**AI ASSISTED CODING**

**LAB - 3 [Assignment - 6.4]**

**NAME :** **B. REVANTH**

**ENROLL NO :**  **2403A52041**

**BATCH : 03**

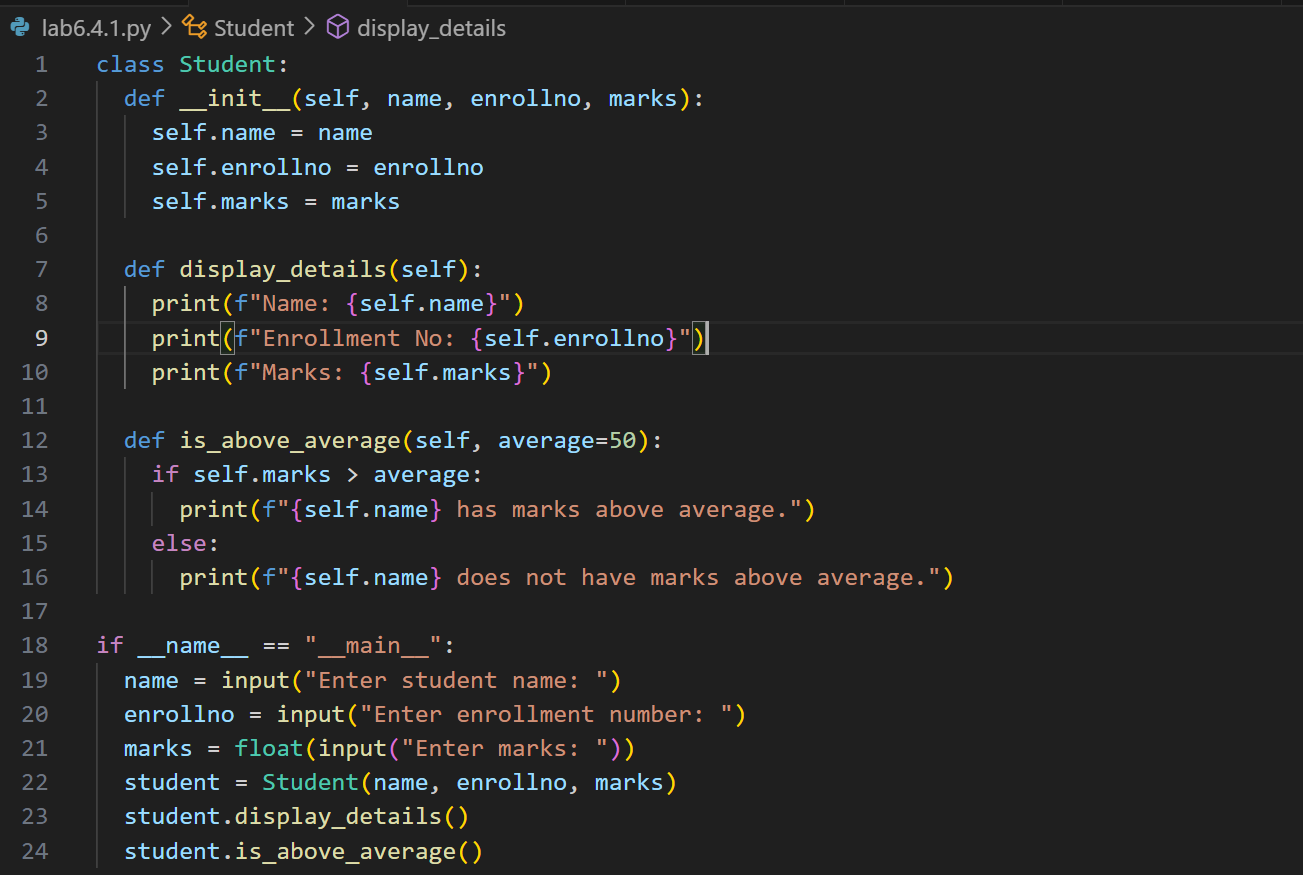
**TASK - 1:**

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.

