

2303A51739

BATCH-25

Assignment – 6.3

Task Description #1: Classes (Student Class)

Scenario

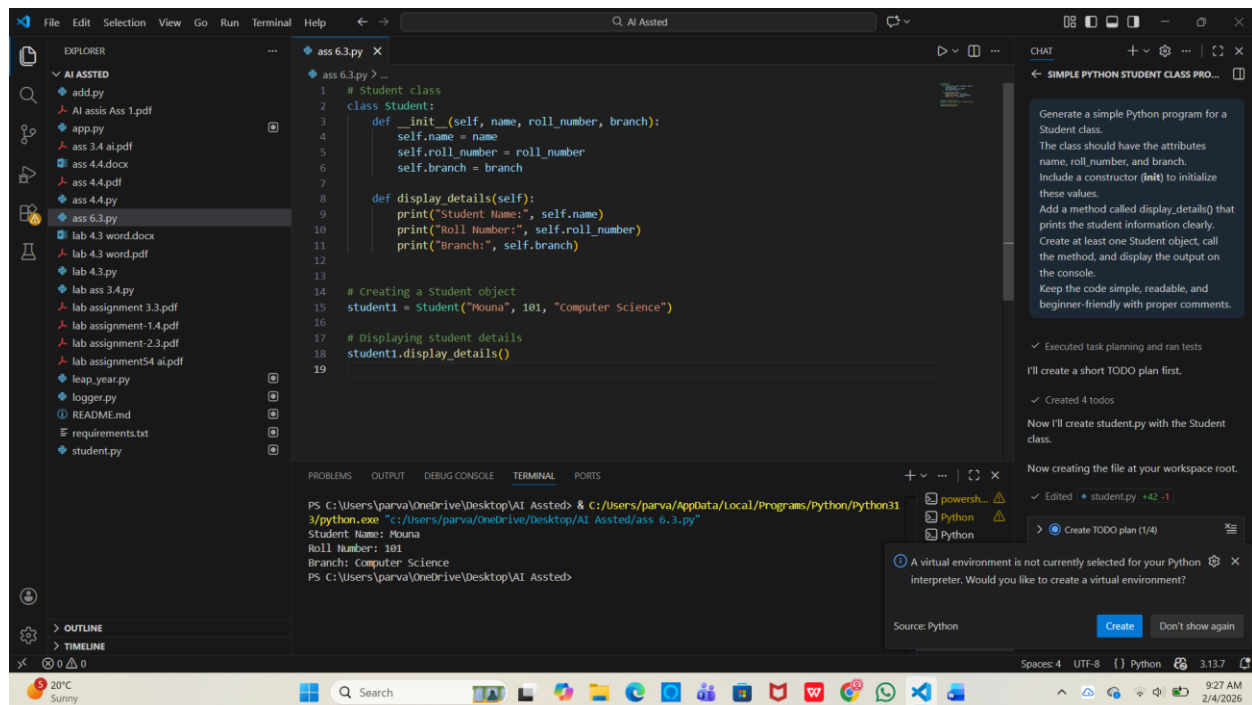
You are developing a simple student information management module.

Task

- Use an AI tool (GitHub Copilot / Cursor AI / Gemini) to complete a Student class.
- The class should include attributes such as name, roll number, and branch.
- Add a method `display_details()` to print student information.
- Execute the code and verify the output.
- Analyze the code generated by the AI tool for correctness and clarity.

Expected Output #1

- A Python class with a constructor (`__init__`) and a `display_details()` method.
- Sample object creation and output displayed on the console.
- Brief analysis of AI-generated code.



Task Description #2: Loops (Multiples of a Number)

Scenario

You are writing a utility function to display multiples of a given number.

Task

- Prompt the AI tool to generate a function that prints the first 10 multiples of a given number using a loop.
- Analyze the generated loop logic.
- Ask the AI to generate the same functionality using another controlled looping structure (e.g., while instead of for).

Expected Output #2

- Correct loop-based Python implementation.
- Output showing the first 10 multiples of a number.
- Comparison and analysis of different looping approaches.

The screenshot shows the Visual Studio Code editor interface. The Explorer panel on the left lists files in a project named 'AI ASSTED'. The main editor window displays a Python file named 'ass 6.3.py' with the following code:

```
19 # Using for loop
20
21 # Using for loop
22 def print_multiples_for(n):
23     for i in range(1, 11):
24         print(n * i)
25
26 print("Using for loop:")
27 print_multiples_for(5)
28
29 # Using while loop
30
31 def print_multiples_while(n):
32     i = 1
33     while i <= 10:
34         print(n * i)
35         i += 1
36
37 print("\nUsing while loop:")
38 print_multiples_while(5)
39
```

The Terminal panel at the bottom shows the output of the script:

```
Using for loop:
5
10
15
20
25
30
35
40
45
50
```

The Chat panel on the right contains a conversation with an AI assistant. The assistant provides instructions on how to use the code and offers to create a short TODO plan.

This screenshot shows the same Visual Studio Code editor interface as the previous one, but the terminal output has been updated to show the results of the while loop:

```
Using while loop:
5
10
15
20
25
30
35
40
45
50
```

The Chat panel on the right remains the same, showing the AI assistant's previous response.

Task Description #3: Conditional Statements (Age Classification)

Scenario

You are building a basic classification system based on age.

