

## AI ASSISTED CODING

Name: P.Swaran Raj

Roll.No: 2303A52247

Batch: 37

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 ( $\text{marks} \geq 40$ ).

#### **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 shows a browser window with two tabs open: "Lab Assignment 6.2" and "AI Assist 6.2 - Colab". The "AI Assist 6.2 - Colab" tab is active, displaying a code editor with the following Python code:

```
[4] 0s
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()}")
...
Alice (Roll No: A001) marks: 75, Passed: True, Failed: False
Bob (Roll No: B002) marks: 35, Passed: False, Failed: True
```

To the right of the code editor is the "Gemini" interface. It displays the user's prompt and the AI's 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()

Ok, 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 AI's response are several suggestions:

- 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

A message box at the bottom asks, "What can I help you build?" and provides 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  $\geq 40$  and Fail otherwise.

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

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

## Observation:

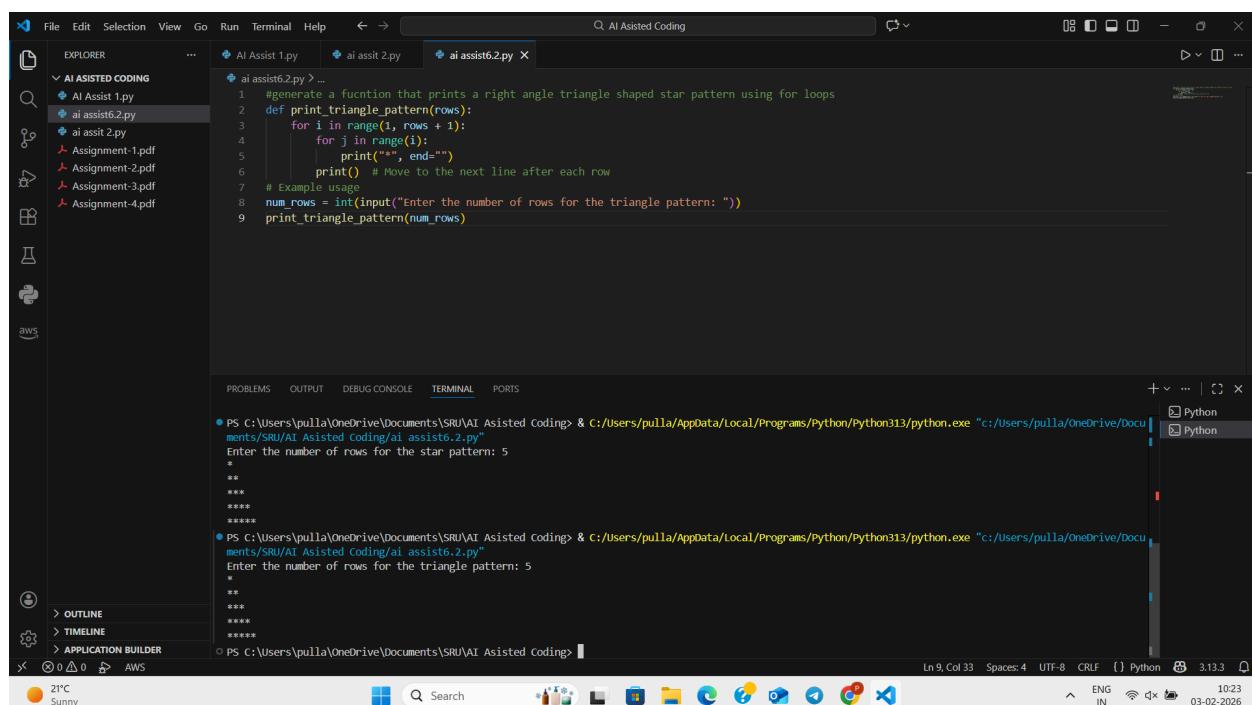
AI generated a clear and correct class structure. The validation logic is simple and easy to understand.