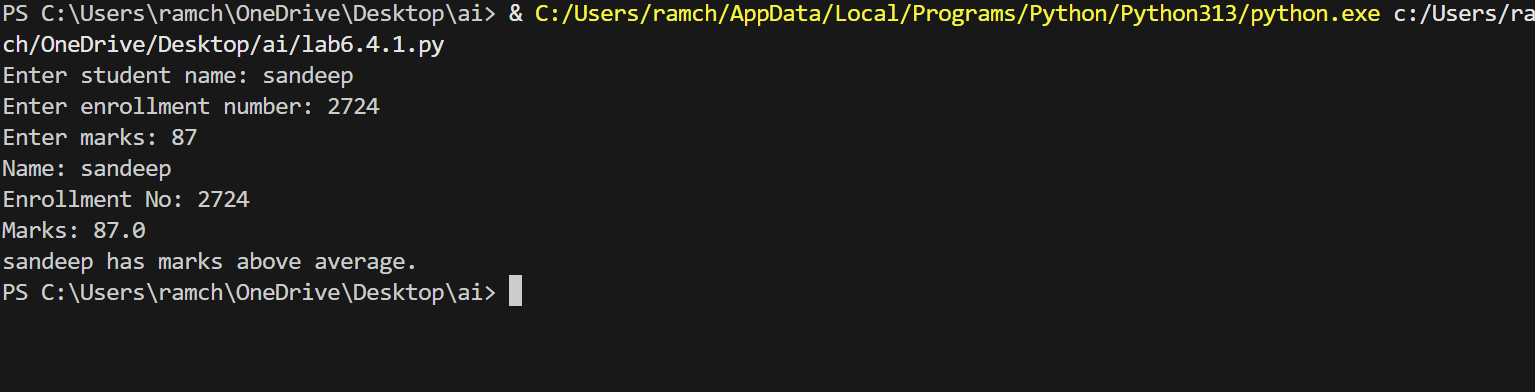
**Prompt:**

Generate a python class named Student with attributes name ,enrol number, marks and complete the methods to display details and check the marks are above average or not.

**Code:**



**Output**:



**Observation:**

The program defines a Student class with attributes name, enrolment number, and marks.  
It displays student details and checks if marks are above the average (default 50) using object-oriented concepts.

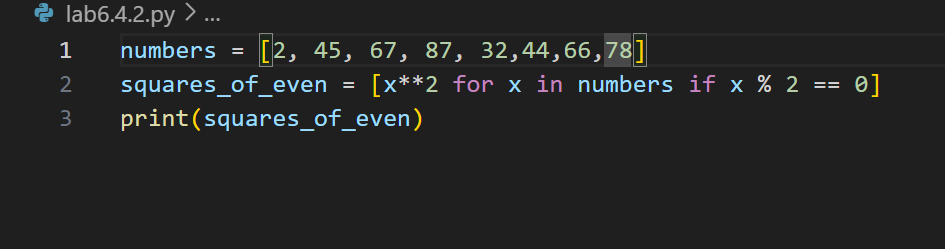
**Task - 2:**

Write the first two lines of 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.

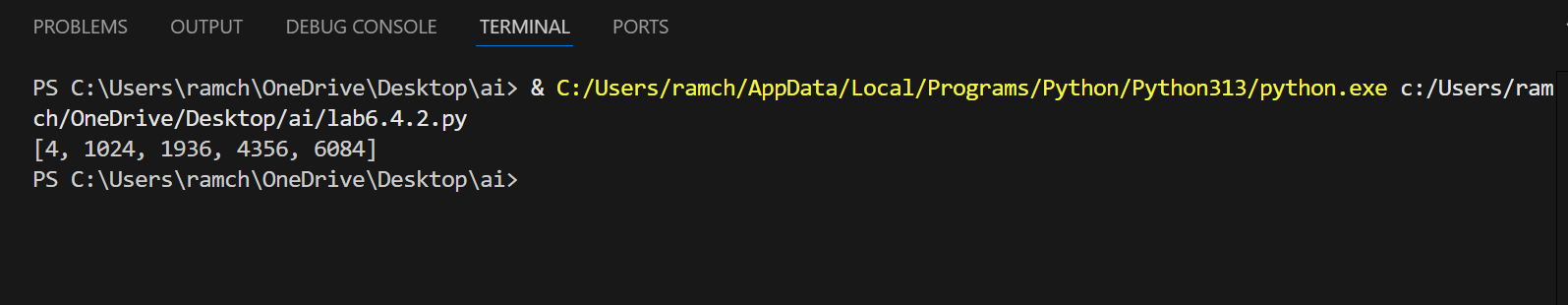
**Prompt:**

Generate a python code which finds the even numbers and calculates the squares of it.

**Code:**



**Output:**

****

**Observation:**

The code processes and finds the even numbers from the list and squares the numbers and stores them in the list. The Github copilot gave the appropriate logic for the given prompt.

**Task - 3:**

Create a class called BankAccount with attributes account\_holder and balance. Use Copilot to complete methods for deposit(), withdraw(), and check for insufficient balance.

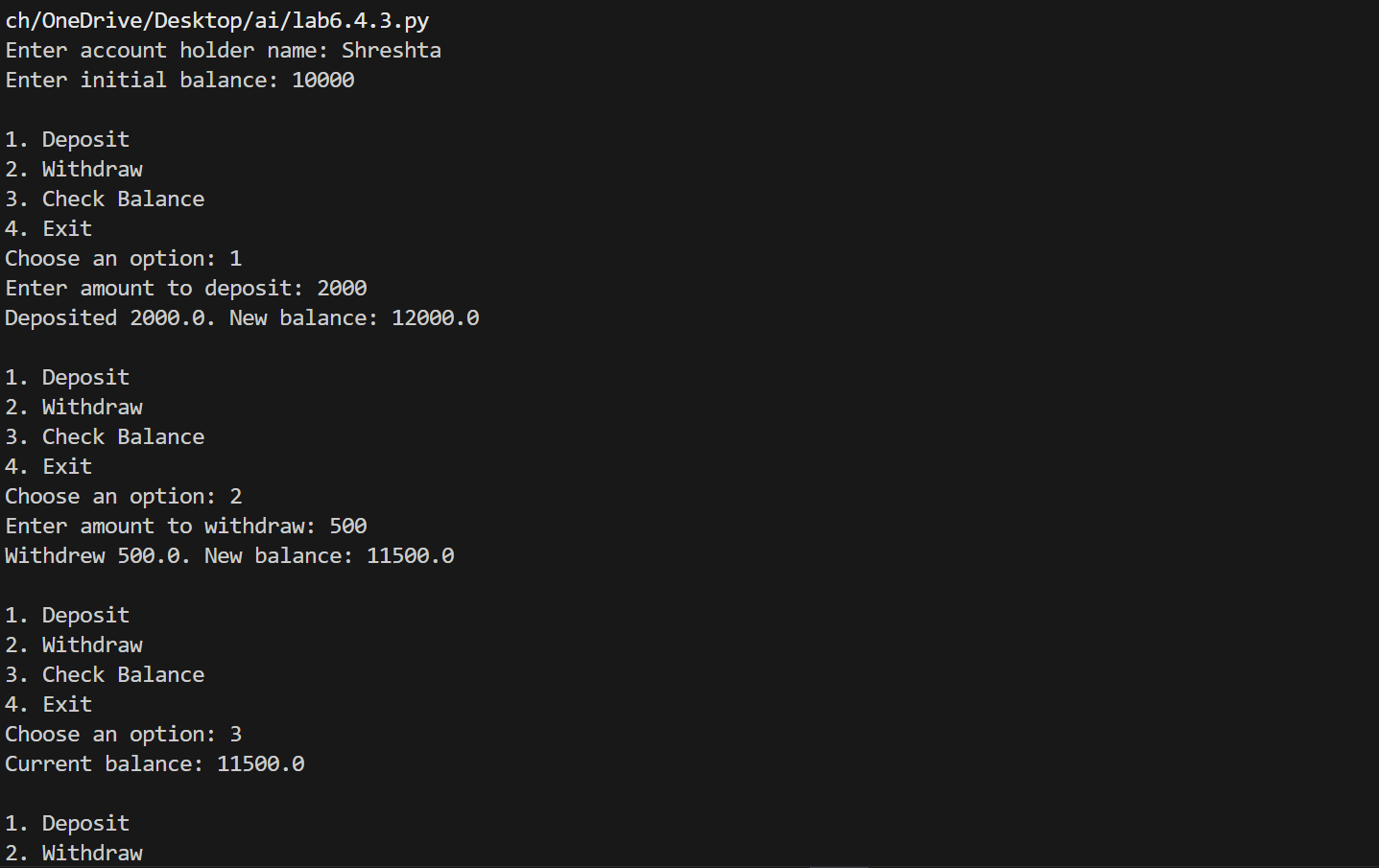
**Prompt**:

Generate a python code with the class Bank Account of attributes account holder and balance also generates methods for deposit, withdraw and check for insufficient balance.

**Code:**



**Output:**



**Observation:**

The code creates a class with name BankAccount and the methods to deposit, withdraw and check the insufficient balance with the menu driven application which takes the input from the user. The github copilot gave the appropriate code according to the scenario.

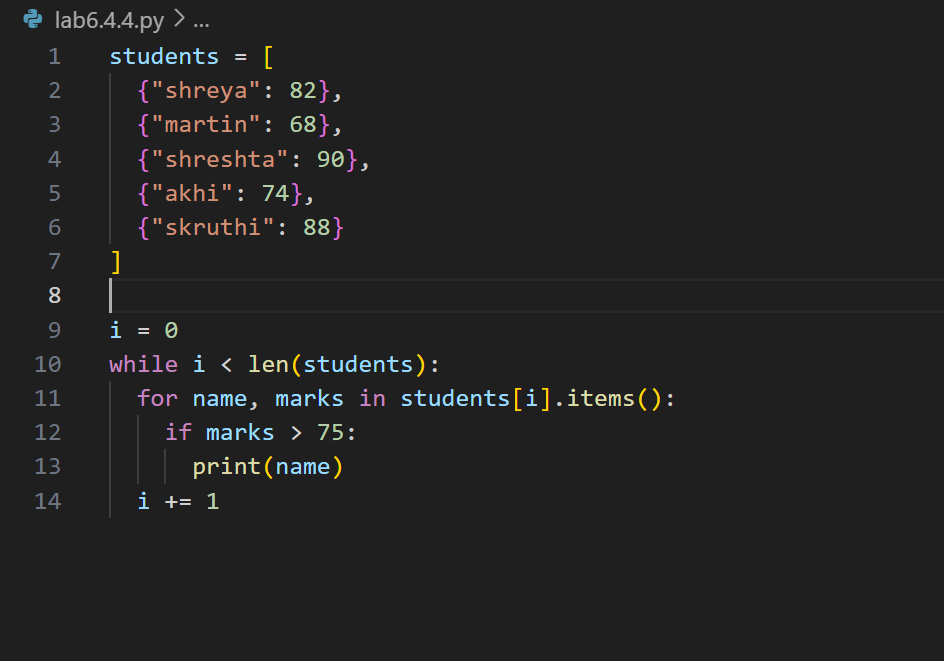
**Task - 4:**

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

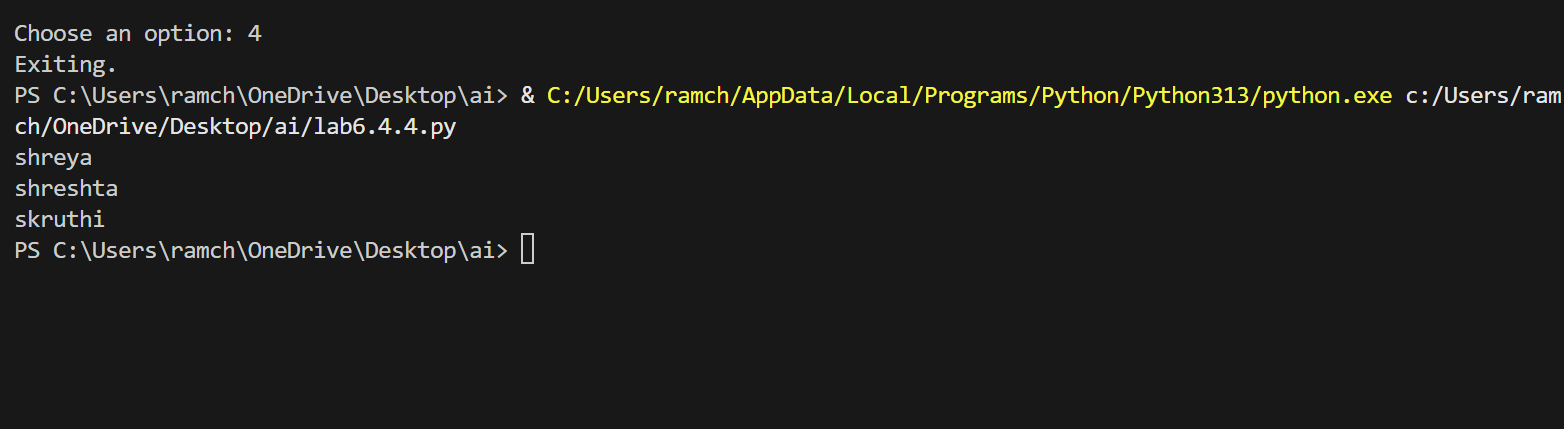
**Prompt:**

Use a while loop to iterate over the list of the dictionaries and print the students names whose marks are above 75.

**Code:**

****

**Output:**

****

**Observation:**

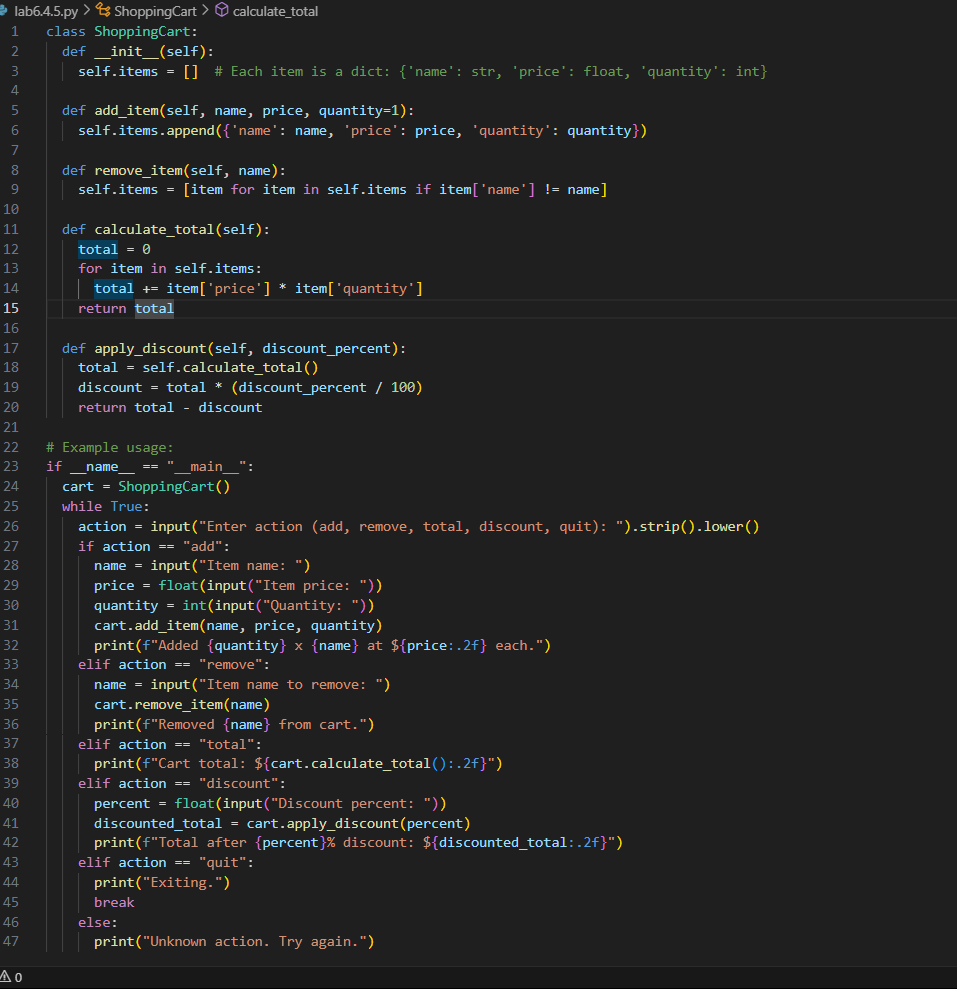
The code iterates over the list of dictionaries and prints the names of the students who got marks above 75. The Github copilot processed well.

