AI Assisted Coding

Assignment - 06

M.Sathwik || 2303A51483|| Batch:- 08

Task Description #1: Classes (Student Class)

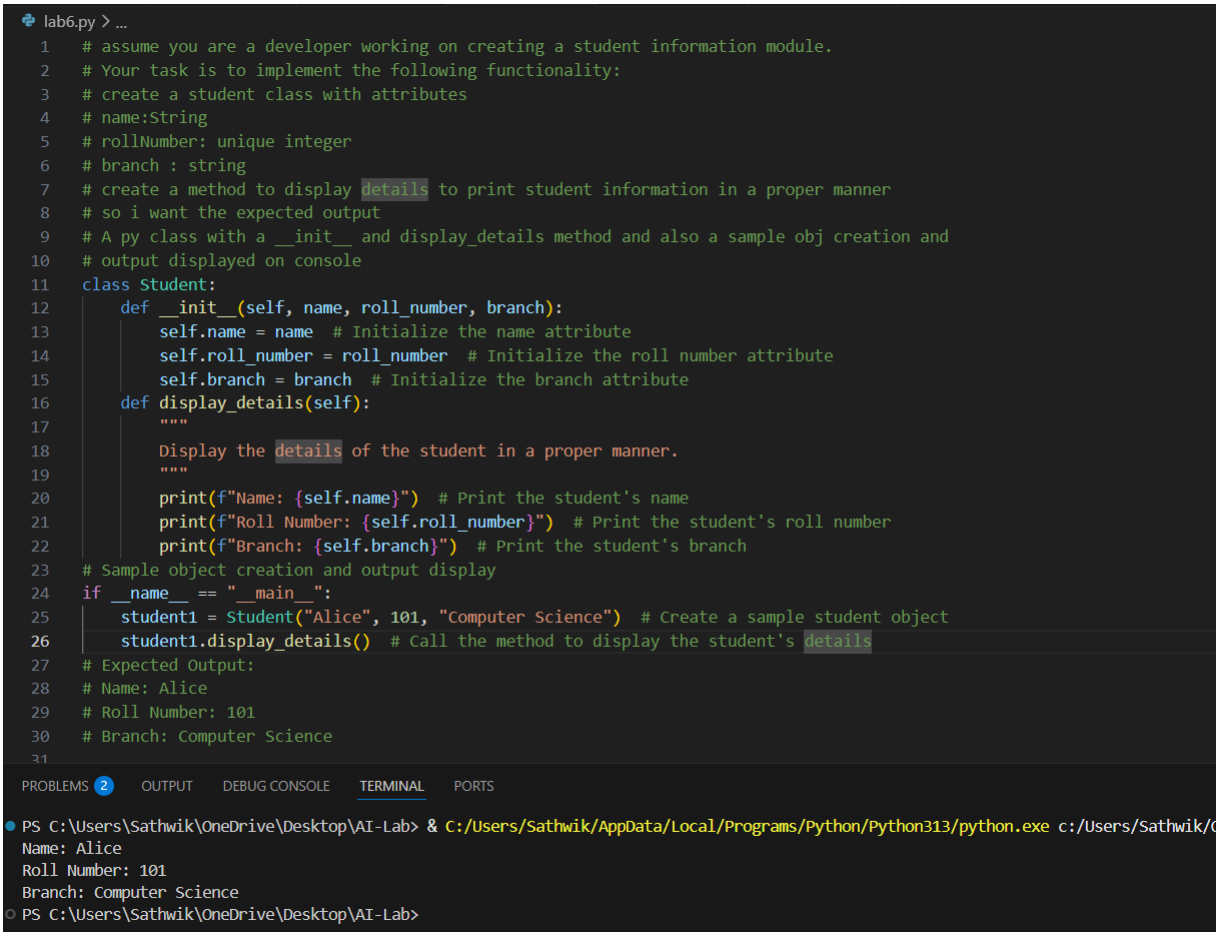
Scenario

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.