## 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"**



The screenshot shows a Microsoft Visual Studio Code (VS Code) interface. The top bar includes File, Edit, Selection, View, Go, Run, Terminal, Help, and a search bar. The title bar says "C:\AI Asisted Coding". The Explorer sidebar on the left shows files like "AI ASSISTED CODING", "Assignment-1.pdf", "Assignment-2.pdf", "Assignment-3.pdf", "Assignment-4.pdf", and Python files "ai assist 1.py", "ai assist 2.py", and "ai assist6.2.py". The main editor area contains the following Python code:

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

The terminal at the bottom shows two command-line sessions. The first session runs "ai assist6.2.py" and prints a triangle pattern for 5 rows. The second session runs "ai assist6.2.py" again and prints a triangle pattern for 5 rows. The status bar at the bottom right shows "Python" and "3.13.3".

## Prompt used:

"#regenerate the same code above in while loop."

The screenshot shows a Microsoft Visual Studio Code (VS Code) interface. The left sidebar has a tree view titled 'EXPLORER' under 'AI ASSISTED CODING' containing files like 'AI Assist 1.py', 'ai assist 2.py', and 'ai assist6.2.py'. The main editor area contains a Python script named 'ai assist6.2.py' with the following code:

```
#generate a function that prints a right angle triangle shaped star pattern using for loops
#regenerate the same code above in while loop
def print_triangle_pattern(rows):
    i = 1
    while i <= rows:
        j = 1
        while j <= i:
            print("*", end="")
            j += 1
        print() # Move to the next line after each row
        i += 1
# Example usage
num_rows = int(input("Enter the number of rows for the triangle pattern: "))
print_triangle_pattern(num_rows)
```

The terminal below shows two runs of the script. In the first run, it asks for '5' rows and prints:

```
*
```

In the second run, it also asks for '5' rows and prints:

```
*
```

The status bar at the bottom indicates the file is saved in Python, the line count is 11, and the column count is 15.

## 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:

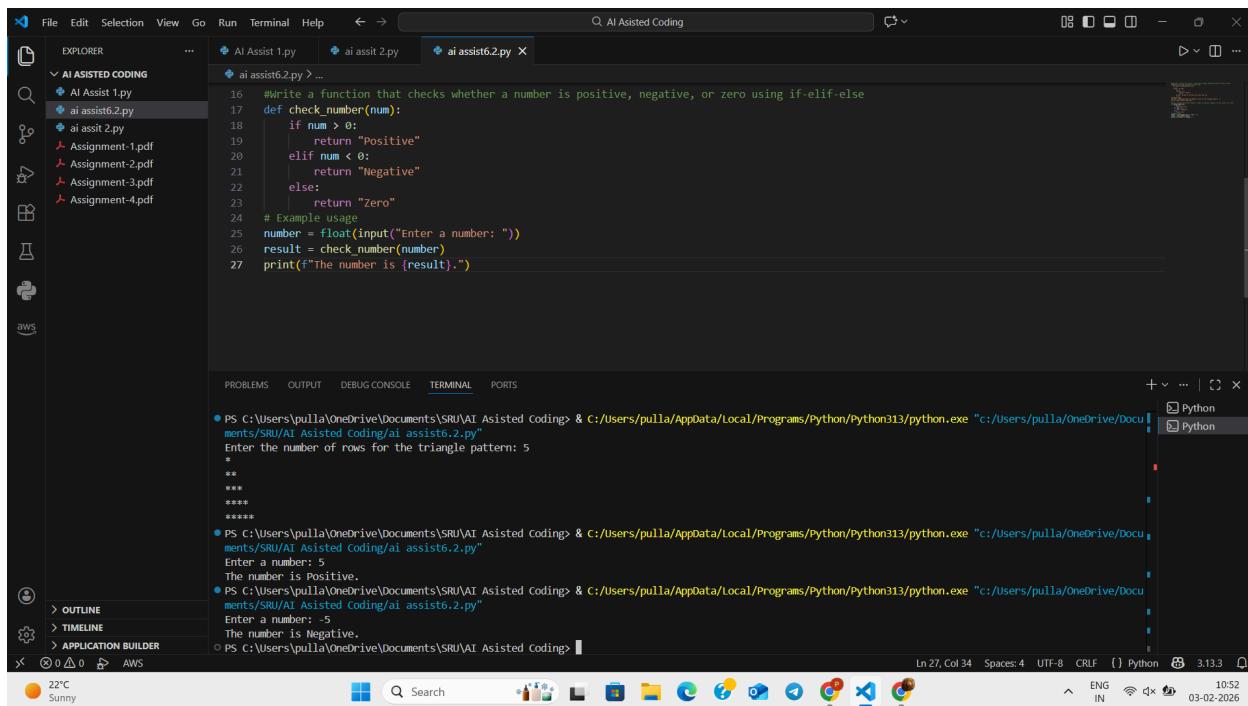
Both for and while loop implementation produced the same correct pattern  
AI generated clear and correct looping logic for iterative pattern generation.

