Name: Bhuddewar Arun Roll: 2303A51735 Batch: 25

SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE	DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

Program Name: B. Tech	Assignment Type: Lab	Academic Year:2025-2026
Course Coordinator Name	Dr. Rishabh Mittal	
Instructor(s) Name	Mr. S Naresh Kumar Ms. B. Swathi Dr. Sasanko Shekhar Gantayat Mr. Md Sallauddin Dr. Mathivanan Mr. Y Srikanth Ms. N Shilpa Dr. Rishabh Mittal (Coordinator) Dr. R. Prashant Kumar Mr. Ankushavali MD Mr. B Viswanath Ms. Sujitha Reddy Ms. A. Anitha Ms. M.Madhuri Ms. Katherashala Swetha Ms. Velpula sumalatha Mr. Bingi Raju	

Course Code	23CS002PC304	Course Title	AI Assisted Coding
Year/Sem	III/II	Regulation	R23
Date and Day of Assignment	Week3 – Wednesday	Time(s)	23CSBTB01 To 23CSBTB52
Duration	2 Hours	Applicable to Batches	All batches

AssignmentNumber:6.3(Present assignment number)/**24**(Total number of assignments)

Q.No.	Question	*Expected Time to complete*
1	**Lab 6: AI-Based Code Completion – Classes, Loops, and Conditionals** **Lab Objectives** • To explore AI-powered auto-completion features for core Python constructs such as classes, loops, and conditional statements. • To analyze how AI tools suggest logic for object-oriented programming and control structures.	Week3 - Wednesday

• To evaluate the correctness, readability, and completeness of AI-generated Python code.

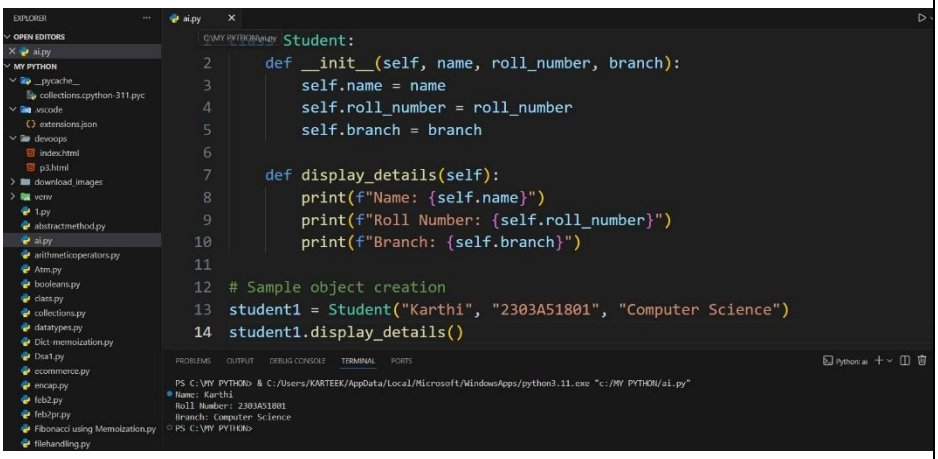
Lab Outcomes (LOs)
After completing this lab, students will be able to:
• Use AI tools to generate and complete Python class definitions and methods.
• Understand and assess AI-suggested loop constructs for iterative tasks.
• Generate and evaluate conditional statements using AI-driven prompts.
• Critically analyze AI-assisted code for correctness, clarity, and efficiency.

