

AI ASSISTED CODING

Roll no.: 2503A51L41

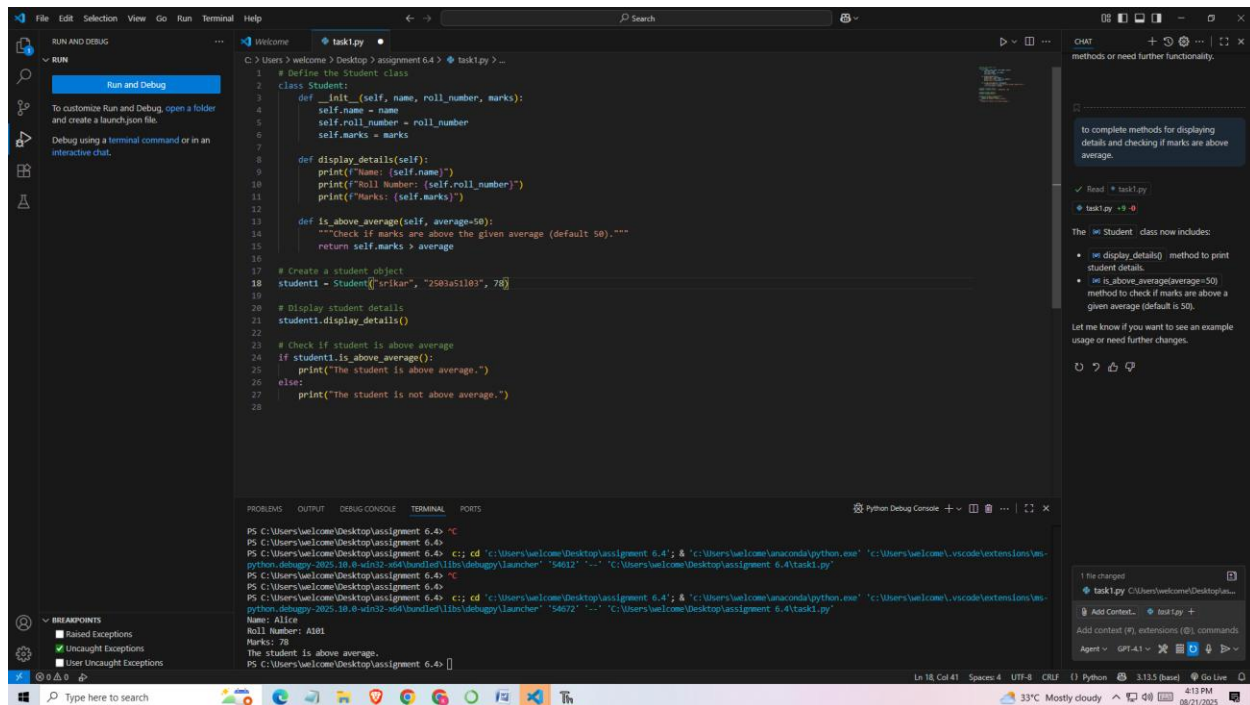
Batch: 24BTCAICSB20

Name: G Hasini

Lab 4: AI-Based Code Auto-Completion – Classes, Loops, and Conditionals in Python using GitHub Copilot

Task #1:

PROMPT : Start a Python class named student with attributes name, roll_number, and marks. Add methods to display student details and check if marks are above average using if-else.



```
1 # Define the Student class
2 class Student:
3     def __init__(self, name, roll_number, marks):
4         self.name = name
5         self.roll_number = roll_number
6         self.marks = marks
7
8     def display_details(self):
9         print("Name: (self.name)")
10        print("Roll Number: (self.roll_number)")
11        print("Marks: (self.marks)")
12
13    def is_above_average(self, average=50):
14        """Check if marks are above the given average (default 50)."""
15        return self.marks > average
16
17 # Create a student object
18 student1 = Student("srikan", "2503A5103", 78)
19
20 # Display student details
21 student1.display_details()
22
23 # Check if student is above average
24 if student1.is_above_average():
25     print("The student is above average.")
26 else:
27     print("The student is not above average.")
28
```