Code:



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

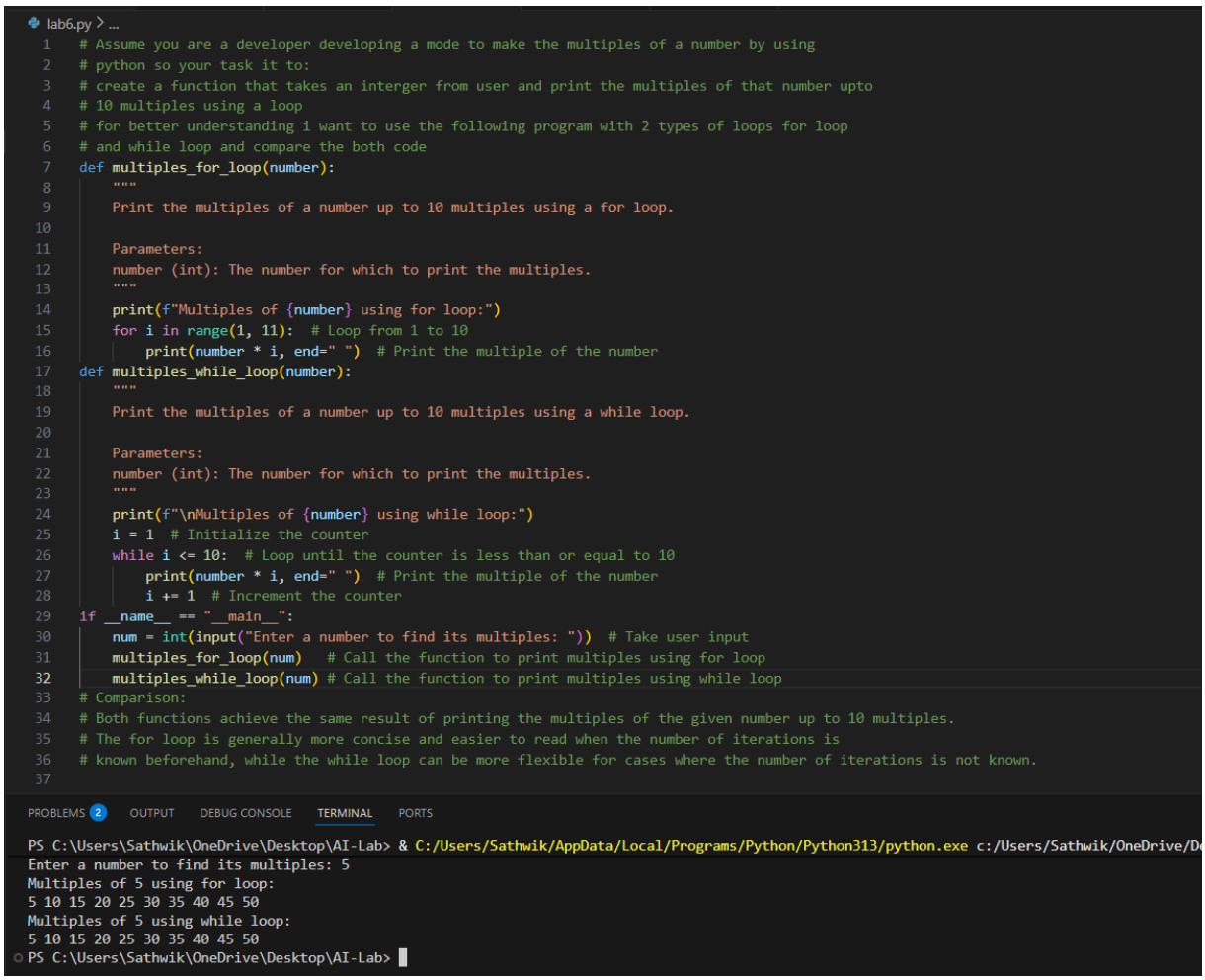
Scenario

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).

