AI Lab-6.4 **Hall Ticket: 2403A51333**

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

Code: 

A screenshot of a computer

AI-generated content may be incorrect.

Observation:

The Student class was successfully defined with the required attributes. The method display\_details() displayed student information in a clear format. The method is\_passed() used an if- else condition to compare student marks against a threshold (default 40). The program demonstrated object-oriented programming (OOP) concepts and conditional logic effectively.

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.

Code:

A screenshot of a computer

AI-generated content may be incorrect.

Output:

A screenshot of a computer

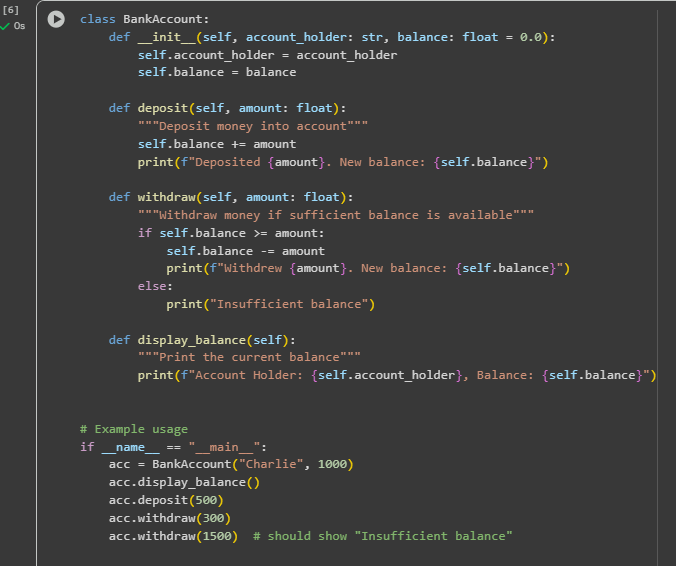
AI-generated content may be incorrect.

Observation:

The program iterated through a list of numbers using a for loop. A comment prompt was added,which guided Copilot to generate the missing logic. Copilot correctly suggested the if condition (if num % 2 == 0) to filter even numbers. It also generated the print statement to display the square of even numbers, while skipping odd numbers.

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.

Code:



Output:

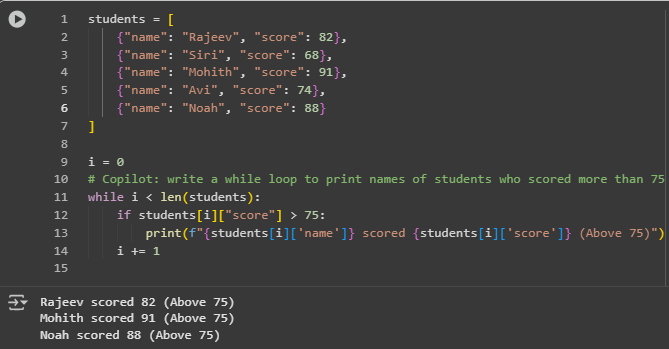
A screenshot of a computer program

AI-generated content may be incorrect.

Observation:

A BankAccount class was created with attributes account\_holder and balance. The method deposit() correctly increased the balance when a valid amount was provided. The method withdraw() used if conditions to check for valid withdrawals and prevented overdrawing when the requested amount exceeded the balance. The method check\_balance() displayed the current account holder and balance. The program demonstrated proper use of self attributes and conditional logic to manage banking operations safely.

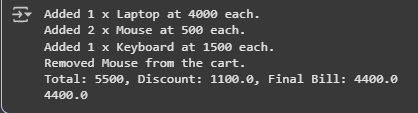
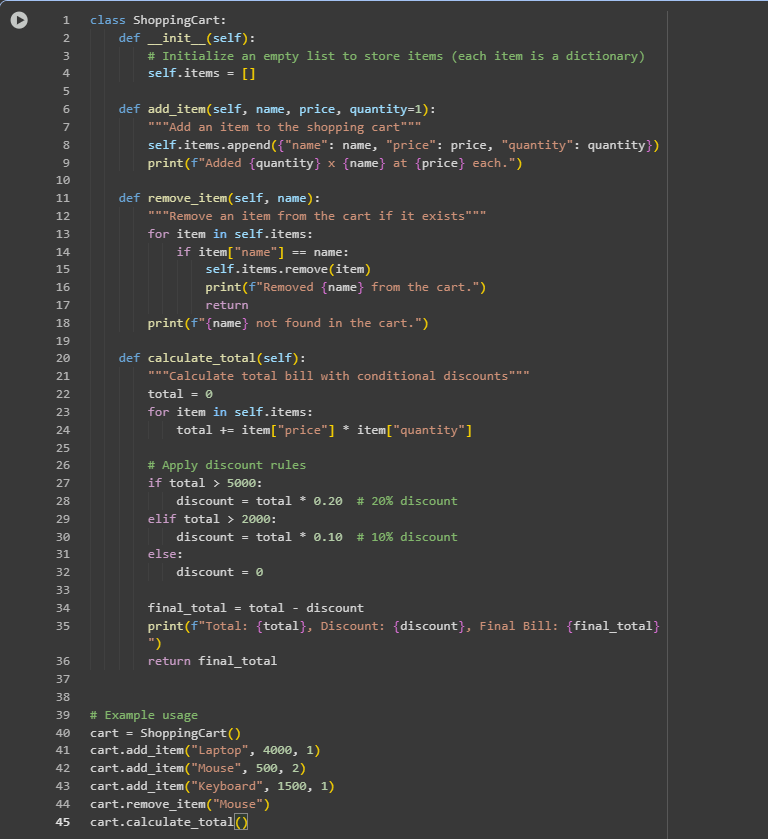
**Task Description -4:**  
• Define a list of student dictionaries with keys name and score. Ask Copilot to write a while loop to print the names of students who scored more than 75.



Observation:

A list of dictionaries was created with keys **name** and **score** for each student. A **while loop** was used to iterate through the list using an index variable. An **if condition** checked whether each student’s score was greater than 75. The loop successfully printed the names and scores of students who scored above the threshold, while skipping others. The task demonstrated the use of **dictionaries, loops, and conditional checks** together in Python.

**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.



Observation:

A Shopping Cart class was created with an empty **items list** to store product details. The method **add\_item()** allowed adding items with name, price, and quantity. The method **remove\_item()** successfully removed items from the list if they existed, otherwise displayed a not-found message. The method **calculate\_total()** iterated through the items list using a loop to calculate the total bill. **If-else conditions** were applied to provide **discounts** based on the total value (10% for totals above 2000 and 20% for totals above 5000). The program demonstrated effective use of **loops, conditionals, and object-oriented concepts** for item management and billing.