

AI ASSISTED CODING

Name: P.Swaran Raj
Batch: 37

Roll.No: 2303A52247
Assignment- 6.2

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 \geq 40).

Prompt used:

“#generate a student class with these attributes name, roll.no, marks, and if marks \geq 40 return a pass() if marks <40 return fail().”

The screenshot displays the Google Colab AI Assist 6.2 interface. On the left, a code editor shows the following Python code:

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

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

    def fail(self):
        return self.marks < 40

# Example usage:
student1 = Student("Alice", "A001", 75)
student2 = Student("Bob", "B002", 35)

print(f"{student1.name} (Roll No: {student1.rollno}) marks: {student1.marks}, Passed: {student1.passed()}")
print(f"{student2.name} (Roll No: {student2.rollno}) marks: {student2.marks}, Passed: {student2.passed()}")
```

On the right, the Gemini AI chat window shows the following response:

P generate a student class with these attributes name, roll.no, marks, and if marks \geq 40 return a pass() if marks <40 return fail()

✦ Okay, I've updated the `Student` class to include a `fail()` method, which returns `True` if the student's marks are less than 40. I've also added an example usage for the new `fail()` method.

The `Student` class now correctly identifies if a student has passed or failed. Alice, with 75 marks, passed and did not fail. Bob, with 35 marks, did not pass and failed, as expected.