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
•



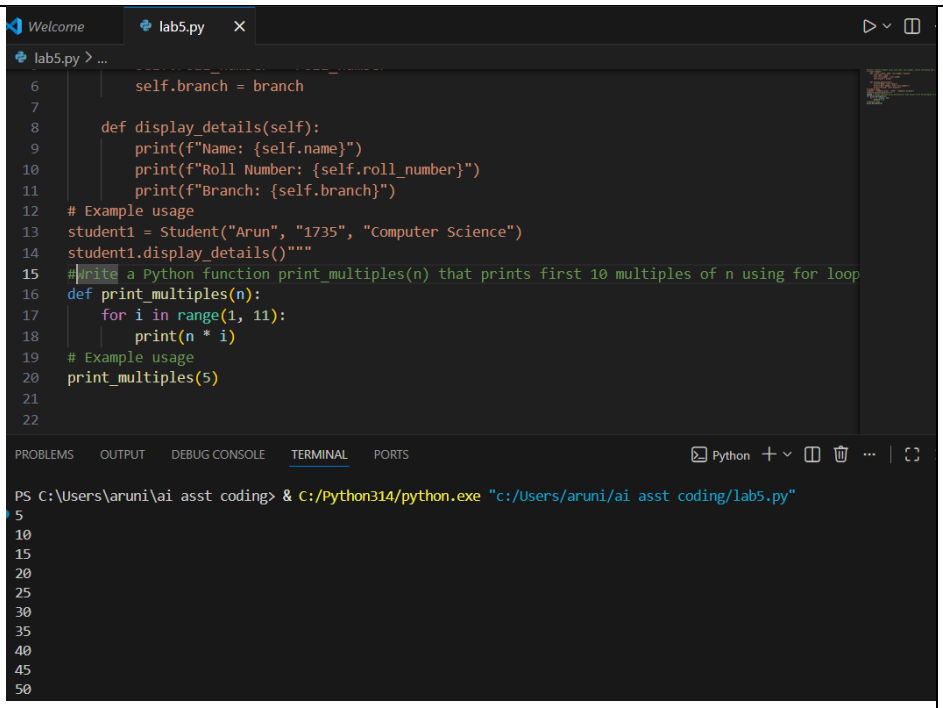
Task Description #2: Loops (Multiples of a Number)

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

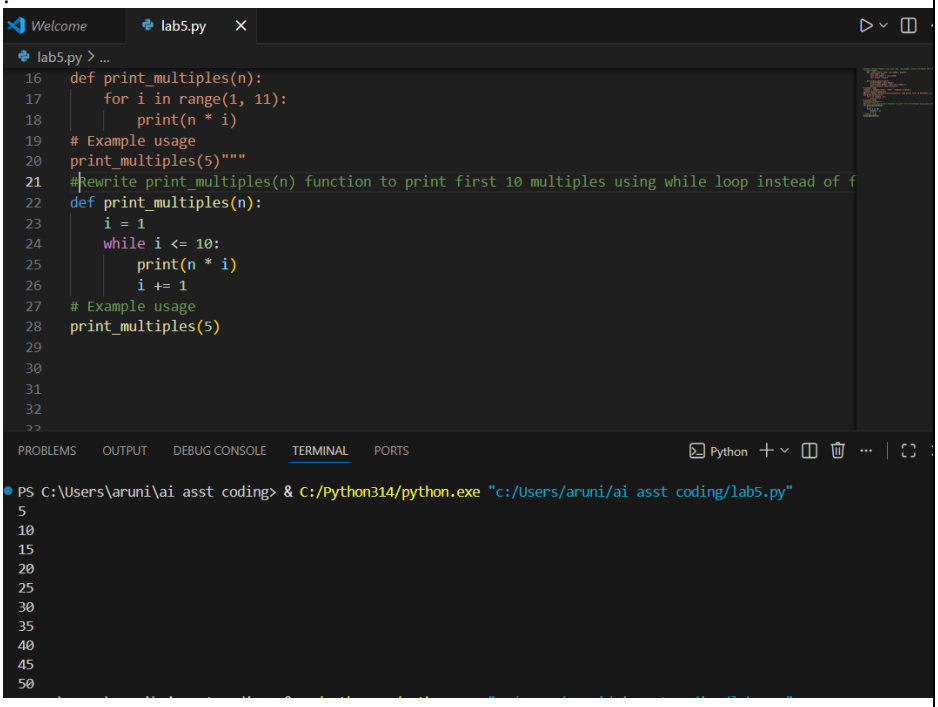
Task
• Prompt the AI tool to generate a function that prints the first 10 multiples of a given number using a loop.
• Analyze the generated loop logic.
• Ask the AI to generate the same functionality using another controlled looping structure (e.g., while instead of for).

Expected Output #2

Name: Bhuddewar Arun Roll: 2303A51735 Batch: 25



.



Task Description #3: Conditional Statements (Age Classification)

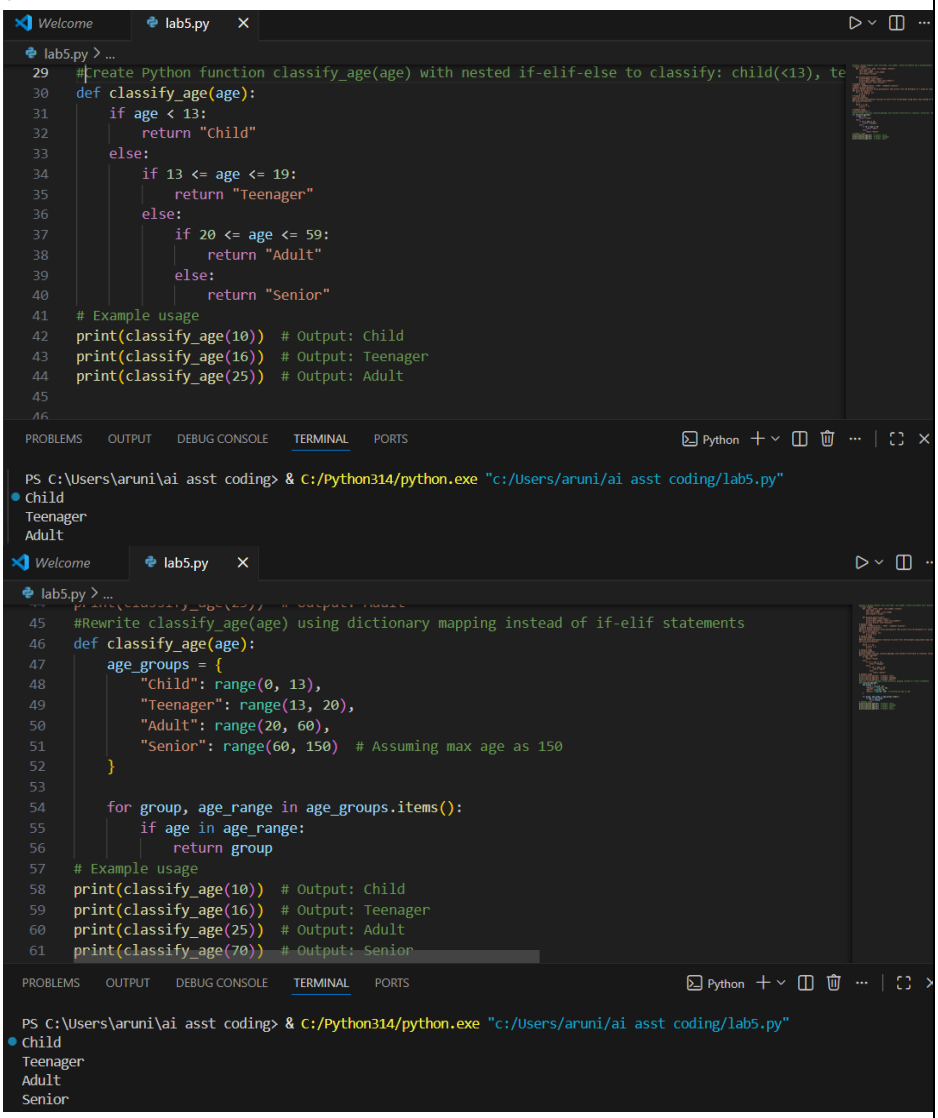
Scenario
You are building a basic classification system based on age.