Code:



```
lab6.py > ...
# Assume you are a developer developing a mode to make the multiples of a number by using
# python so your task it to:
# create a function that takes an interger from user and print the multiples of that number upto
# 10 multiples using a loop
# for better understanding i want to use the following program with 2 types of loops for loop
# and while loop and compare the both code
def multiples_for_loop(number):
    """
    Print the multiples of a number up to 10 multiples using a for loop.

    Parameters:
    number (int): The number for which to print the multiples.
    """
    print(f"Multiples of {number} using for loop:")
    for i in range(1, 11):  # Loop from 1 to 10
        print(number * i, end=" ")  # Print the multiple of the number
def multiples_while_loop(number):
    """
    Print the multiples of a number up to 10 multiples using a while loop.

    Parameters:
    number (int): The number for which to print the multiples.
    """
    print(f"\nMultiples of {number} using while loop:")
    i = 1  # Initialize the counter
    while i <= 10:  # Loop until the counter is less than or equal to 10
        print(number * i, end=" ")  # Print the multiple of the number
        i += 1  # Increment the counter
if __name__ == "__main__":
    num = int(input("Enter a number to find its multiples: "))  # Take user input
    multiples_for_loop(num)   # Call the function to print multiples using for loop
    multiples_while_loop(num) # Call the function to print multiples using while loop
# Comparison:
# Both functions achieve the same result of printing the multiples of the given number up to 10 multiples.
# The for loop is generally more concise and easier to read when the number of iterations is
# known beforehand, while the while loop can be more flexible for cases where the number of iterations is not known.

PROBLEMS 2  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\Sathwik\OneDrive\Desktop\AI-Lab> & C:/Users/Sathwik/AppData/Local/Programs/Python/Python313/python.exe c:/Users/Sathwik/OneDrive/D
Enter a number to find its multiples: 5
Multiples of 5 using for loop:
5 10 15 20 25 30 35 40 45 50
Multiples of 5 using while loop:
5 10 15 20 25 30 35 40 45 50
PS C:\Users\Sathwik\OneDrive\Desktop\AI-Lab>
```

Task Description #3: Conditional Statements (Age Classification)

Scenario

You are building a basic classification system based on age.

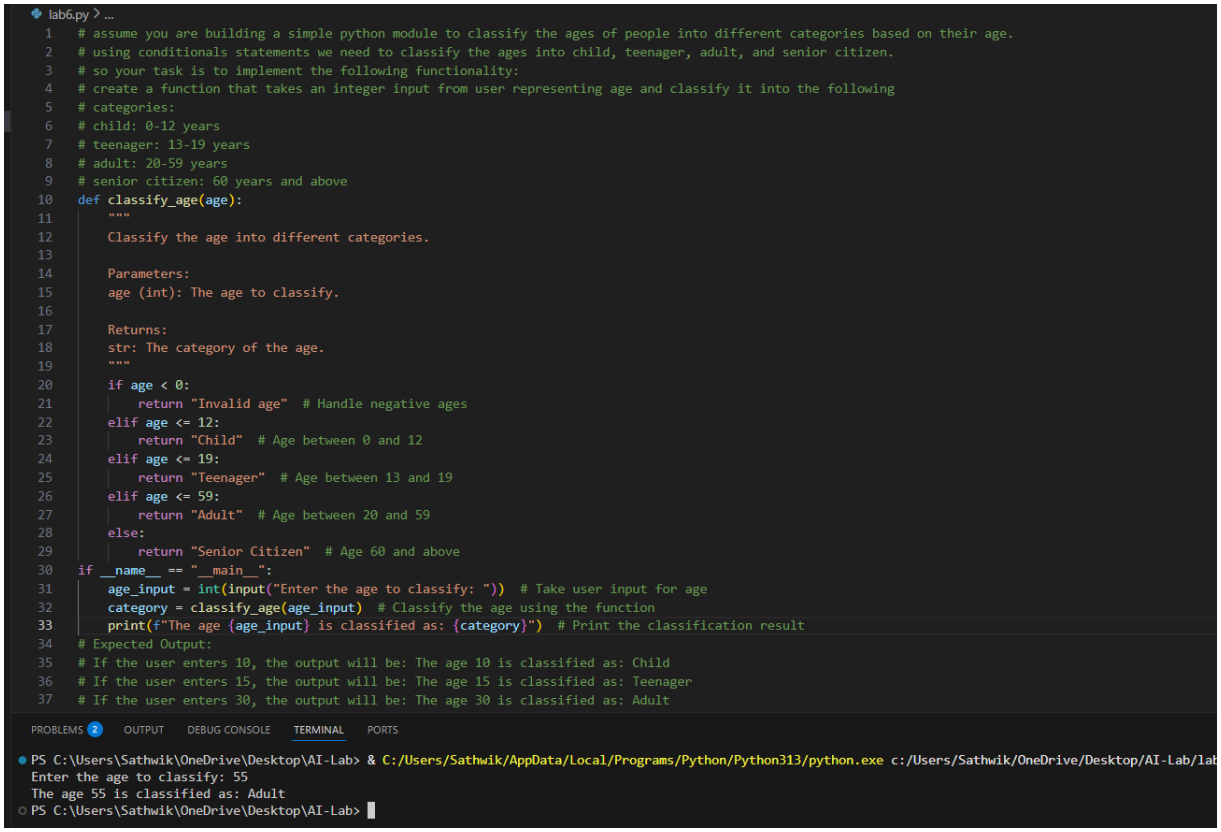
Task

- 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).