Below the chat window, there are several suggested prompts:

- Add a method to calculate the average marks of students
- Create a list to store multiple Student objects
- Explain how to use the Student class to manage a group of students

At the bottom of the chat window, there is a text input field "What can I help you build?" and a "Gemini 2.5 Flash" button.

Explanation:

The Student class stores student details using a constructor. The `is_pass()` method checks whether marks are greater than or equal to 40 and returns pass or fail status.

Output:

The method correctly returns Pass for marks ≥ 40 and Fail otherwise.

Alice (Roll No: A001) marks: 75, Passed: True, Failed: False

Bob (Roll No: B002) marks: 35, Passed: False, Failed: True

Observation:

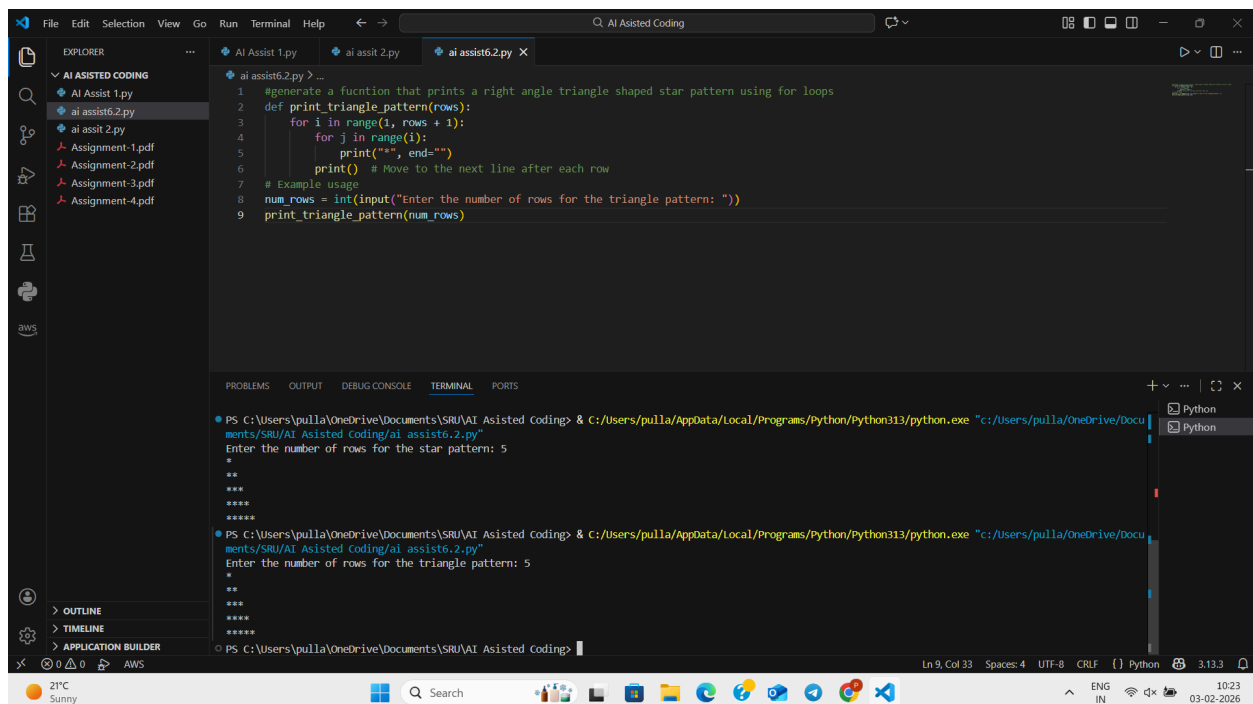
AI generated a clear and correct class structure. The validation logic is simple and easy to understand.

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.

Prompt used:

“#generate a function that prints a right angle triangle shaped star pattern using for loops”



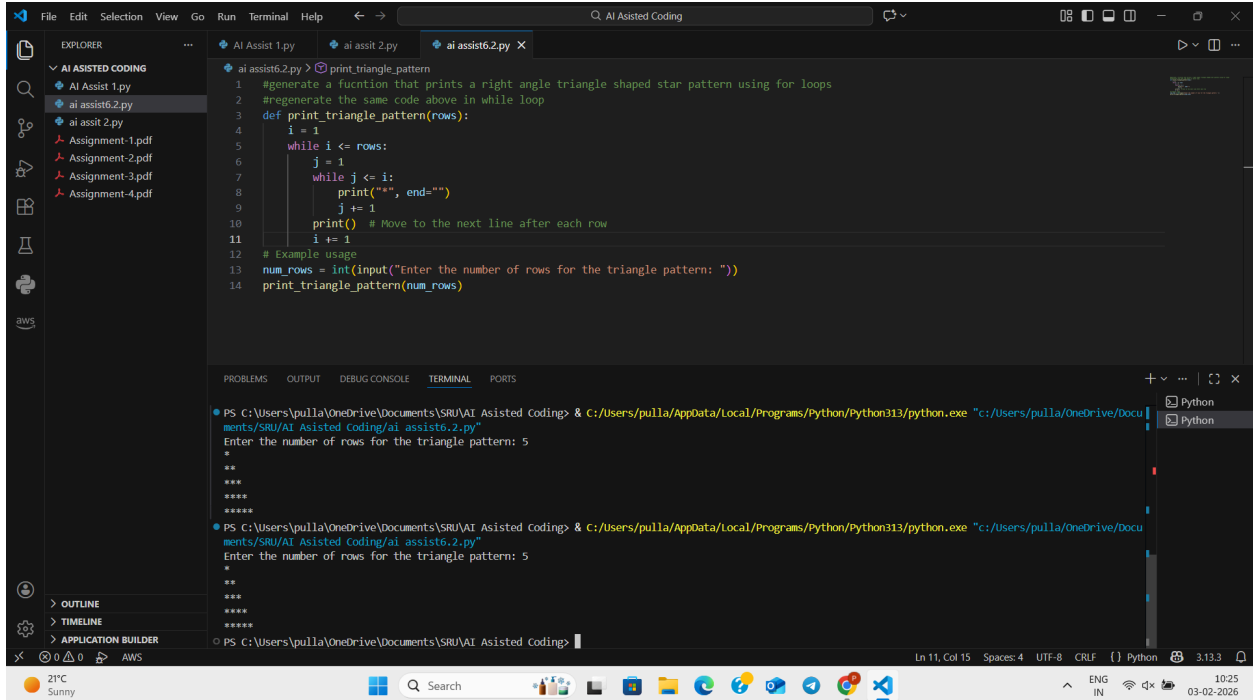
```
1 #generate a function that prints a right angle triangle shaped star pattern using for loops
2 def print_triangle_pattern(rows):
3     for i in range(1, rows + 1):
4         for j in range(i):
5             print("*", end="")
6             print() # Move to the next line after each row
7 # Example usage
8 num_rows = int(input("Enter the number of rows for the triangle pattern: "))
9 print_triangle_pattern(num_rows)
```