Task

- Ask the AI tool to generate nested if-elif-else conditional statements to classify age groups

(e.g., child, teenager, adult, senior).

- Analyze the generated conditions and logic.
- Ask the AI to generate the same classification using alternative conditional structures (e.g., simplified conditions or dictionary-based logic).

Expected Output #3

- A Python function that classifies age into appropriate groups.
- Clear and correct conditional logic.
- Explanation of how the conditions work.

The screenshot shows a VS Code editor with a file explorer on the left containing various files like 'add.py', 'age_classifier.py', and 'ass 6.3.py'. The main editor window displays a Python script named 'ass 6.3.py' with the following code:

```
36
37 print("\nUsing while loop:")
38 print_multiples_while(5)"""
39
40
41 def classify_age(age):
42     if age < 13:
43         print("Child")
44     elif age >= 13 and age <= 19:
45         print("Teenager")
46     elif age >= 20 and age <= 59:
47         print("Adult")
48     else:
49         print("Senior")
50
51
52 # Example usage
53 classify_age(25) # Output: Adult
54
```

The terminal at the bottom shows the execution of the script, outputting 'Teenager' and 'Adult' for different age inputs. On the right, a chat window titled 'SIMPLE PYTHON STUDENT CLASS PRO...' contains a prompt: 'Generate a Python function that classifies a person based on age using nested if-elif-else statements. The age groups should be: Child (below 13), Teenager (13-19), Adult (20-59), Senior (60 and above). The function should take age as input and print the corresponding age group. Keep the code simple and beginner-friendly.' The chat also shows a response from the AI assistant.

Task Description #4: For and While Loops (Sum of First n Numbers)

Scenario

You need to calculate the sum of the first n natural numbers.

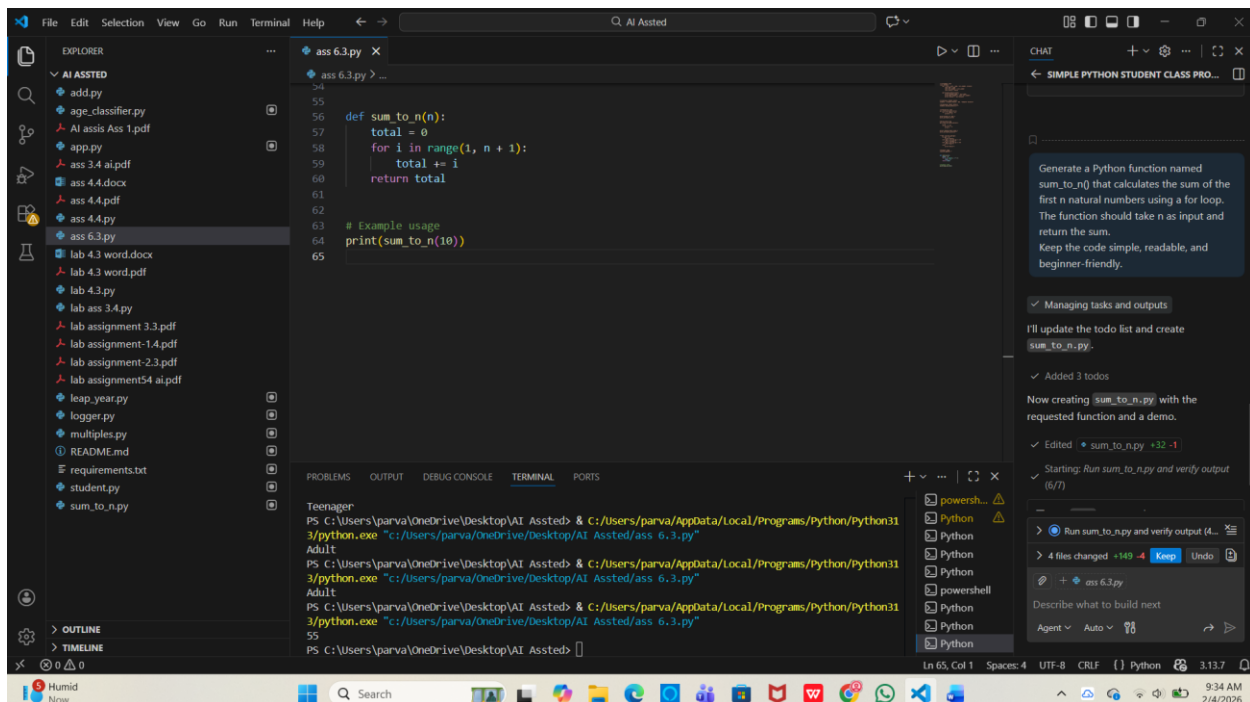
Task

- Use AI assistance to generate a `sum_to_n()` function using a for loop.
- Analyze the generated code.
- Ask the AI to suggest an alternative implementation using a while loop or a mathematical

formula.

Expected Output #4

- Python function to compute the sum of first n numbers.
- Correct output for sample inputs.
- Explanation and comparison of different approaches.



ask Description #5: Classes (Bank Account Class)

Scenario

You are designing a basic banking application.

Task

- Use AI tools to generate a Bank Account class with methods such as deposit(), withdraw(), and check_balance().
- Analyze the AI-generated class structure and logic.
- Add meaningful comments and explain the working of the code.

Expected Output #5

- Complete Python Bank Account class.

- Demonstration of deposit and withdrawal operations with updated balance.
- Well-commented code with a clear explanation.

