

# AI Assignment-5.1

Name: G. Bala Varshitha

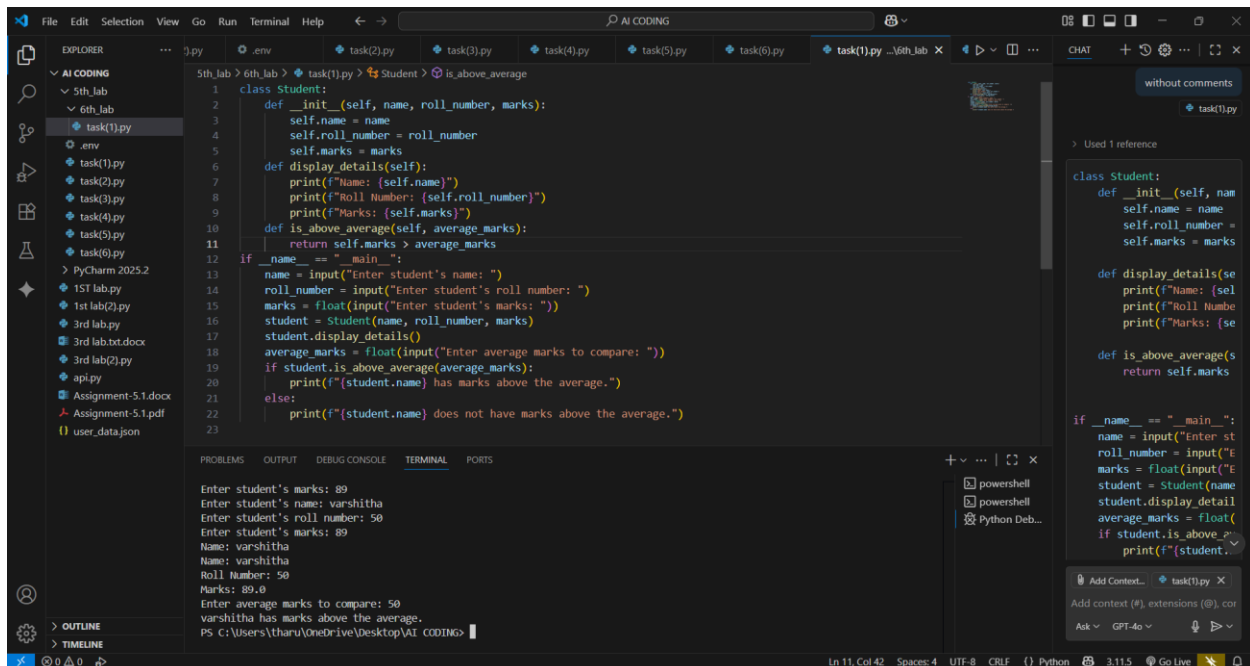
HT.NO: 2403A52050

Batch: AIB03

Task 1: Start a Python class named Student with attributes name, roll\_number, and marks. Prompt GitHub Copilot to complete methods for displaying details and checking if marks are above average

PROMPT: Start a Python class named Student with attributes name, roll number, and marks.

CODE:



```
File Edit Selection View Go Run Terminal Help AI CODING task(1).py task(2).py task(3).py task(4).py task(5).py task(6).py task(1).py _6th_lab X
```

```
5th_lab > 6th_lab > task(1).py > Student > is_above_average
1 class Student:
2     def __init__(self, name, roll_number, marks):
3         self.name = name
4         self.roll_number = roll_number
5         self.marks = marks
6     def display_details(self):
7         print(f"Name: {self.name}")
8         print(f"Roll Number: {self.roll_number}")
9         print(f"Marks: {self.marks}")
10    def is_above_average(self, average_marks):
11        return self.marks > average_marks
12    if __name__ == "__main__":
13        name = input("Enter student's name: ")
14        roll_number = input("Enter student's roll number: ")
15        marks = float(input("Enter student's marks: "))
16        student = Student(name, roll_number, marks)
17        student.display_details()
18        average_marks = float(input("Enter average marks to compare: "))
19        if student.is_above_average(average_marks):
20            print(f"{student.name} has marks above the average.")
21        else:
22            print(f"{student.name} does not have marks above the average.")
23
```

```
Enter student's marks: 89
Enter student's name: varshitha
Enter student's roll number: 50
Enter student's marks: 89
Name: varshitha
Name: varshitha
Roll Number: 50
Marks: 89.0
Enter average marks to compare: 50
varshitha has marks above the average.
PS C:\Users\tharu\OneDrive\Desktop\AI CODING>
```