```
PS C:\Users\pulla\OneDrive\Documents\SRU\AI Asisted Coding> & C:/Users/pulla/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/pulla/OneDrive/Docu
ments\SRU\AI Asisted Coding\ai assist6.2.py"
Enter the number of rows for the star pattern: 5
*
**
***
****
*****

PS C:\Users\pulla\OneDrive\Documents\SRU\AI Asisted Coding> & C:/Users/pulla/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/pulla/OneDrive/Docu
ments\SRU\AI Asisted Coding\ai assist6.2.py"
Enter the number of rows for the triangle pattern: 5
*
**
***
****
*****
```

Prompt used:

“#regenerate the same code above in while loop.”



The screenshot shows a VS Code editor with a file explorer on the left containing 'AI ASSISTED CODING' and several PDF files. The main editor displays a Python script named 'ai assist6.2.py' with the following code:

```
1 #generate a fuction that prints a right angle triangle shaped star pattern using for loops
2 #regenerate the same code above in while loop
3 def print_triangle_pattern(rows):
4     i = 1
5     while i <= rows:
6         j = 1
7         while j <= i:
8             print("*", end="")
9             j += 1
10        print() # Move to the next line after each row
11        i += 1
12
13 # Example usage
14 num_rows = int(input("Enter the number of rows for the triangle pattern: "))
15 print_triangle_pattern(num_rows)
```

The terminal at the bottom shows the execution of the script, which prompts for the number of rows (5) and displays the following star pattern:

```
Enter the number of rows for the triangle pattern: 5
*
**
***
****
*****
```

Explanation:

The for loop increases stars row by row using iteration count.

The while loop achieves the same pattern by manually updating loop conditions.

Output:

Enter the number of rows for the triangle pattern: 5

```
*
**
***
****
*****
```

Observation:

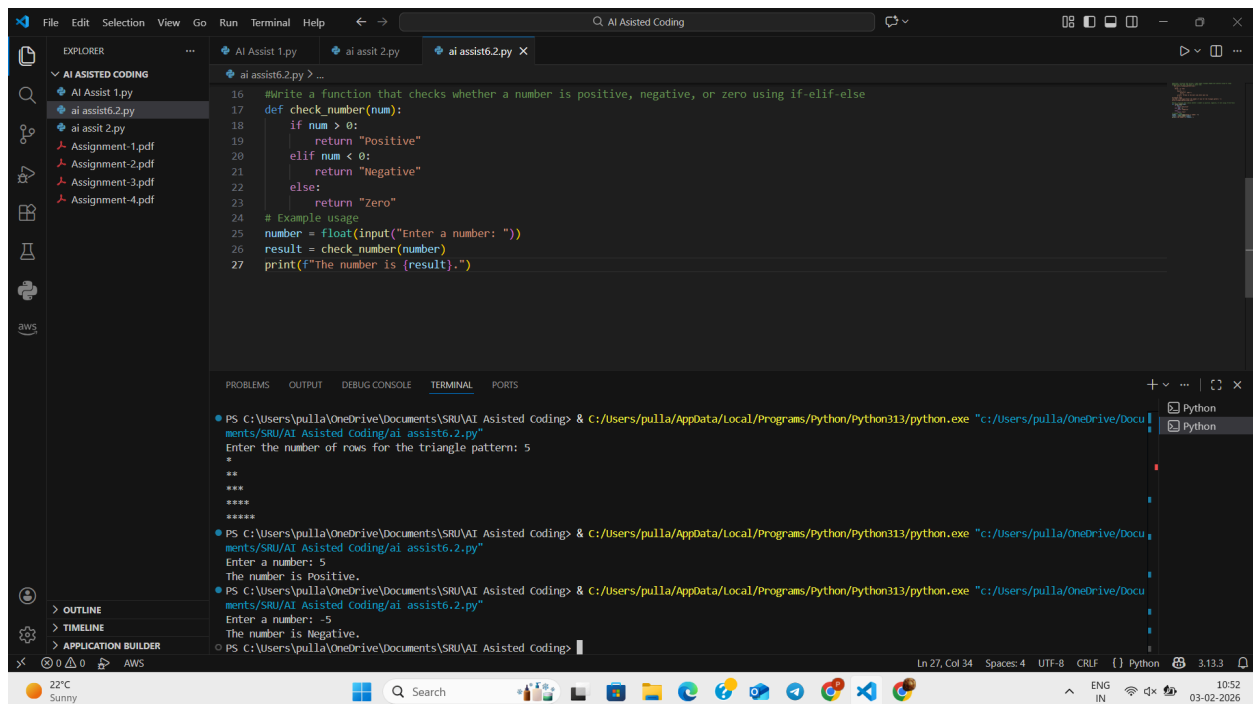
Both for and while loop implementation produced the same correct pattern
AI generated clear and correct looping logic for iterative pattern generation.

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.

Prompt used:

“#Write a function that checks whether a number is positive, negative, or zero using if-elif-else.”



The screenshot shows a Visual Studio Code editor with a Python file named `ai assist6.2.py`. The code defines a function `check_number(num)` that uses `if-elif-else` to classify a number as "Positive", "Negative", or "Zero". Below the function, there is an example usage where the user is prompted to enter a number, and the result is printed. The terminal at the bottom shows the execution of the script, with the user entering 5, -5, and 0, and the program outputting "The number is Positive.", "The number is Negative.", and "The number is Zero." respectively.