## 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 the Visual Studio Code interface. The left sidebar has an 'EXPLORER' view with files like 'AI Assisted 1.py', 'ai assist 2.py', and 'ai assist6.2.py'. The main editor window contains Python code for a function that classifies numbers. The terminal at the bottom shows three runs of the script with user input and output. The status bar at the bottom right shows the date and time.

```
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}.")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\pulla\OneDrive\Documents\SRU\AI Assisted Coding> & C:/Users/pulla/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/pulla/OneDrive/Documents\SRU\AI Assisted Coding\ai assist6.2.py"
Enter the number for the triangle pattern: 5
*
**
***
****
*****
PS C:\Users\pulla\OneDrive\Documents\SRU\AI Assisted Coding> & C:/Users/pulla/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/pulla/OneDrive/Documents\SRU\AI Assisted Coding\ai assist6.2.py"
Enter a number: 5
The number is Positive.
PS C:\Users\pulla\OneDrive\Documents\SRU\AI Assisted Coding> & C:/Users/pulla/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/pulla/OneDrive/Documents\SRU\AI Assisted Coding\ai assist6.2.py"
Enter a number: -5
The number is Negative.
PS C:\Users\pulla\OneDrive\Documents\SRU\AI Assisted Coding>
```

22°C Sunny 22°C 10:52 ENG IN 03-02-2026

### 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 the Visual Studio Code interface. The code editor displays a Python script named `ai assist6.2.py`. The script defines a function `check_discount` that takes age and is\_member as parameters. It uses nested if statements to check for senior discount (age >= 60) and additional discount (is\_member). The terminal below shows the execution of the script and its output, which includes user prompts for age and membership status, and the resulting discount message. The status bar at the bottom indicates the file is 3.13.3 and the date is 03-02-2026.

```
#Create a function check_discount(age, is_member) using nested if statements to determine discount eligibility.
def check_discount(age, is_member):
    if is_member:
        if age < 18:
            return "Eligible for 20% discount"
        elif age >= 65:
            return "Eligible for 30% discount"
        else:
            return "Eligible for 10% discount"
    else:
        return "Not eligible for discount"

age = int(input("Enter your age: "))
is_member = input("Are you a member? (yes/no): ").strip().lower() == 'yes'
discount_message = check_discount(age, is_member)
print(discount_message)
```