Code:



```
lab6.py > ...
# assume you are building a simple python module to classify the ages of people into different categories based on their age.
# using conditionals statements we need to classify the ages into child, teenager, adult, and senior citizen.
# so your task is to implement the following functionality:
# create a function that takes an integer input from user representing age and classify it into the following
# categories:
# child: 0-12 years
# teenager: 13-19 years
# adult: 20-59 years
# senior citizen: 60 years and above
def classify_age(age):
    """
    Classify the age into different categories.

    Parameters:
    age (int): The age to classify.

    Returns:
    str: The category of the age.
    """
    if age < 0:
        return "Invalid age"  # Handle negative ages
    elif age <= 12:
        return "Child"  # Age between 0 and 12
    elif age <= 19:
        return "Teenager"  # Age between 13 and 19
    elif age <= 59:
        return "Adult"  # Age between 20 and 59
    else:
        return "Senior Citizen"  # Age 60 and above
if __name__ == "__main__":
    age_input = int(input("Enter the age to classify: "))  # Take user input for age
    category = classify_age(age_input)  # Classify the age using the function
    print(f"The age {age_input} is classified as: {category}")  # Print the classification result
# Expected Output:
# If the user enters 10, the output will be: The age 10 is classified as: Child
# If the user enters 15, the output will be: The age 15 is classified as: Teenager
# If the user enters 30, the output will be: The age 30 is classified as: Adult

PROBLEMS 2  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\Sathwik\OneDrive\Desktop\AI-Lab> & C:/Users/Sathwik/AppData/Local/Programs/Python/Python313/python.exe c:/Users/Sathwik/OneDrive/Desktop/AI-Lab/lab
Enter the age to classify: 55
The age 55 is classified as: Adult
PS C:\Users\Sathwik\OneDrive\Desktop\AI-Lab>
```

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.

Code:

```python
# Assume you are a developer working on creating a module to compare for loop and while
# loop in python
# so your task is to implement the following functionality:
# create a function that takes an integer from user
# you need to calculate the 1st sum of natural numbers using for loop and while loop both
# and compare the both code
# First i want to execute the code using for loop and then using while loop and then compare
# the both code
def sum_natural_numbers_for_loop(n):
    """
    Calculate the sum of the first n natural numbers using a for loop.
    Parameters:
    n (int): The number of natural numbers to sum.
    Returns:
    int: The sum of the first n natural numbers.
    """
    total_sum = 0  # Initialize the sum variable
    for i in range(1, n + 1):  # Loop from 1 to n
        total_sum += i  # Add the current number to the total sum
    return total_sum  # Return the calculated sum
def sum_natural_numbers_while_loop(n):
    """
    Calculate the sum of the first n natural numbers using a while loop.
    Parameters:
    n (int): The number of natural numbers to sum.
    Returns:
    int: The sum of the first n natural numbers.
    """
    total_sum = 0  # Initialize the sum variable
    i = 1  # Initialize the counter
    while i <= n:  # Loop until the counter is less than or equal to n
        total_sum += i  # Add the current number to the total sum
        i += 1  # Increment the counter
    return total_sum  # Return the calculated sum
if __name__ == "__main__":
    n = int(input("Enter a number to calculate the sum of the first n natural numbers: "))  # Take user input
    sum_for_loop = sum_natural_numbers_for_loop(n)  # Calculate sum using for loop
    sum_while_loop = sum_natural_numbers_while_loop(n)  # Calculate sum using while loop
    print(f"Sum of the first {n} natural numbers using for loop: {sum_for_loop}")  # Print result from for loop
    print(f"Sum of the first {n} natural numbers using while loop: {sum_while_loop}")  # Print result from while loop
    # Comparison:
    # Both functions calculate the same result, which is the sum of the first n natural numbers.
    # The for loop is more concise and easier to read when the number of iterations is known.
    # While the while loop can be more flexible for cases where the number of iterations is not predetermined.

```

```
PROBLEMS 2  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS C:\Users\Sathwik\OneDrive\Desktop\AI-Lab> & C:/Users/Sathwik/AppData/Local/Programs/Python/Python313/python.exe c:/Users/Sathwik/OneDrive/Desktop/AI-Lab/lab6
Enter a number to calculate the sum of the first n natural numbers: 145
Sum of the first 145 natural numbers using for loop: 10585
Sum of the first 145 natural numbers using while loop: 10585
PS C:\Users\Sathwik\OneDrive\Desktop\AI-Lab>
```

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.