```
16 #write a function that checks whether a number is positive, negative, or zero using if-elif-else
17 def check_number(num):
18     if num > 0:
19         return "Positive"
20     elif num < 0:
21         return "Negative"
22     else:
23         return "Zero"
24 # Example usage
25 number = float(input("Enter a number: "))
26 result = check_number(number)
27 print(f"The number is {result}.")
```

Terminal Output:

```
PS C:\Users\pulla\OneDrive\Documents\SRU\AI Asisted Coding> & C:/Users/pulla/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/pulla/OneDrive/Docu
ments\SRU\AI Asisted Coding/ai assist6.2.py"
Enter the number of rows for the triangle pattern: 5
*
**
***
****
*****

PS C:\Users\pulla\OneDrive\Documents\SRU\AI Asisted Coding> & C:/Users/pulla/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/pulla/OneDrive/Docu
ments\SRU\AI Asisted Coding/ai assist6.2.py"
Enter a number: 5
The number is Positive.

PS C:\Users\pulla\OneDrive\Documents\SRU\AI Asisted Coding> & C:/Users/pulla/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/pulla/OneDrive/Docu
ments\SRU\AI Asisted Coding/ai assist6.2.py"
Enter a number: -5
The number is Negative.

PS C:\Users\pulla\OneDrive\Documents\SRU\AI Asisted Coding>
```

Explanation:

The function compares the input number with zero using conditional statements. Each condition correctly classifies the number.

Output:

Positive / Negative / Zero displayed based on input value.

Observation:

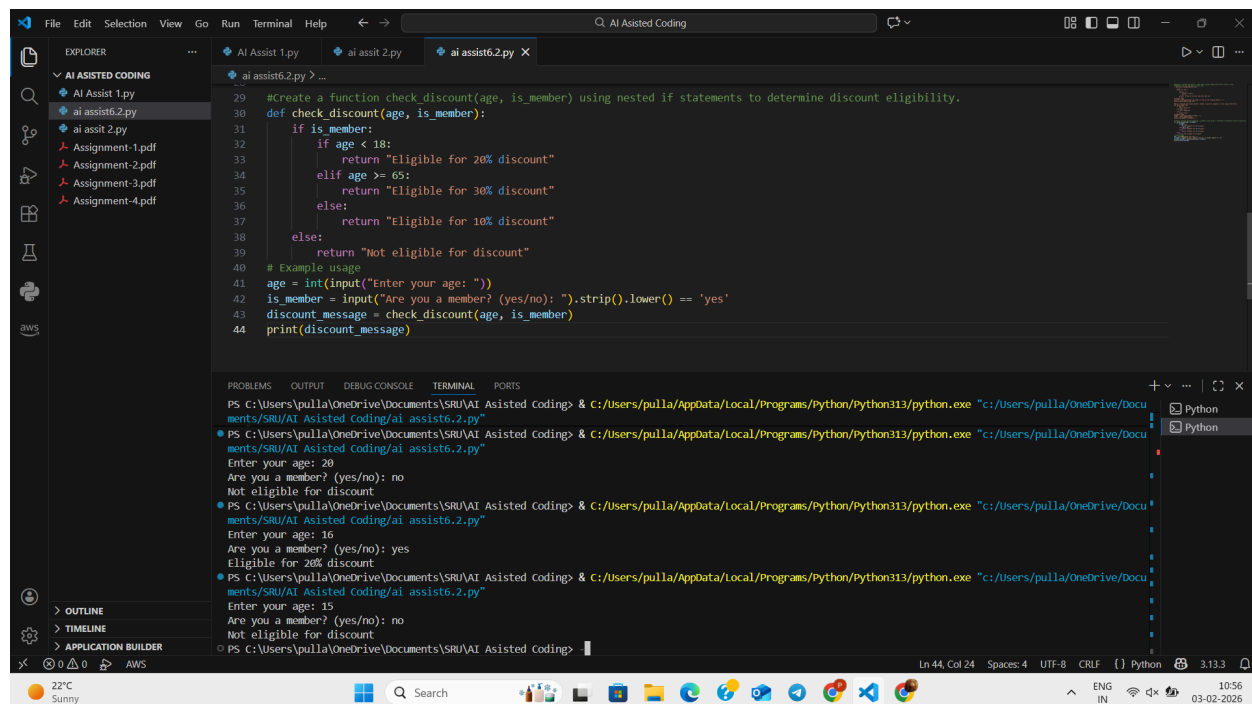
Decision logic is accurate and well-structured.
AI correctly handled all possible conditions.

Task Description-4 (Nested Conditionals)

Generate a function `check_discount(age, is_member)` that determines discount eligibility: Age $\geq 60 \rightarrow$ Senior discount, Member \rightarrow Additional discount, Use nested if statements.

Prompt used:

“#Create a function `check_discount(age, is_member)` using nested if statements to determine discount eligibility.”



The screenshot shows a Visual Studio Code editor window with a file explorer on the left and a terminal at the bottom. The file explorer shows a project named 'AI ASSISTED CODING' with files 'ai assist 1.py', 'ai assist 2.py', and 'ai assist6.2.py'. The main editor window displays the code for 'ai assist6.2.py'. The code defines a function `check_discount` with nested if statements to determine discount eligibility based on age and membership status. Below the code, the terminal shows the execution of the script, which prompts the user for age and membership status, and displays the resulting discount eligibility.