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

•



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

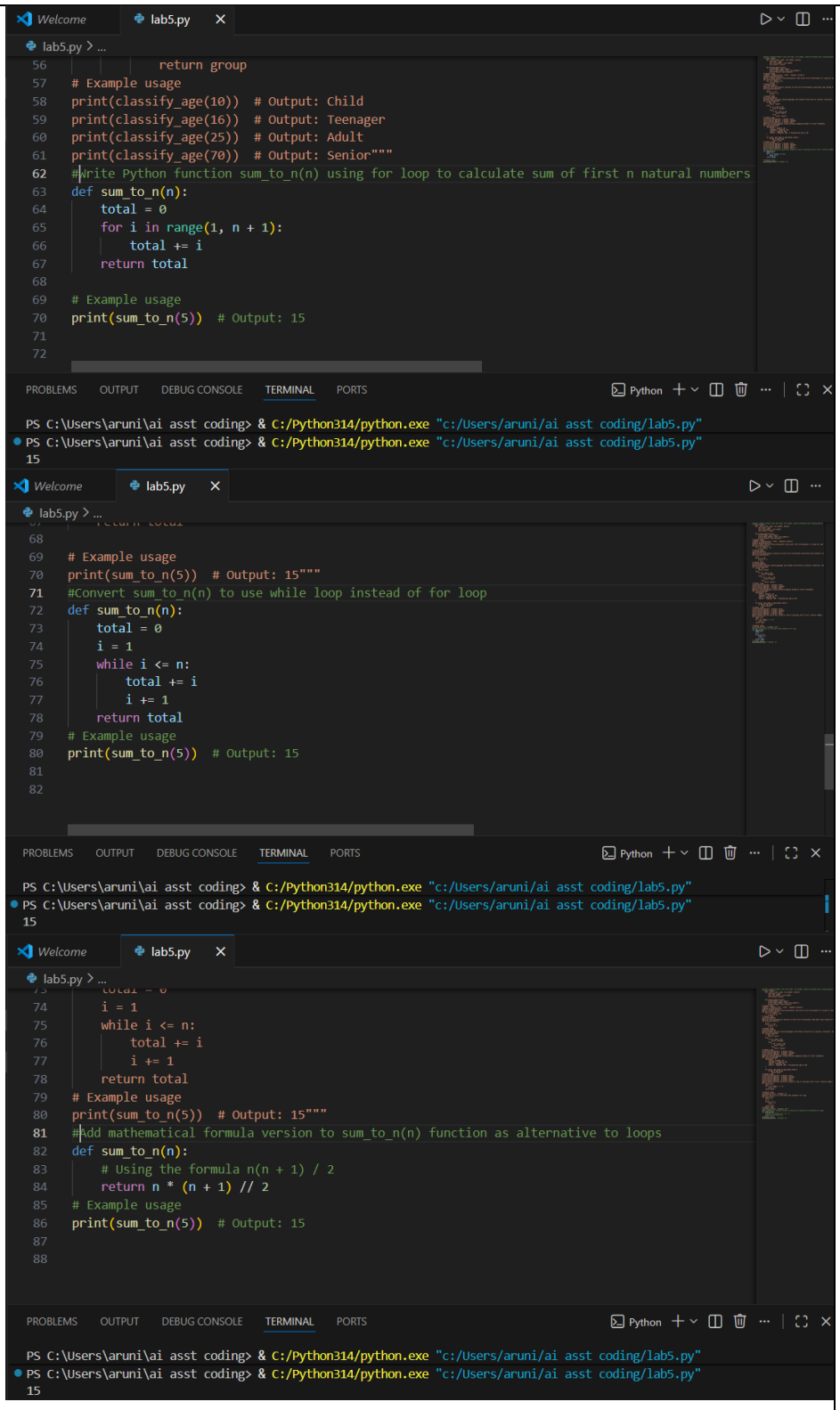
Scenario
You need to calculate the sum of the first n natural numbers.