Python Debug Console

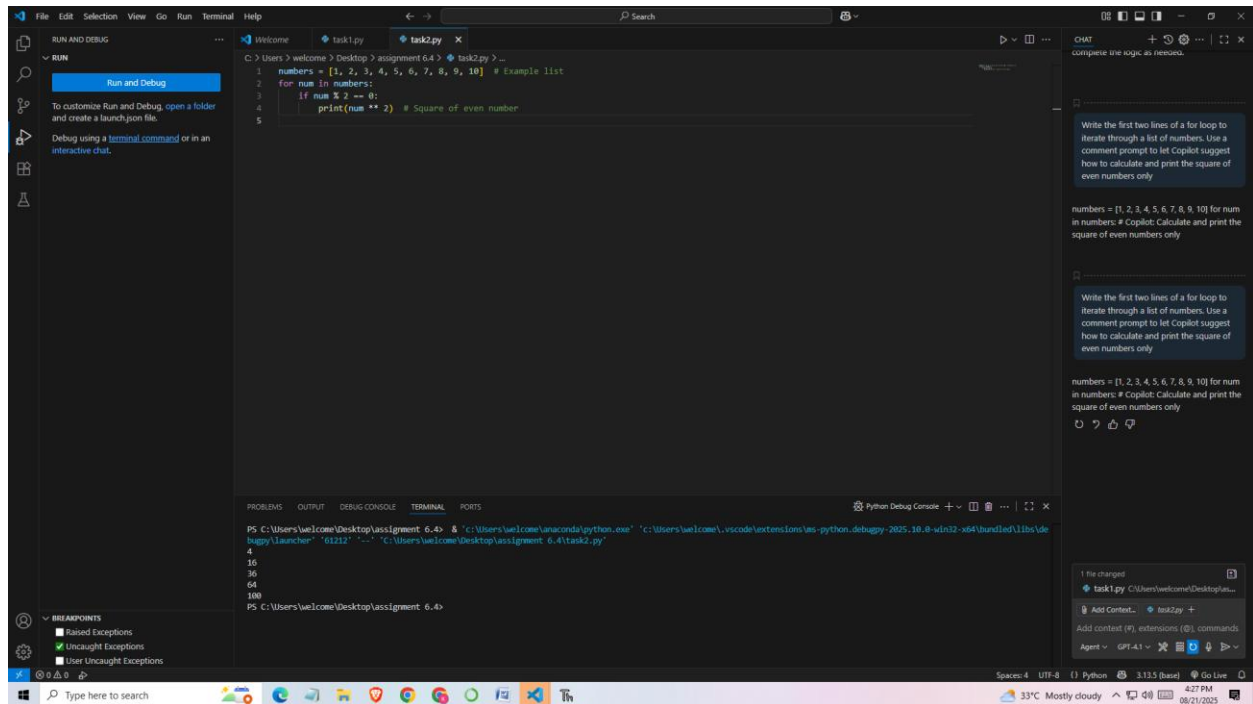
```
PS C:\Users\welcome\Desktop\assignment 6.4> cd "C:\Users\welcome\Desktop\assignment 6.4"; & "C:\Users\welcome\anaconda\python.exe" "C:\Users\welcome\.vscode\extensions\ms-python.debugpy-2025.10.0-4032-044\bundled\libs\debugpy\launcher" "54812" -- "C:\Users\welcome\Desktop\assignment 6.4\task1.py"
PS C:\Users\welcome\Desktop\assignment 6.4> cd "C:\Users\welcome\Desktop\assignment 6.4"; & "C:\Users\welcome\anaconda\python.exe" "C:\Users\welcome\.vscode\extensions\ms-python.debugpy-2025.10.0-4032-044\bundled\libs\debugpy\launcher" "54812" -- "C:\Users\welcome\Desktop\assignment 6.4\task1.py"
Name: Alice
Roll Number: 2503A5103
Marks: 78
The student is above average.
```

observation:

- 1.__init__() initializes the attributes (name, roll_number, marks).
- 2.display_details() prints the student information in a formatted way.is_above_average() checks if marks > 50 (or defined average) using if-else and returns the result.

Task#2:

PROMPT: Write the first two lines of a for loop to iterate through a list of numbers. Add a comment suggesting to calculate and print the square of even numbers only.