```
29 #Create a function check_discount(age, is_member) using nested if statements to determine discount eligibility.
30 def check_discount(age, is_member):
31     if is_member:
32         if age < 18:
33             return "Eligible for 20% discount"
34         elif age >= 65:
35             return "Eligible for 30% discount"
36         else:
37             return "Eligible for 10% discount"
38     else:
39         return "Not eligible for discount"
40 # Example usage
41 age = int(input("Enter your age: "))
42 is_member = input("Are you a member? (yes/no): ").strip().lower() == 'yes'
43 discount_message = check_discount(age, is_member)
44 print(discount_message)
```

Terminal Output:

```
PS C:\Users\pulla\OneDrive\Documents\SRU\AI Asisted Coding> & C:/Users/pulla/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/pulla/OneDrive/Docu
ments\SRU\AI Asisted Coding\ai assist6.2.py"
Enter your age: 20
Are you a member? (yes/no): no
Not eligible for discount.
PS C:\Users\pulla\OneDrive\Documents\SRU\AI Asisted Coding> & C:/Users/pulla/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/pulla/OneDrive/Docu
ments\SRU\AI Asisted Coding\ai assist6.2.py"
Enter your age: 16
Are you a member? (yes/no): yes
Eligible for 20% discount
PS C:\Users\pulla\OneDrive\Documents\SRU\AI Asisted Coding> & C:/Users/pulla/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/pulla/OneDrive/Docu
ments\SRU\AI Asisted Coding\ai assist6.2.py"
Enter your age: 15
Are you a member? (yes/no): no
Not eligible for discount.
PS C:\Users\pulla\OneDrive\Documents\SRU\AI Asisted Coding>
```

Code explanation:

The function first checks age for senior discount eligibility. Nested conditions check membership status for additional discounts.

Output:

Correct discount eligibility is displayed based on age and membership.

Observation:

Nested condition flow is clear and logically correct.

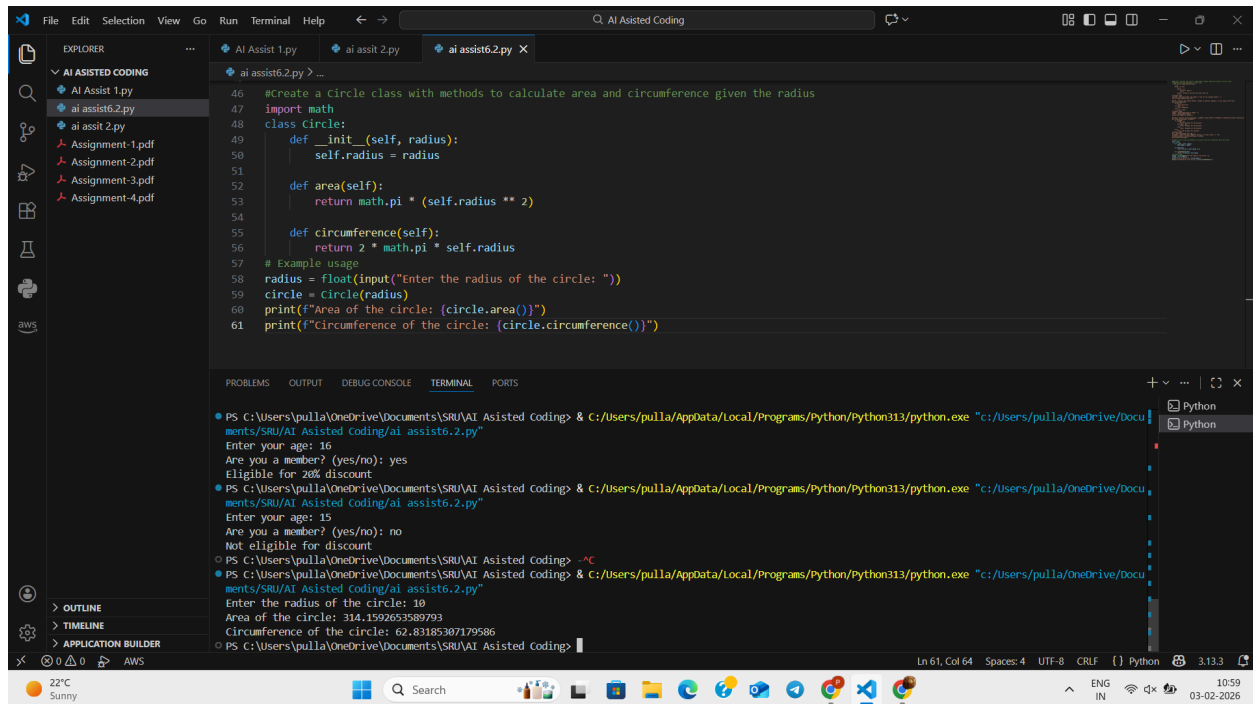
AI handled multiple conditions effectively.

Task Description-5 (Class – Mathematical Opera)

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

Prompt used:

“#Create a Circle class with methods to calculate area and circumference given the radius.”



The screenshot shows a Visual Studio Code editor with a Python file named `ai assist6.2.py`. The code defines a `Circle` class with an `__init__` method that takes a radius and stores it as a class attribute. It also includes `area` and `circumference` methods that use the stored radius to calculate the respective values. The code includes example usage where a radius is input by the user, a `Circle` object is created, and its area and circumference are printed.