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

•

Name: Bhuddewar Arun Roll: 2303A51735 Batch: 25



```
Welcome  lab5.py
lab5.py > ...
            return group
# Example usage
print(classify_age(10))  # Output: Child
print(classify_age(16))  # Output: Teenager
print(classify_age(25))  # Output: Adult
print(classify_age(70))  # Output: Senior"""
#Write Python function sum_to_n(n) using for loop to calculate sum of first n natural numbers
def sum_to_n(n):
    total = 0
    for i in range(1, n + 1):
        total += i
    return total

# Example usage
print(sum_to_n(5))  # Output: 15


PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  Python
PS C:\Users\aruni\ai asst coding> & C:/Python314/python.exe "c:/Users/aruni/ai asst coding/lab5.py"
PS C:\Users\aruni\ai asst coding> & C:/Python314/python.exe "c:/Users/aruni/ai asst coding/lab5.py"
15
```

```
Welcome  lab5.py
lab5.py > ...


# Example usage
print(sum_to_n(5))  # Output: 15"""
#Convert sum_to_n(n) to use while loop instead of for loop
def sum_to_n(n):
    total = 0
    i = 1
    while i <= n:
        total += i
        i += 1
    return total
# Example usage
print(sum_to_n(5))  # Output: 15


PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  Python
PS C:\Users\aruni\ai asst coding> & C:/Python314/python.exe "c:/Users/aruni/ai asst coding/lab5.py"
PS C:\Users\aruni\ai asst coding> & C:/Python314/python.exe "c:/Users/aruni/ai asst coding/lab5.py"
15
```

```
Welcome  lab5.py
lab5.py > ...
    i = 1
    while i <= n:
        total += i
        i += 1
    return total
# Example usage
print(sum_to_n(5))  # Output: 15"""
#Add mathematical formula version to sum_to_n(n) function as alternative to loops
def sum_to_n(n):
    # Using the formula n(n + 1) / 2
    return n * (n + 1) // 2
# Example usage
print(sum_to_n(5))  # Output: 15


PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  Python
PS C:\Users\aruni\ai asst coding> & C:/Python314/python.exe "c:/Users/aruni/ai asst coding/lab5.py"
PS C:\Users\aruni\ai asst coding> & C:/Python314/python.exe "c:/Users/aruni/ai asst coding/lab5.py"
15
```

.

Task Description #5: Classes (Bank Account Class)

Scenario

Name: Bhuddewar Arun Roll: 2303A51735 Batch: 25

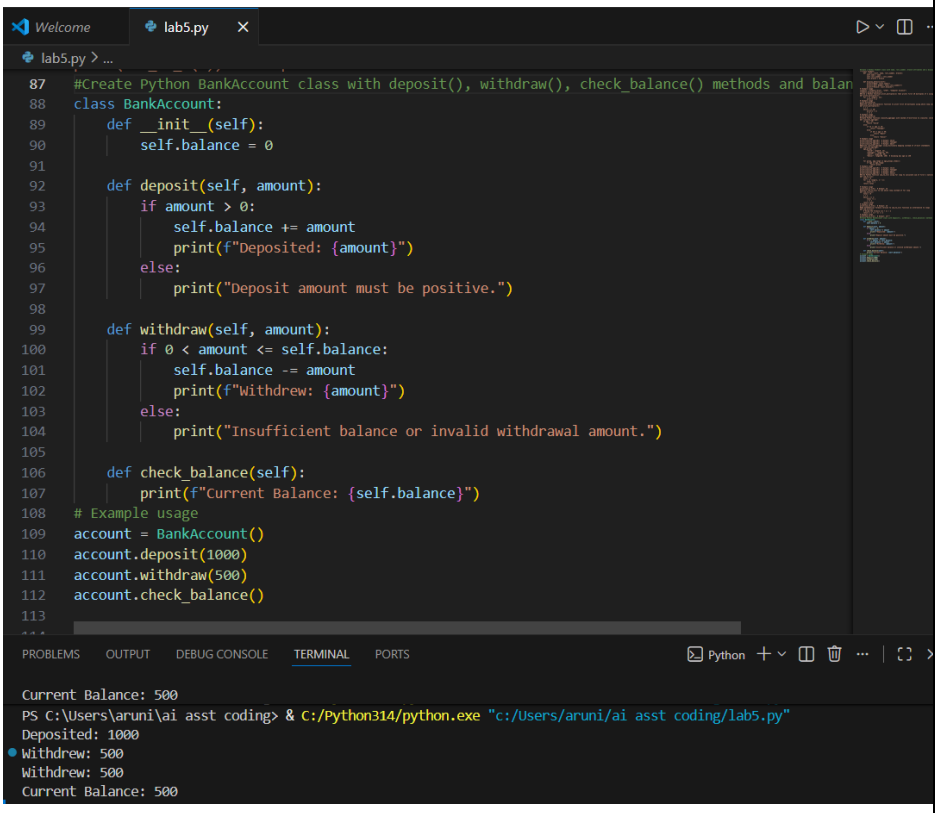
You are designing a basic banking application.

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

•



.

Note: Report should be submitted as a word document for all tasks in a single document with prompts, comments & code explanation, and output and if required, screenshots.