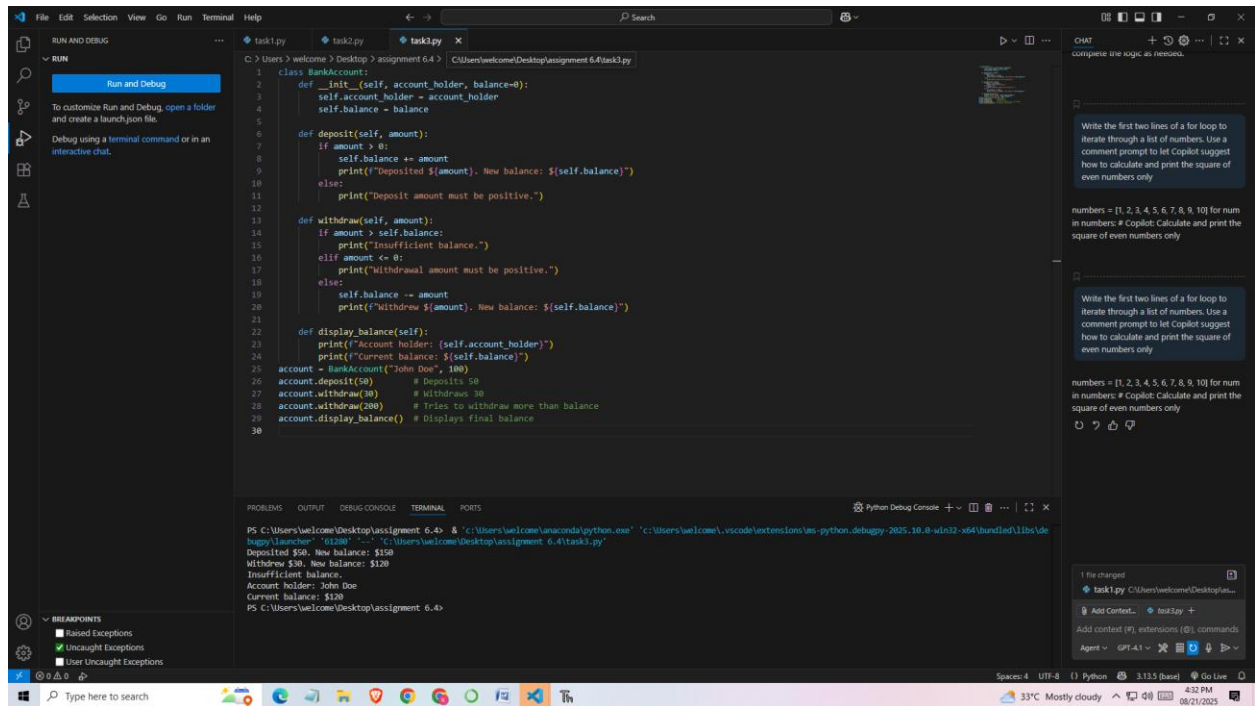


Observation:

- numbers: a list of integers from 1 to 10.
- for num in numbers:: loops through each number in the list.
- if num % 2 == 0:: checks if the number is even using the modulus operator.
- print(num ** 2): prints the square of the number if it is even.

Task#3:

PROMPT: Create a class called BankAccount with account_holder and balance. Add methods for deposit(), withdraw() and check for insufficient balance.