**Task - 5:**  
Begin writing a class ShoppingCart with an empty items list. Prompt Copilot to generate methods to add item, remove\_item, and use a loop to calculate the total bill using conditional discounts.

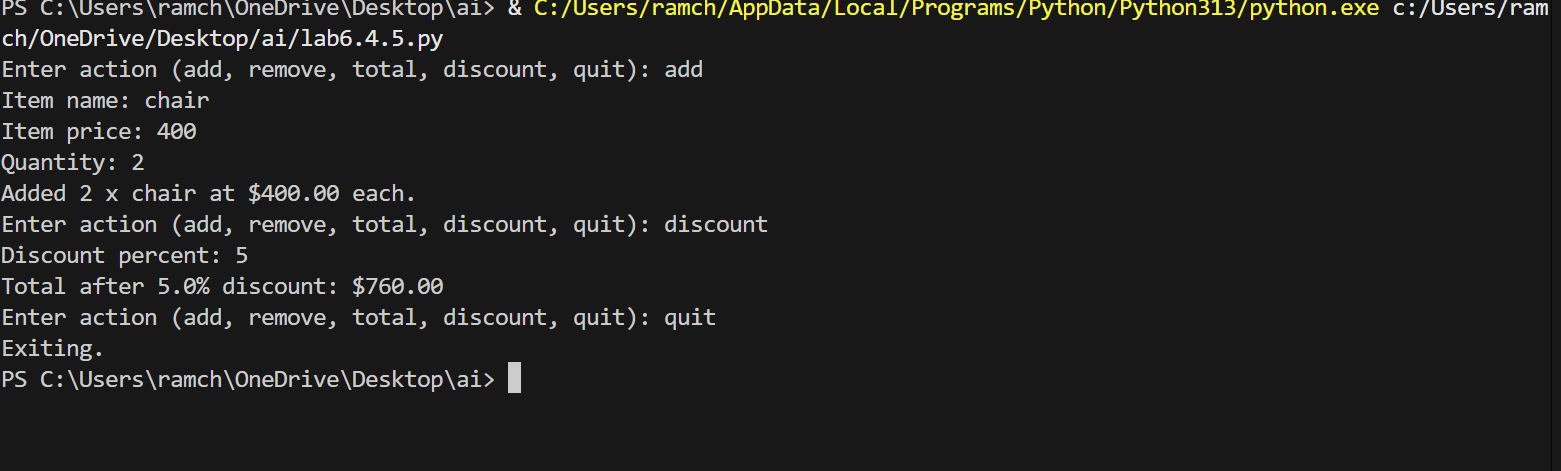
**Prompt:**

Generate a python class named shopping cart with empty lists then also generate the methods to add item, remove item, use loops to calculate bill applying the discount.

**Code:**

****

**Output:**

****

**Observation:**

The code creates a class named shopping cart with the empty lists and the methods to add, remove and calculating the bills applying discounts. The github copilot understood the scenario in a better way and processed the output well.