### 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:

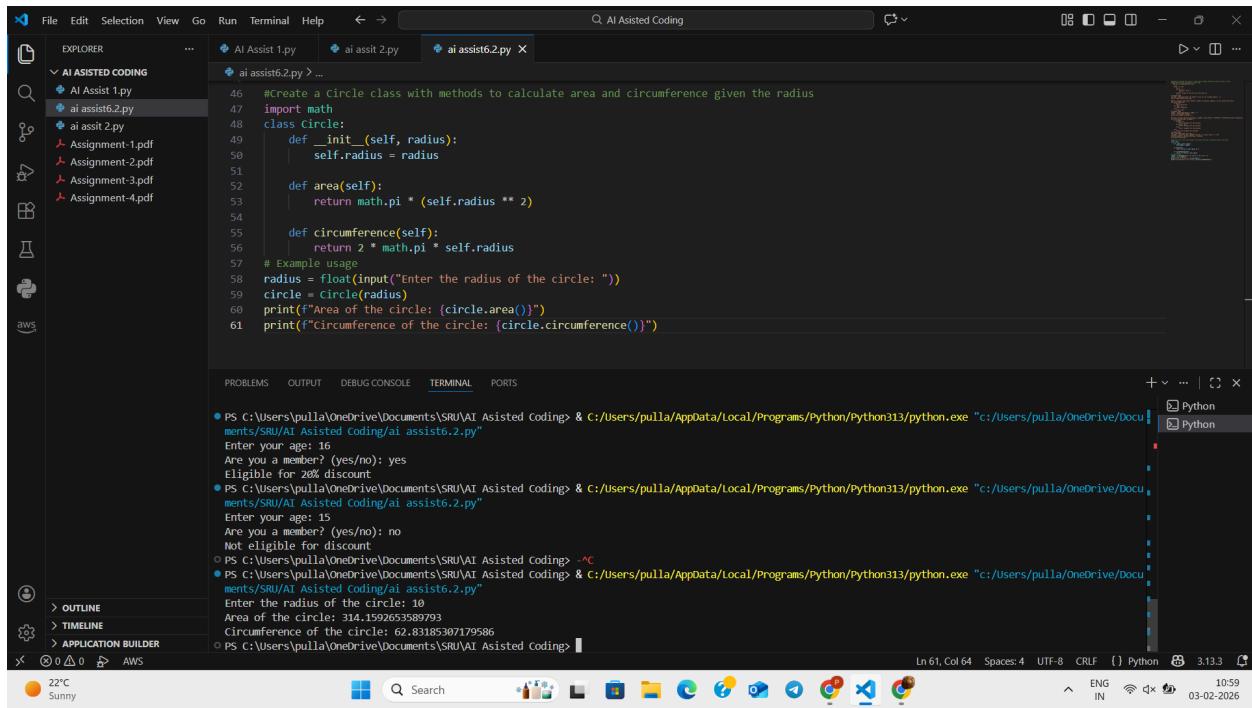
Nested condition flow is clear and logically correct.  
AI handled multiple conditions effectively.

## 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 code editor interface with the following details:

- File Explorer:** Shows files like "AI Assist 1.py", "ai assist 2.py", and "ai assist6.2.py".
- Code Editor:** Displays Python code for a Circle class:

```
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 # Example usage
58 radius = float(input("Enter the radius of the circle: "))
59 circle = Circle(radius)
60 print("Area of the circle: {circle.area()}")
61 print("Circumference of the circle: {circle.circumference()}")
```
- Terminal:** Shows command-line interaction:
  - PS C:\Users\pulla\OneDrive\Documents\SRU\AI Assisted Coding & C:/Users/pulla/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/pulla/OneDrive/Documents/SRU/AI Assisted 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 Assisted Coding & C:/Users/pulla/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/pulla/OneDrive/Documents/SRU/AI Assisted 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 Assisted Coding & C:/Users/pulla/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/pulla/OneDrive/Documents/SRU/AI Assisted Coding/ai assist6.2.py"
  - Enter the radius of the circle: 10
  - Area of the circle: 314.1592653589793
  - Circumference of the circle: 62.8318530717958
  - PS C:\Users\pulla\OneDrive\Documents\SRU\AI Assisted Coding>
- Status Bar:** Shows "Ln 61, Col 64" and "3.13.3".

### 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.