**LAB ASSIGNMENT-6.4**

**Task Description #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.

**Expected Outcome #1:**

**•** Completed class with Copilot-generated methods like display\_details() and is\_passed(), demonstrating use of if-else conditions.

Initial code:

class Student:

    def \_\_init\_\_(self, name, roll\_number, marks):

        self.name = name

        self.roll\_number = roll\_number

        self.marks = marks

Prompt:

generate function display\_details() to display the student details.generate is\_passed() function to check if the student's marks are above the passing average (e.g., 50). give dynamic inputs

Complete code:



**Task Description #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.

**Expected Outcome #2:**

**•** A complete loop generated by Copilot with conditional logic (if number % 2 == 0) and appropriate output

Initial Code:

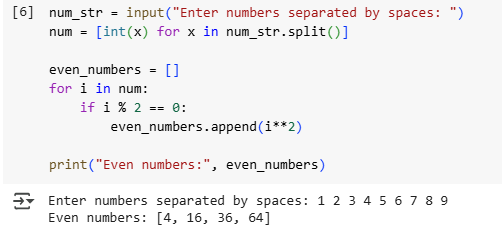
num= [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

for i in num:

Prompt:

Calculate the square of the even numbers and print them

COMPLETE CODE:



**Task Description #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.

**Expected Outcome #3:**

**•** Functional class with complete method definitions using if conditions and self attributes. Code should prevent overdrawing.

Initial code:

class BankAccount:

    def \_\_init\_\_(self, account\_holder, initial\_balance=0):

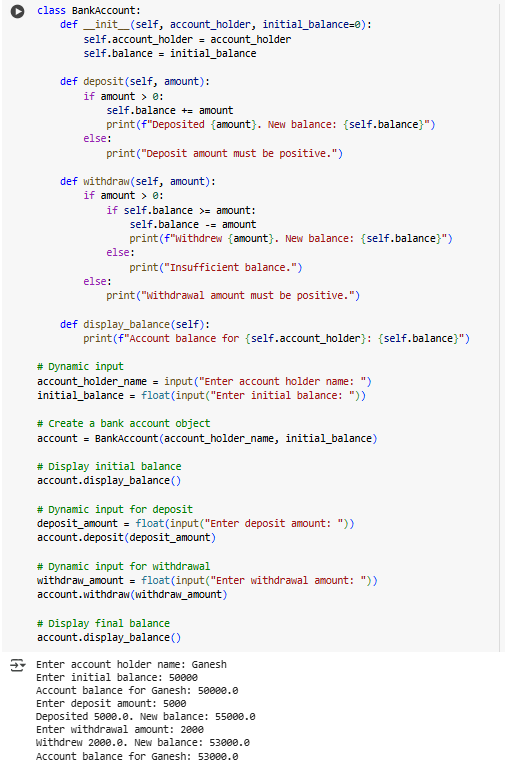
        self.account\_holder = account\_holder

        self.balance = initial\_balance

Prompt:

generate deposit(), withdraw(), and check for insufficient balance using if conditions and self attributes. give dynamic inputs

Complete code:



**Task Description #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.

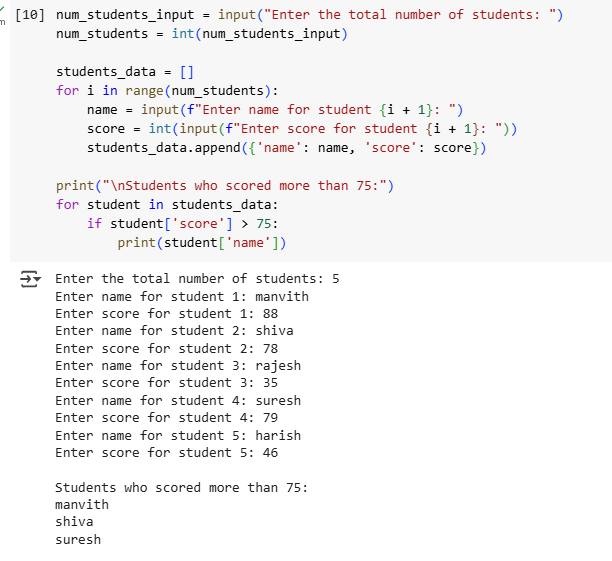
**Expected Outcome #4:**

**•** A complete while loop generated by Copilot with proper condition checks and formatted output.

Prompt:

write a while loop to print the names of students who scored more than 75 with dynamic inputs.

Complete Code:



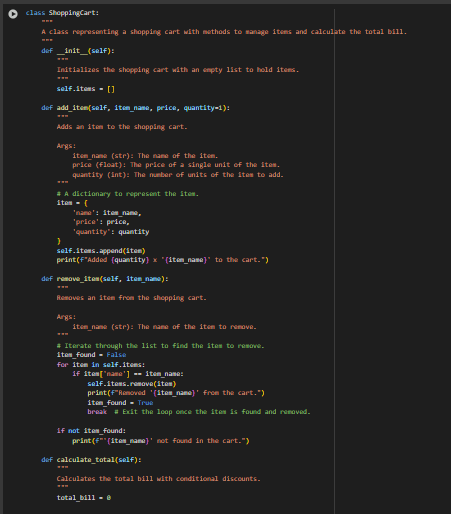
**Task Description #5:**

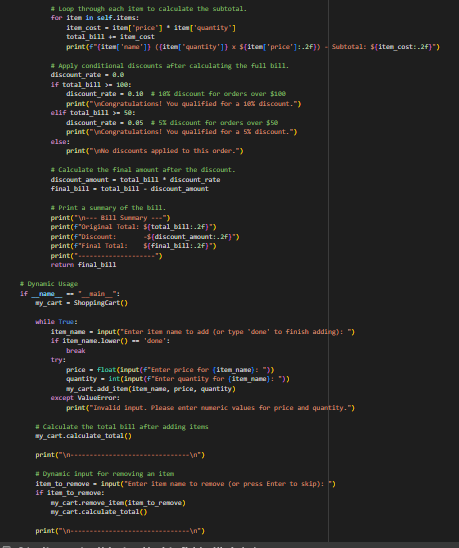
**•** Begin writing a class ShoppingCart with an empty items list. Prompt Copilot to generate methods to add\_item, remove\_item, and use a loop to calculate the total bill using conditional discounts.

**Expected Outcome #5:**

**•** A fully implemented ShoppingCart class with Copilot-generated loops and if-else statements handling item management and discount logic.

**COMPLETE CODE:**





**OUTPUT:**

