School of Computer Science and Artificial Intelligence

Lab Assignment-6.3

Course Title	**: AI Assistant Coding**
Name of Student	**: Shaik Naved Ahmed**
Enrollment No.	**: 2303A54053**
Batch No.	**: 47-B**

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.

Prompt:

Generate a Python class named Student with attributes name, roll number, and branch.
Include a constructor (__init__) to initialize these attributes and a method display_details() to print the student information.
Create a sample Student object, execute the code, and display the output.
Finally, provide a brief analysis of the code focusing on correctness and clarity.

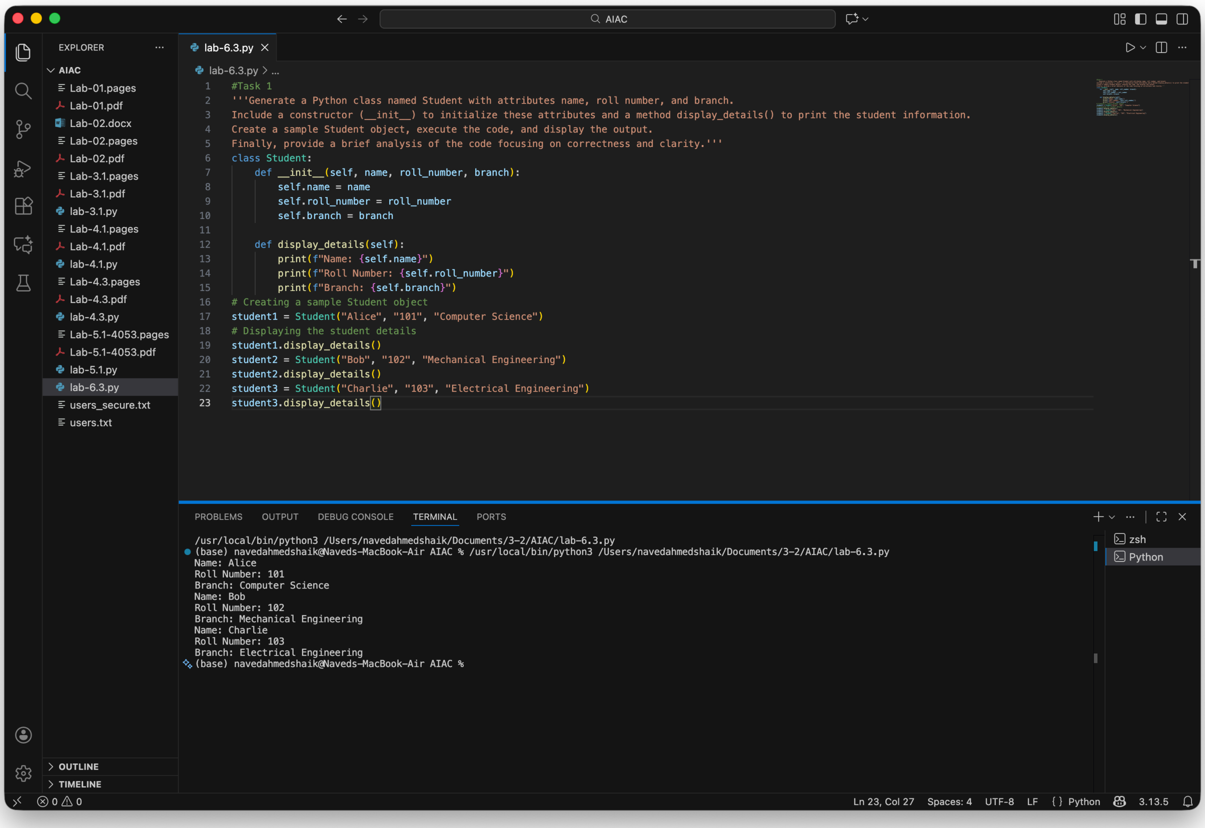
INPUT & OUTPUT:

Input (Student Data)	Output (Displayed Details)
Name: Alice, Roll No: 101, Branch: Computer Science	Name: AliceRoll Number: 101Branch: Computer Science
Name: Bob, Roll No: 102, Branch: Mechanical Engineering	Name: BobRoll Number: 102Branch: Mechanical Engineering
Name: Charlie, Roll No: 103, Branch: Electrical Engineering	Name: CharlieRoll Number: 103Branch: Electrical Engineering

EXPLANATION:

The Student class is used to store and display student information. The constructor (__init__) initializes the student's name, roll number, and branch when an object is created. The display_details() method prints these details in a readable format. Multiple student objects are created, and the method is called for each object to display their respective information.

SCREENSHOT OF GENERATED CODE:



```python
#Task 1
'''Generate a Python class named Student with attributes name, roll number, and branch.
Include a constructor (__init__) to initialize these attributes and a method display_details() to print the student information.
Create a sample Student object, execute the code, and display the output.
Finally, provide a brief analysis of the code focusing on correctness and clarity.'''
class Student:
    def __init__(self, name, roll_number, branch):
        self.name = name
        self.roll_number = roll_number
        self.branch = branch

    def display_details(self):
        print(f"Name: {self.name}")
        print(f"Roll Number: {self.roll_number}")
        print(f"Branch: {self.branch}")
# Creating a sample Student object
student1 = Student("Alice", "101", "Computer Science")
# Displaying the student details
student1.display_details()
student2 = Student("Bob", "102", "Mechanical Engineering")
student2.display_details()
student3 = Student("Charlie", "103", "Electrical Engineering")
student3.display_details()
```

```
/usr/local/bin/python3 /Users/navedahmedshaik/Documents/3-2/AIAC/lab-6.3.py
(base) navedahmedshaik@Naveds-MacBook-Air AIAC % /usr/local/bin/python3 /Users/navedahmedshaik/Documents/3-2/AIAC/lab-6.3.py
Name: Alice
Roll Number: 101
Branch: Computer Science
Name: Bob
Roll Number: 102
Branch: Mechanical Engineering
Name: Charlie
Roll Number: 103
Branch: Electrical Engineering
(base) navedahmedshaik@Naveds-MacBook-Air AIAC %
```

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.

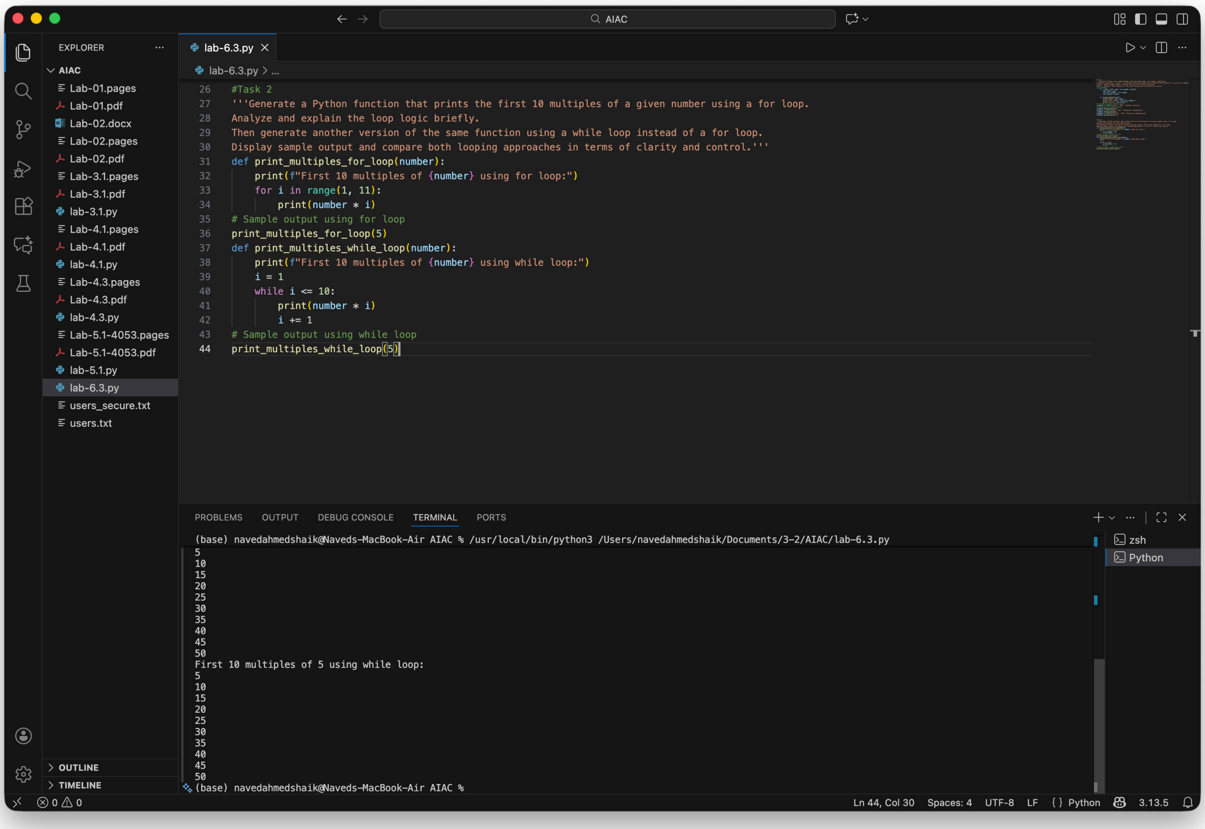
• Output showing the first 10 multiples of a number.

• Comparison and analysis of different looping approaches.

Prompt:

Generate a Python function that prints the first 10 multiples of a given number using a for loop.
Analyze and explain the loop logic briefly.
Then generate another version of the same function using a while loop instead of a for loop.
Display sample output and compare both looping approaches in terms of clarity and control.

SCREENSHOT OF GENERATED CODE:



```python
#Task 2
'''Generate a Python function that prints the first 10 multiples of a given number using a for loop.
Analyze and explain the loop logic briefly.
Then generate another version of the same function using a while loop instead of a for loop.
Display sample output and compare both looping approaches in terms of clarity and control.'''
def print_multiples_for_loop(number):
    print(f"First 10 multiples of {number} using for loop:")
    for i in range(1, 11):
        print(number * i)
# Sample output using for loop
print_multiples_for_loop(5)
def print_multiples_while_loop(number):
    print(f"First 10 multiples of {number} using while loop:")
    i = 1
    while i <= 10:
        print(number * i)
        i += 1
# Sample output using while loop
print_multiples_while_loop(5)
```

```
(base) navedahmedshaik@Naveds-MacBook-Air AIAC % /usr/local/bin/python3 /Users/navedahmedshaik/Documents/3-2/AIAC/lab-6.3.py
5
10
15
20
25
30
35
40
45
50
First 10 multiples of 5 using while loop:
5
10
15
20
25
30
35
40
45
50
(base) navedahmedshaik@Naveds-MacBook-Air AIAC %
```

INPUT & OUTPUT:

Input Number	Method	Output (First 10 Multiples)
5	For Loop	5, 10, 15, 20, 25, 30, 35, 40, 45, 50
5	While Loop	5, 10, 15, 20, 25, 30, 35, 40, 45, 50

EXPLANATION:

The program prints the first 10 multiples of a given number using two different looping techniques. The first function uses a for loop, which iterates from 1 to 10 and prints the product of the number and the loop variable. The second function uses a while loop, where a counter starts at 1 and continues until it reaches 10, printing each multiple and then incrementing the counter. Both functions produce the same output but demonstrate different loop constructs.

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.

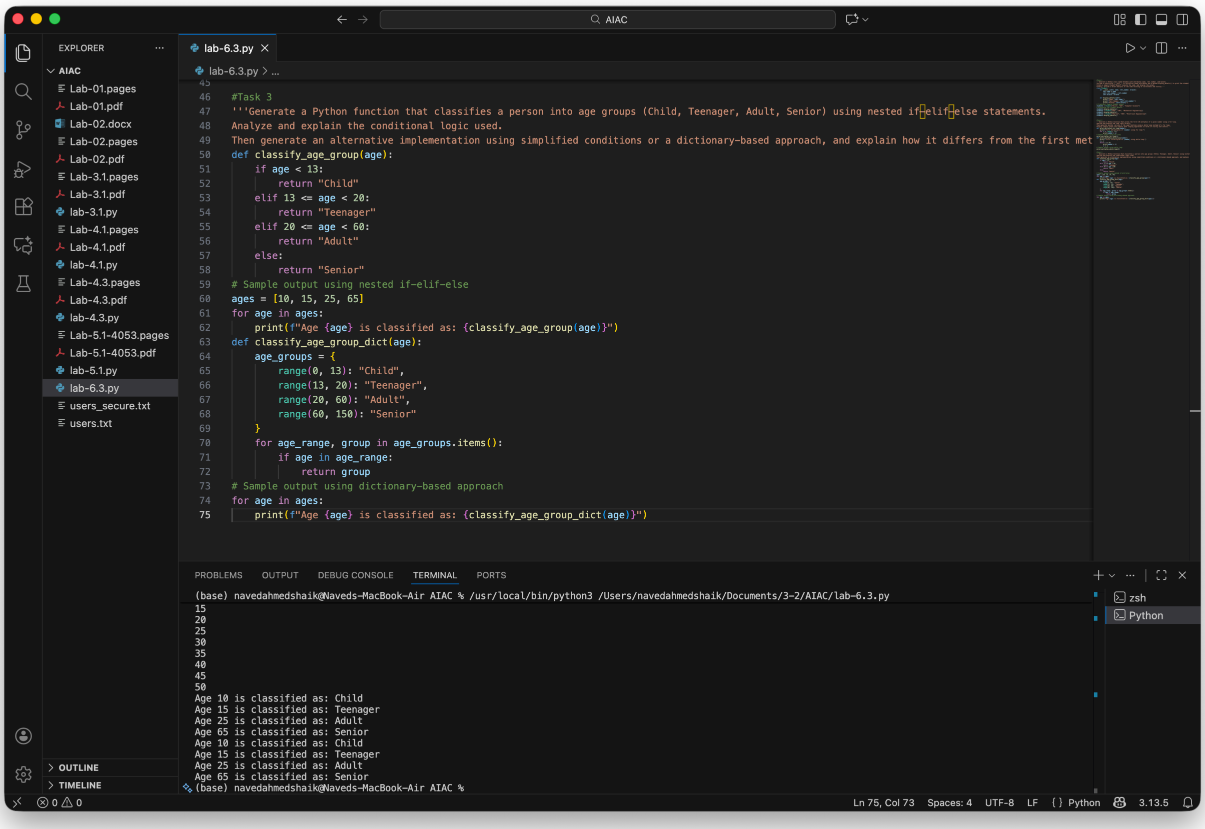
• Clear and correct conditional logic.

• Explanation of how the conditions work.

Prompt:

Generate a Python function that classifies a person into age groups (Child, Teenager, Adult, Senior) using nested if–elif–else statements.
Analyze and explain the conditional logic used.
Then generate an alternative implementation using simplified conditions or a dictionary-based approach, and explain how it differs from the first method.

SCREENSHOT OF GENERATED CODE:



```python
#Task 3
'''Generate a Python function that classifies a person into age groups (Child, Teenager, Adult, Senior) using nested if–elif–else statements.
Analyze and explain the conditional logic used.
Then generate an alternative implementation using simplified conditions or a dictionary-based approach, and explain how it differs from the first met
def classify_age_group(age):
    if age < 13:
        return "Child"
    elif 13 <= age < 20:
        return "Teenager"
    elif 20 <= age < 60:
        return "Adult"
    else:
        return "Senior"
# Sample output using nested if-elif-else
ages = [10, 15, 25, 65]
for age in ages:
    print(f"Age {age} is classified as: {classify_age_group(age)}")
def classify_age_group_dict(age):
    age_groups = {
        range(0, 13): "Child",
        range(13, 20): "Teenager",
        range(20, 60): "Adult",
        range(60, 150): "Senior"
    }
    for age_range, group in age_groups.items():
        if age in age_range:
            return group
# Sample output using dictionary-based approach
for age in ages:
    print(f"Age {age} is classified as: {classify_age_group_dict(age)}")
```

```
(base) navedahmedshaik@Naveds-MacBook-Air AIAC % /usr/local/bin/python3 /Users/navedahmedshaik/Documents/3-2/AIAC/lab-6.3.py
15
20
25
30
35
40
45
50
Age 10 is classified as: Child
Age 15 is classified as: Teenager
Age 25 is classified as: Adult
Age 65 is classified as: Senior
Age 10 is classified as: Child
Age 15 is classified as: Teenager
Age 25 is classified as: Adult
Age 65 is classified as: Senior
(base) navedahmedshaik@Naveds-MacBook-Air AIAC %
```

INPUT & OUTPUT:

Input Age	Output (Age Group)
10	Child
15	Teenager
25	Adult
65	Senior

EXPLANATION:

The program classifies a person's age into different groups using two approaches. The first function uses nested if–elif–else conditions to check the age range and return the appropriate category. The second function uses a dictionary with age ranges as keys and age group names as values, then checks which range the given age falls into. Both methods classify the same ages and produce identical results, showing two different ways to solve the same problem.

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.

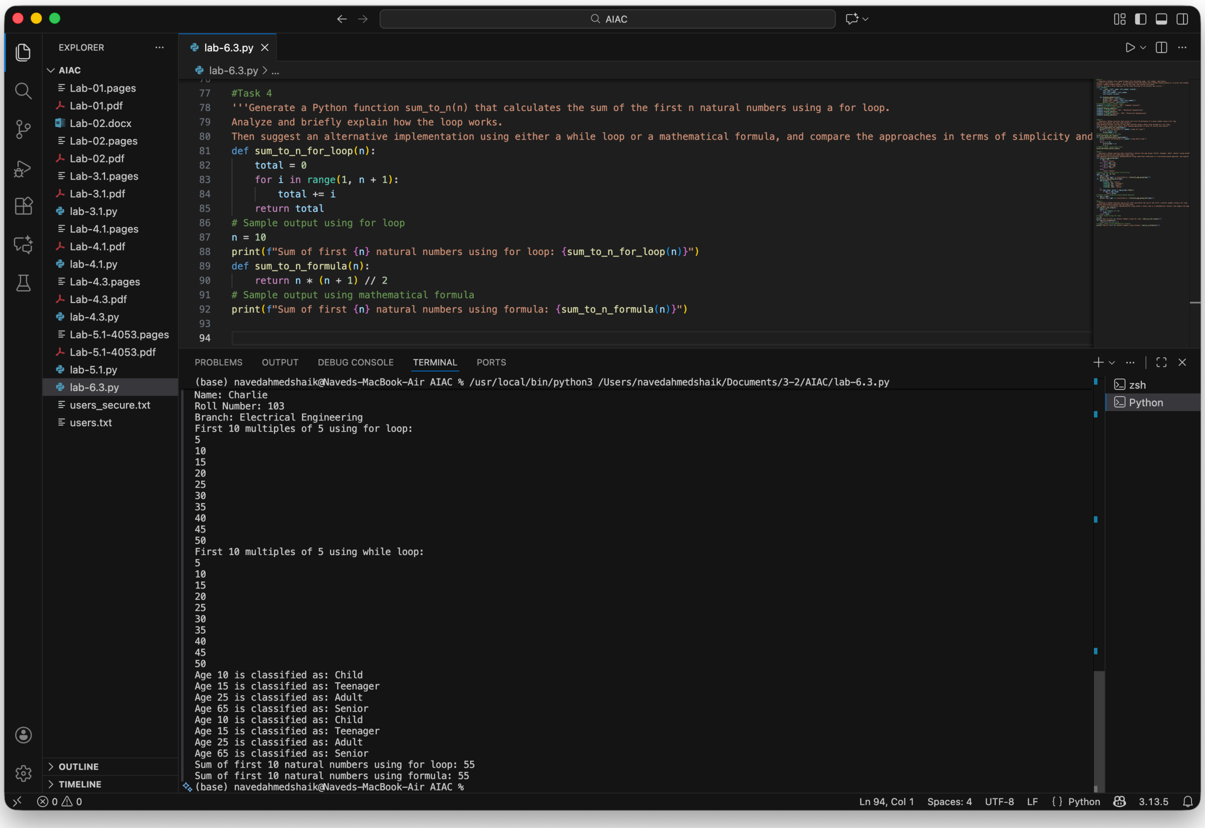
• Correct output for sample inputs.

• Explanation and comparison of different approaches.

Prompt:

Generate a Python function sum_to_n(n) that calculates the sum of the first n natural numbers using a for loop.
Analyze and briefly explain how the loop works.
Then suggest an alternative implementation using either a while loop or a mathematical formula, and compare the approaches in terms of simplicity and efficiency.

SCREENSHOT OF GENERATED CODE:



```python
#Task 4
'''Generate a Python function sum_to_n(n) that calculates the sum of the first n natural numbers using a for loop.
Analyze and briefly explain how the loop works.
Then suggest an alternative implementation using either a while loop or a mathematical formula, and compare the approaches in terms of simplicity and
def sum_to_n_for_loop(n):
    total = 0
    for i in range(1, n + 1):
        total += i
    return total
# Sample output using for loop
n = 10
print(f"Sum of first {n} natural numbers using for loop: {sum_to_n_for_loop(n)}")
def sum_to_n_formula(n):
    return n * (n + 1) // 2
# Sample output using mathematical formula
print(f"Sum of first {n} natural numbers using formula: {sum_to_n_formula(n)}")
```

```
(base) navedahmedshaik@Naveds-MacBook-Air AIAC % /usr/local/bin/python3 /Users/navedahmedshaik/Documents/3-2/AIAC/lab-6.3.py
Name: Charlie
Roll Number: 103
Branch: Electrical Engineering
First 10 multiples of 5 using for loop:
5
10
15
20
25
30
35
40
45
50
First 10 multiples of 5 using while loop:
5
10
15
20
25
30
35
40
45
50
Age 10 is classified as: Child
Age 15 is classified as: Teenager
Age 25 is classified as: Adult
Age 65 is classified as: Senior
Age 10 is classified as: Child
Age 15 is classified as: Teenager
Age 25 is classified as: Adult
Age 65 is classified as: Senior
Sum of first 10 natural numbers using for loop: 55
Sum of first 10 natural numbers using formula: 55
(base) navedahmedshaik@Naveds-MacBook-Air AIAC %
```

INPUT & OUTPUT:

Input (n)	Method	Output
10	For Loop	55
10	Formula	55

EXPLANATION:

The program calculates the sum of the first *n* natural numbers using two different methods. The first function uses a for loop to add numbers from 1 to *n* one by one. The second function uses the mathematical formula n(n+1)/2 to compute the sum directly. Both approaches give the same result, but the formula method is more efficient.

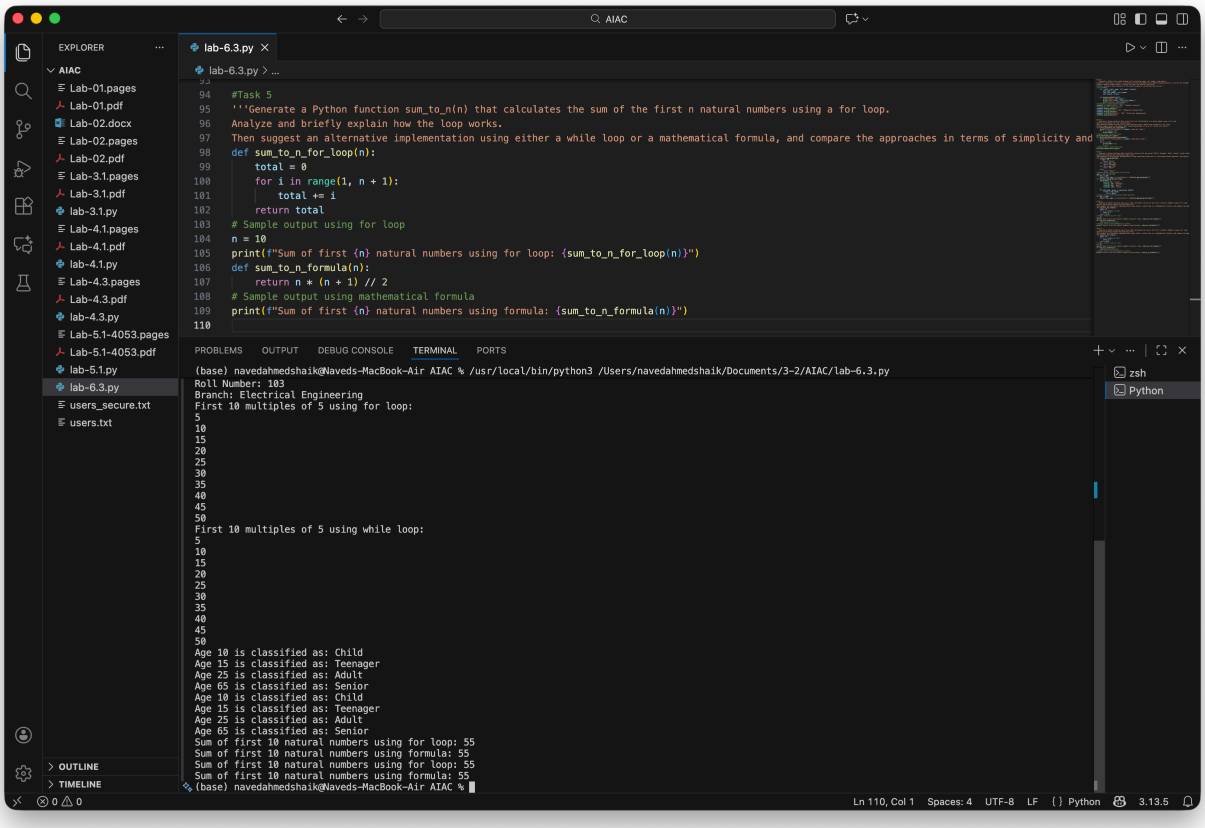
Task Description #5: Classes (Bank Account Class)

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.

Prompt:

Generate a Python function sum_to_n(n) that calculates the sum of the first n natural numbers using a for loop. Analyze and briefly explain how the loop works.
Then suggest an alternative implementation using either a while loop or a mathematical formula, and compare the approaches in terms of simplicity and efficiency.

SCREENSHOT OF GENERATED CODE:



```python
#Task 5
'''Generate a Python function sum_to_n(n) that calculates the sum of the first n natural numbers using a for loop.
Analyze and briefly explain how the loop works.
Then suggest an alternative implementation using either a while loop or a mathematical formula, and compare the approaches in terms of simplicity and
def sum_to_n_for_loop(n):
    total = 0
    for i in range(1, n + 1):
        total += i
    return total
# Sample output using for loop
n = 10
print(f"Sum of first {n} natural numbers using for loop: {sum_to_n_for_loop(n)}")
def sum_to_n_formula(n):
    return n * (n + 1) // 2
# Sample output using mathematical formula
print(f"Sum of first {n} natural numbers using formula: {sum_to_n_formula(n)}")
```

```
(base) navedahmedshaik@Naveds-MacBook-Air AIAC % /usr/local/bin/python3 /Users/navedahmedshaik/Documents/3-2/AIAC/lab-6.3.py
Roll Number: 103
Branch: Electrical Engineering
First 10 multiples of 5 using for loop:
5
10
15
20
25
30
35
40
45
50
First 10 multiples of 5 using while loop:
5
10
15
20
25
30
35
40
45
50
Age 10 is classified as: Child
Age 15 is classified as: Teenager
Age 25 is classified as: Adult
Age 65 is classified as: Senior
Age 10 is classified as: Child
Age 15 is classified as: Teenager
Age 25 is classified as: Adult
Age 65 is classified as: Senior
Sum of first 10 natural numbers using for loop: 55
Sum of first 10 natural numbers using formula: 55
Sum of first 10 natural numbers using for loop: 55
Sum of first 10 natural numbers using formula: 55
(base) navedahmedshaik@Naveds-MacBook-Air AIAC %
```

INPUT & OUTPUT:

Input (n)	Method	Output	Reason
10	For Loop	55	Adds numbers from 1 to 10 iteratively.
10	Formula	55	Uses the formula ($10 \times 11 \div 2 = 55$).

EXPLANATION:

The program finds the sum of the first *n* natural numbers using two approaches. The first function uses a for loop to iteratively add numbers from 1 to *n*. The second function uses a mathematical formula $n(n+1)/2$ to compute the sum directly without looping. Both methods return the same result, but the formula-based approach is more efficient.