Code:

lab6.py > BankAccount > deposit

```python
# Act as a developer creating a module that your are developing a banking application.
# Your task is to implement the following functionality:
# create a basic function that such as deposit(), withdraw() and check balance().
# also add the step by step explanation of the code. and read the input from user and display the output on console.
class BankAccount:
    def __init__(self, account_holder, initial_balance=0):
        self.account_holder = account_holder  # Initialize the account holder's name
        self.balance = initial_balance        # Initialize the account balance
    def deposit(self, amount):
        """
        Deposit a certain amount into the bank account.

        Parameters:
        amount (float): The amount to be deposited.
        """
        if amount > 0:  # Check if the deposit amount is positive
            self.balance += amount  # Add the deposit amount to the balance
            print(f"Deposited ${amount:.2f}. New balance: ${self.balance:.2f}")  # Print the new balance
        else:
            print("Deposit amount must be positive.")  # Handle invalid deposit amounts
    def withdraw(self, amount):
        """
        Withdraw a certain amount from the bank account.

        Parameters:
        amount (float): The amount to be withdrawn.
        """
        if amount > self.balance:  # Check if there are sufficient funds for withdrawal
            print("Insufficient funds.")  # Handle insufficient funds
        elif amount <= 0:  # Check if the withdrawal amount is positive
            print("Withdrawal amount must be positive.")  # Handle invalid withdrawal amounts
        else:
            self.balance -= amount  # Subtract the withdrawal amount from the balance
            print(f"Withdrew ${amount:.2f}. New balance: ${self.balance:.2f}")  # Print the new balance
    def check_balance(self):
        """
        Check and display the current balance of the bank account.
        """
        print(f"Current balance: ${self.balance:.2f}")  # Print the current balance
if __name__ == "__main__":
    account_holder_name = input("Enter the account holder's name: ")  # Take user input for account holder's name
    initial_deposit = float(input("Enter the initial deposit amount: "))  # Take user input for initial deposit
    account = BankAccount(account_holder_name, initial_deposit)  # Create a bank account object with the provided details
    while True:  # Start an infinite loop to allow multiple transactions
        print("\nChoose an option:")
        print("1. Deposit")
        print("2. Withdraw")
        print("3. Check Balance")
        print("4. Exit")
        choice = input("Enter your choice (1-4): ")  # Take user input for the desired action
```

```
Click to add a breakpoint
        print("3. Check Balance")
        print("4. Exit")
        choice = input("Enter your choice (1-4): ")  # Take user input for the desired action
        if choice == "1":
            amount = float(input("Enter the amount to deposit: "))  # Take user input for deposit amount
            account.deposit(amount)  # Call the deposit method
        elif choice == "2":
            amount = float(input("Enter the amount to withdraw: "))  # Take user input for withdrawal amount
            account.withdraw(amount)  # Call the withdraw method
        elif choice == "3":
            account.check_balance()  # Call the check balance method
        elif choice == "4":
            print("Thank you for using our banking application. Goodbye!")  # Exit message
            break  # Exit the loop and end the program
        else:
            print("Invalid choice. Please enter a number between 1 and 4.")  # Handle invalid menu choices
```

PROBLEMS 2 | OUTPUT | DEBUG CONSOLE | TERMINAL | PORTS

```
PS C:\Users\Sathwik\OneDrive\Desktop\AI-Lab> & C:/Users/Sathwik/AppData/Local/Programs/Python/Python313/python.exe c:/Users/Sathwik/OneDrive/Desktop/AI-L
Enter the account holder's name: Sathwik
Enter the initial deposit amount: 25000

Choose an option:
1. Deposit
2. Withdraw
3. Check Balance
4. Exit
Enter your choice (1-4): 2
Enter the amount to withdraw: 1200
Withdrew $1200.00. New balance: $23800.00

Choose an option:
1. Deposit
2. Withdraw
3. Check Balance
4. Exit
Enter your choice (1-4): 3
Current balance: $23800.00

Choose an option:
1. Deposit
2. Withdraw
3. Check Balance
4. Exit
Enter your choice (1-4): 4
Thank you for using our banking application. Goodbye!
PS C:\Users\Sathwik\OneDrive\Desktop\AI-Lab>
```