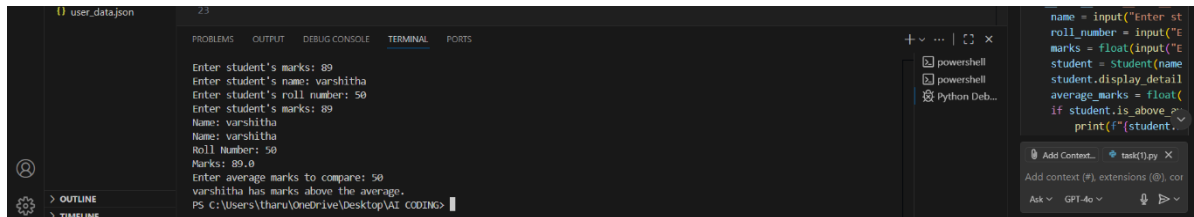
```
class Student:
    def __init__(self, name, roll_number, marks):
        self.name = name
        self.roll_number = roll_number
        self.marks = marks

    def display_details(self):
        print(f"Name: {self.name}")
        print(f"Roll Number: {self.roll_number}")
        print(f"Marks: {self.marks}")

    def is_above_average(self, average_marks):
        return self.marks > average_marks

if __name__ == "__main__":
    name = input("Enter student's name: ")
    roll_number = input("Enter student's roll number: ")
    marks = float(input("Enter student's marks: "))
    student = Student(name, roll_number, marks)
    student.display_details()
    average_marks = float(input("Enter average marks to compare: "))
    if student.is_above_average(average_marks):
        print(f"{student.name} has marks above the average.")
    else:
        print(f"{student.name} does not have marks above the average.")
```

OP:



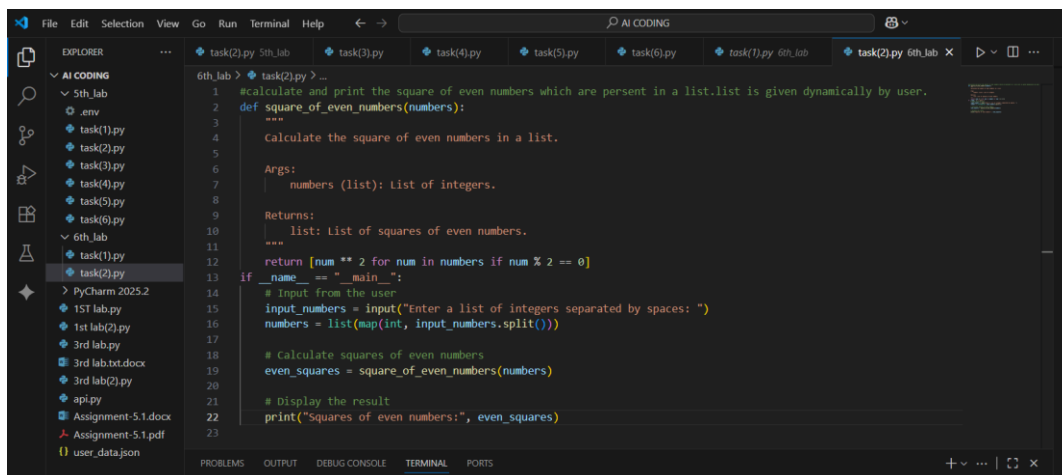
The screenshot shows a VS Code editor with a Python file named `task(2).py`. The code prompts the user for student marks, name, and roll number, then calculates the average and compares it to a user-defined average. The terminal output shows the execution of the script with sample data.

```
Enter student's marks: 89
Enter student's name: varshitha
Enter student's roll number: 50
Enter student's marks: 89
Name: varshitha
Roll Number: 50
Marks: 89.0
Enter average marks to compare: 50
varshitha has marks above the average.
PS C:\Users\tharu\OneDrive\Desktop\AI CODING>
```

Task 2: Write the first two lines of a for loop to iterate through a list of numbers. Use a comment prompt to let Copilot suggest how to calculate and print the square of even numbers only.

Prompt: calculate and print the square of even numbers which are present in a list. list is given dynamically by user

CODE:



The screenshot shows a VS Code editor with a Python file named `task(2).py`. The code defines a function `square_of_even_numbers` that takes a list of integers and returns a list of squares of even numbers. The terminal output shows the execution of the script with sample data.

```
6th_lab > task(2).py >
1 #calculate and print the square of even numbers which are present in a list. list is given dynamically by user.
2 def square_of_even_numbers(numbers):
3     """
4     Calculate the square of even numbers in a list.
5
6     Args:
7         numbers (list): List of integers.
8
9     Returns:
10        list: List of squares of even numbers.
11    """
12    return [num ** 2 for num in numbers if num % 2 == 0]
13 if __name__ == "__main__":
14     # Input from the user
15     input_numbers = input("Enter a list of integers separated by spaces: ")
16     numbers = list(map(int, input_numbers.split()))
17
18     # Calculate squares of even numbers
19     even_squares = square_of_even_numbers(numbers)
20
21     # Display the result
22     print("Squares of even numbers:", even_squares)
23
```

OP:



The screenshot shows the terminal output of the Python script. It prompts the user for roll number, marks, and average, then calculates the squares of even numbers from a list of integers.

```
Roll Number: 50
Marks: 89.0
Enter average marks to compare: 50
varshitha has marks above the average.
PS C:\Users\tharu\OneDrive\Desktop\AI CODING> ^C
PS C:\Users\tharu\OneDrive\Desktop\AI CODING>
PS C:\Users\tharu\OneDrive\Desktop\AI CODING> cd 'c:\Users\tharu\OneDrive\Desktop\AI CODING'; & 'c:\Program Files\Python311\python.exe' 'c:\Users\tharu\vscode\extensions\ms-python.debugpy-2025.10.0-win32-x64\bundle\libs\debugpy\launcher' '65002'
Enter a list of integers separated by spaces: 1 2 3 4 5 6
Squares of even numbers: [4, 16, 36]
PS C:\Users\tharu\OneDrive\Desktop\AI CODING>
```

Observation: