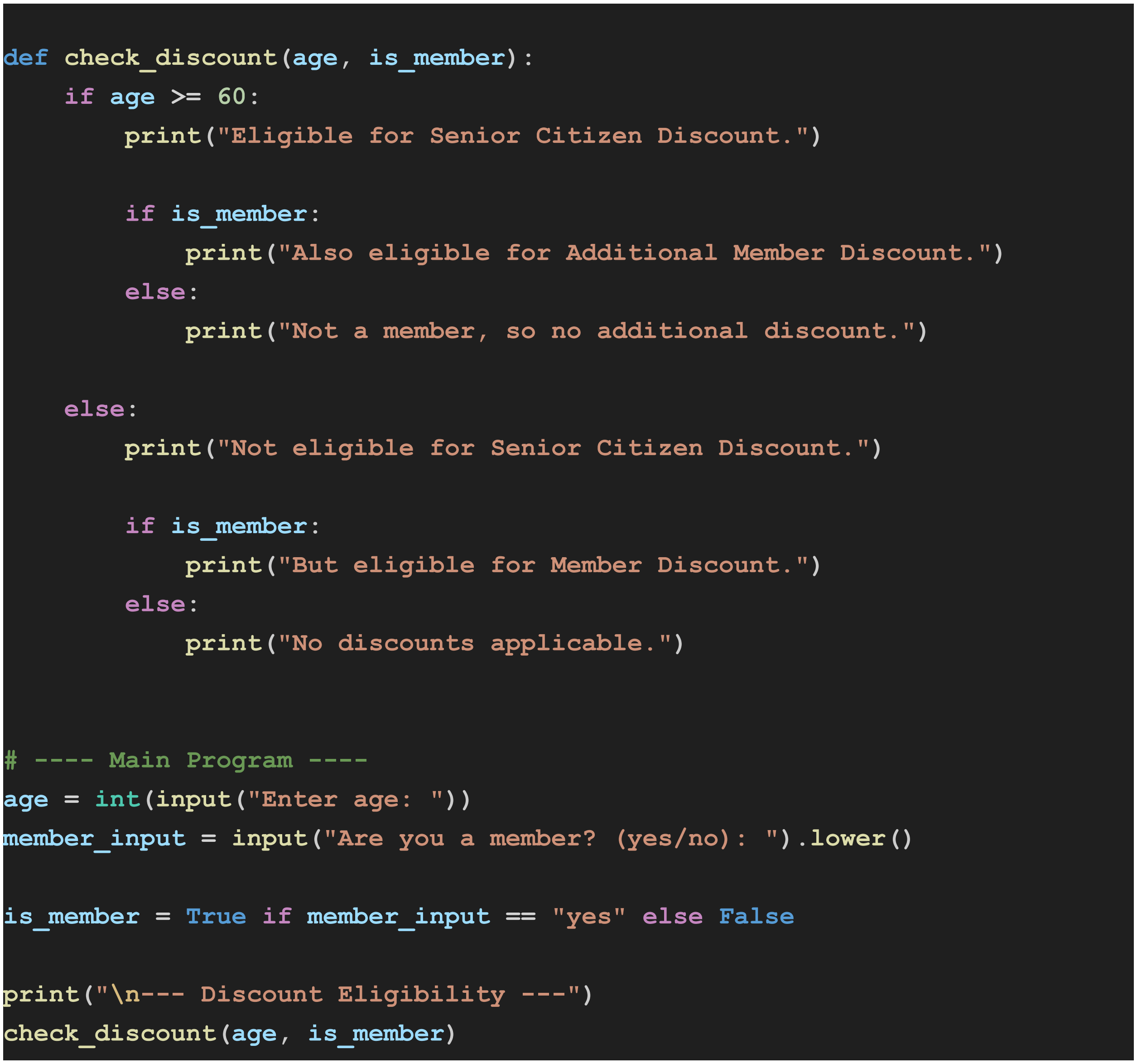
- **Generate a function check_discount(age, is_member) that determines discount eligibility:**
- **Age ≥ 60 → Senior discount**
- **Member → Additional discount**

Use nested if statements.

Expected Output-4

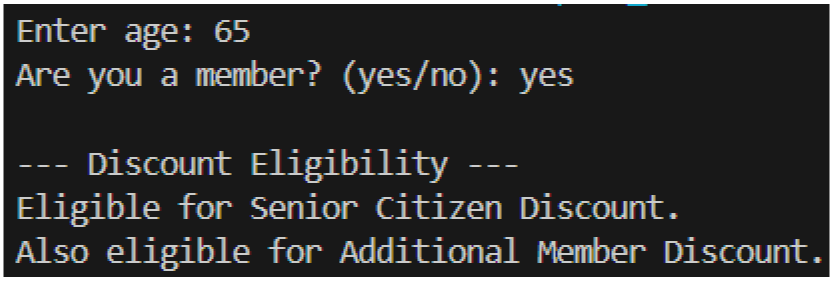
- **Python code using nested conditionals**
- **Clear explanation of decision flow**

CODE:



```python
def check_discount(age, is_member):
    if age >= 60:
        print("Eligible for Senior Citizen Discount.")

        if is_member:
            print("Also eligible for Additional Member Discount.")
        else:
            print("Not a member, so no additional discount.")

    else:
        print("Not eligible for Senior Citizen Discount.")

        if is_member:
            print("But eligible for Member Discount.")
        else:
            print("No discounts applicable.")


# ---- Main Program ----
age = int(input("Enter age: "))
member_input = input("Are you a member? (yes/no): ").lower()

is_member = True if member_input == "yes" else False

print("\n--- Discount Eligibility ---")
check_discount(age, is_member)
```

OUTPUT:



```
Enter age: 65
Are you a member? (yes/no): yes

--- Discount Eligibility ---
Eligible for Senior Citizen Discount.
Also eligible for Additional Member Discount.
```

Prompt:

> **Write a function `check_discount(age, is_member)` using nested if statements. Age ≥ 60 gets senior discount, members get additional discount.**

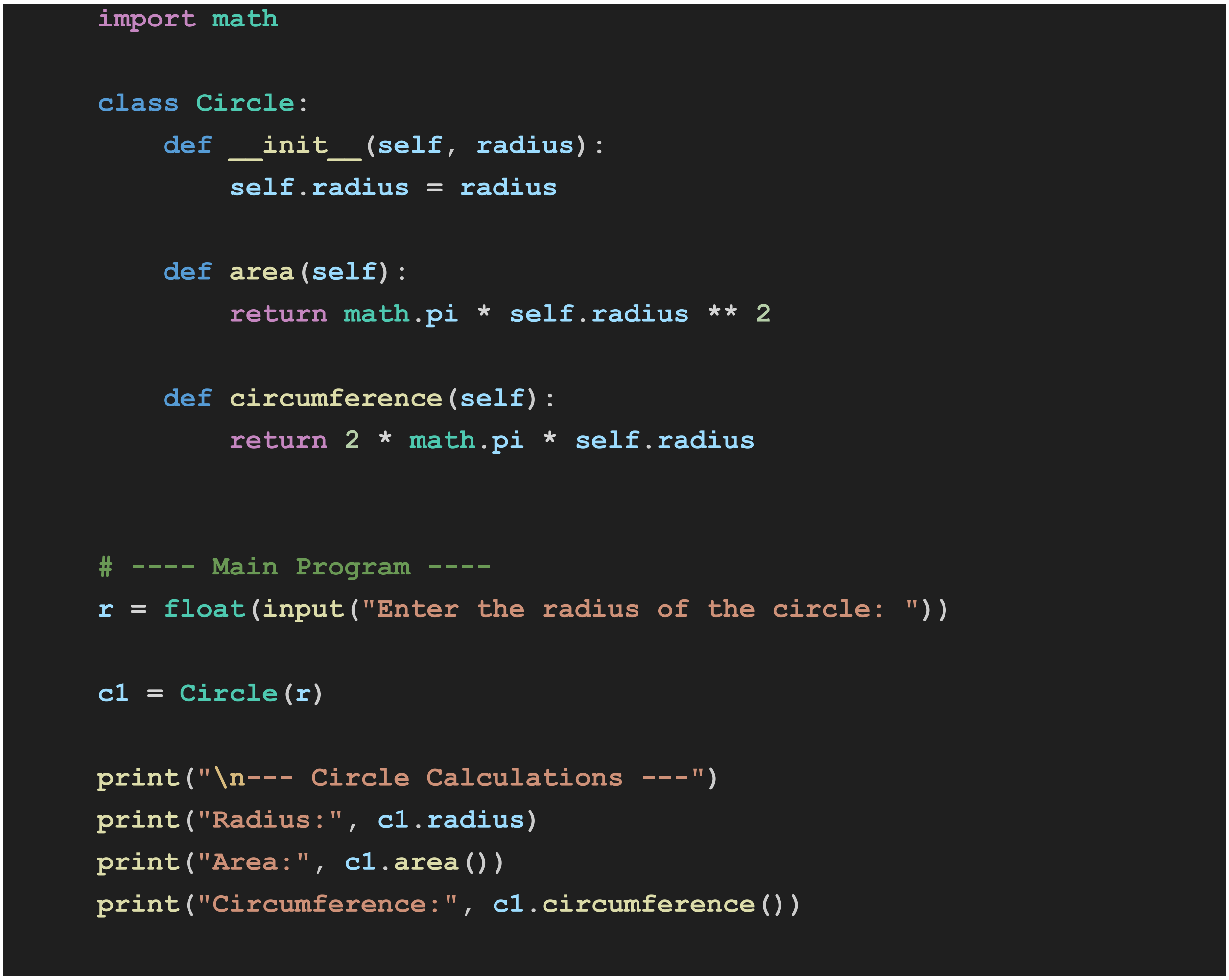
Explanation

AI:

- **Used nested `if` statements**
- **Applied different decisions inside each block**

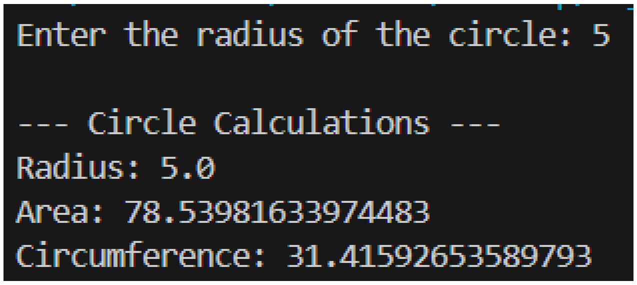
Task Description-5 (Class – Mathematical Opera)
• Ask AI to create a Circle class with methods to calculate area ()
and circumference () given the radius.
Expected Output-5
• Correct mathematical computation
• Well-structured class with methods
• Code explanation provided

CODE:



```python
import math

class Circle:
    def __init__(self, radius):
        self.radius = radius

    def area(self):
        return math.pi * self.radius ** 2

    def circumference(self):
        return 2 * math.pi * self.radius


# ---- Main Program ----
r = float(input("Enter the radius of the circle: "))

c1 = Circle(r)

print("\n--- Circle Calculations ---")
print("Radius:", c1.radius)
print("Area:", c1.area())
print("Circumference:", c1.circumference())
```

OUTPUT:



```
Enter the radius of the circle: 5

--- Circle Calculations ---
Radius: 5.0
Area: 78.53981633974483
Circumference: 31.41592653589793
```

Prompt:

Create a Circle class with methods to calculate area and circumference using the radius.

AI correctly:

- **Designed a class**
- **Used mathematical formulas**
- **Placed calculations inside methods**