```
46 #Create a Circle class with methods to calculate area and circumference given the radius
47 import math
48 class Circle:
49     def __init__(self, radius):
50         self.radius = radius
51
52     def area(self):
53         return math.pi * (self.radius ** 2)
54
55     def circumference(self):
56         return 2 * math.pi * self.radius
57
58 # Example usage
59 radius = float(input("Enter the radius of the circle: "))
60 circle = Circle(radius)
61 print(f"Area of the circle: {circle.area()}")
62 print(f"Circumference of the circle: {circle.circumference()}")
```

The terminal output shows the execution of the script. It prompts the user for their age and membership status, then for the radius of a circle. For a radius of 10, it calculates an area of approximately 314.16 and a circumference of approximately 62.83.

```
PS C:\Users\pulla\OneDrive\Documents\SRU\AI Asisted Coding> & C:/Users/pulla/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/pulla/OneDrive/Docu
ments/SRU/AI Asisted Coding/ai assist6.2.py"
Enter your age: 16
Are you a member? (yes/no): yes
Eligible for 20% discount
PS C:\Users\pulla\OneDrive\Documents\SRU\AI Asisted Coding> & C:/Users/pulla/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/pulla/OneDrive/Docu
ments/SRU/AI Asisted Coding/ai assist6.2.py"
Enter your age: 15
Are you a member? (yes/no): no
Not eligible for discount
PS C:\Users\pulla\OneDrive\Documents\SRU\AI Asisted Coding> -c
PS C:\Users\pulla\OneDrive\Documents\SRU\AI Asisted Coding> & C:/Users/pulla/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/pulla/OneDrive/Docu
ments/SRU/AI Asisted Coding/ai assist6.2.py"
Enter the radius of the circle: 10
Area of the circle: 314.1592653589793
Circumference of the circle: 62.83185307179586
PS C:\Users\pulla\OneDrive\Documents\SRU\AI Asisted Coding>
```

Explanation:

The Circle class uses radius as an attribute.

Area and circumference are calculated using standard mathematical formulas.

Output:

Correct area and circumference values are displayed for the given radius.

Observation:

AI generated accurate mathematical logic.

The class design is clean and reusable.

Conclusion:

This lab demonstrates how AI effectively assists in generating correct and well-structured Python code for classes, loops, and conditional logic.