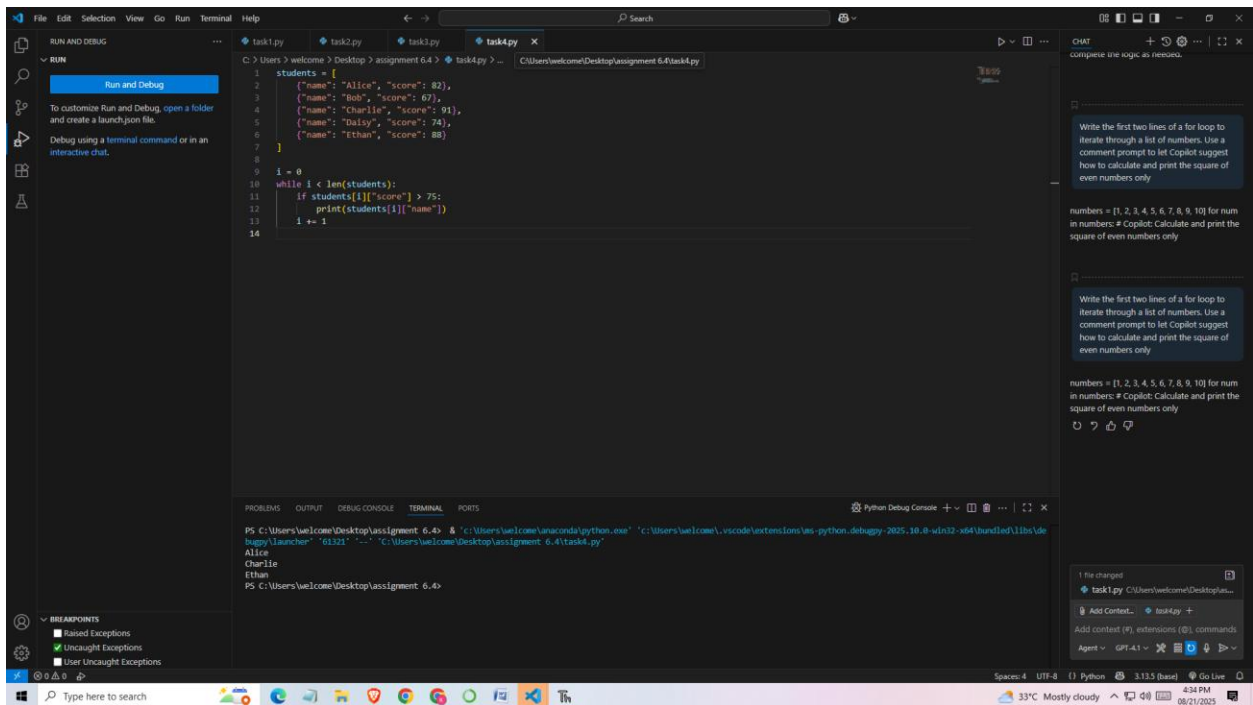


Observation:

- `__init__()` sets up the account.
- `deposit()` and `withdraw()` handle balance changes with checks.
- `display_balance()` shows the account state.

Task #4:

PROMPT: Define a list of student dictionaries with keys name and score. Write a while loop to print the names of students who scored more than 75.



Observation:

A list of dictionaries stores student data.

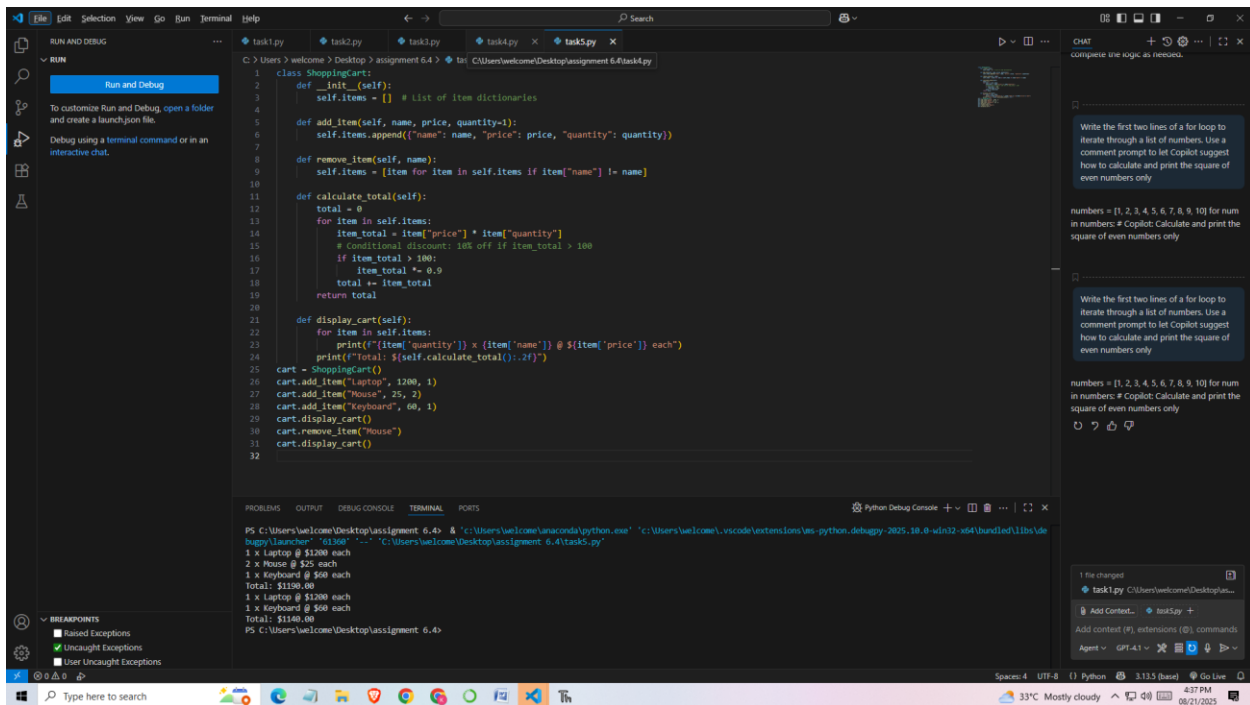
A counter/index variable is used for the while loop.

loop → if score > 75, the student's name is printed.

The loop increments the counter until all students are checked.

Task #5:

PROMPT: Start a ShoppingCart class with an empty items list. Add methods to add_item, remove_item, and calculate_total with discounts using a for loop and if-else conditions



Observation:

- `items` is a list of dictionaries representing cart items.
- `add_item()` adds a product.
- `remove_item()` filters out items by name.
- `calculate_total()` uses a for loop and gives a 10% discount for high-value items.
- `display_cart()` prints the full receipt.