AI ASSISTED CODING

ASSIGNMENT-6.4

BATCH:04

ROLL NO:2403A52096

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.

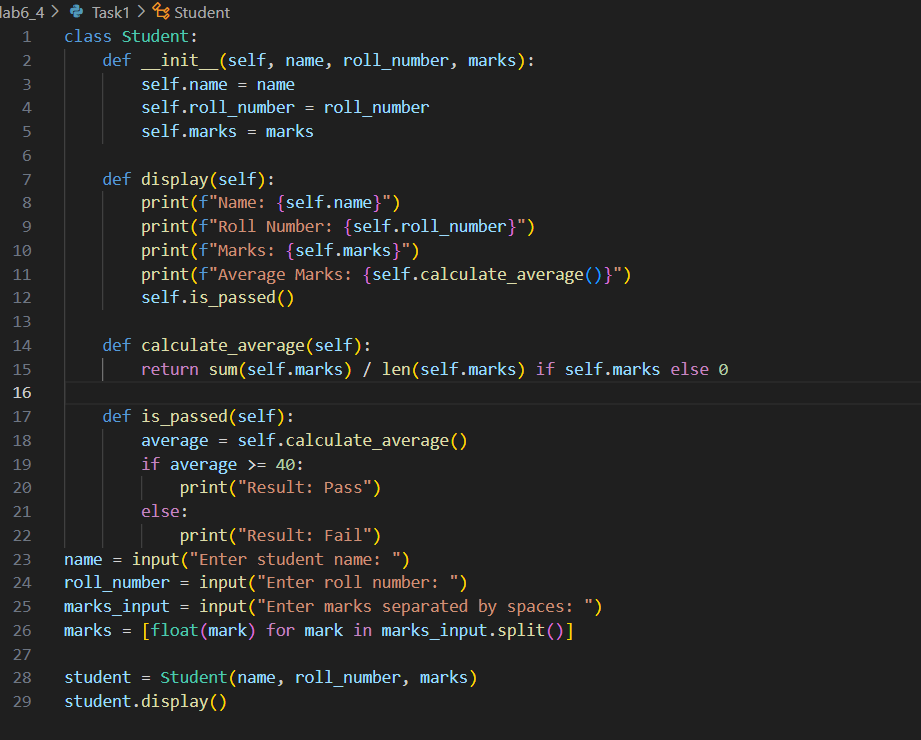
PROMPT:

1.Create a Python class named Student with attributes: name, roll\_number, and marks.

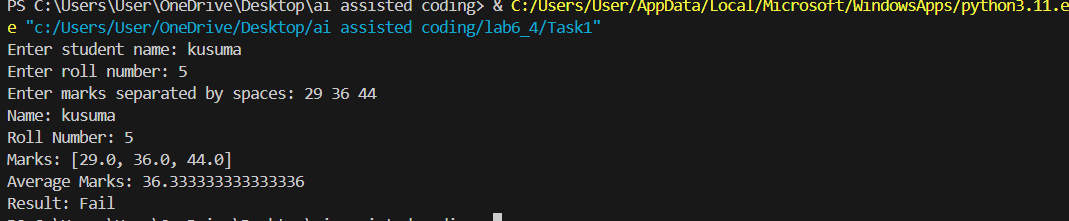
2.Add a method display\_details() to print all student information.

3.Add a method is\_passed() that checks if marks are above average (e.g., 40) and prints "Passed" or "Failed".

CODE:



OUTPUT:



OBSERVATION:

This code defines a student class that stores a student's name, roll number, and marks. It calculates the average marks, displays all details, and prints whether the student has passed (average ≥ 40) or failed. The program takes user input for student details and marks, then displays the results.

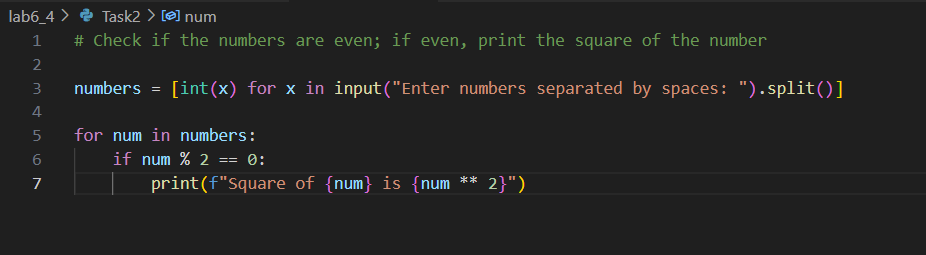
TASK2:

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.

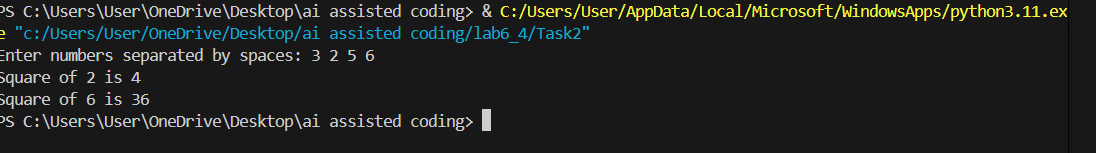
PROMPT:

write a code to print the square of the even numbers only.

CODE:



OUTPUT:



OBSERVATION:

This code reads a list of numbers from the user, checks each number to see if it is even, and prints the square of each even number. Odd numbers are skipped.

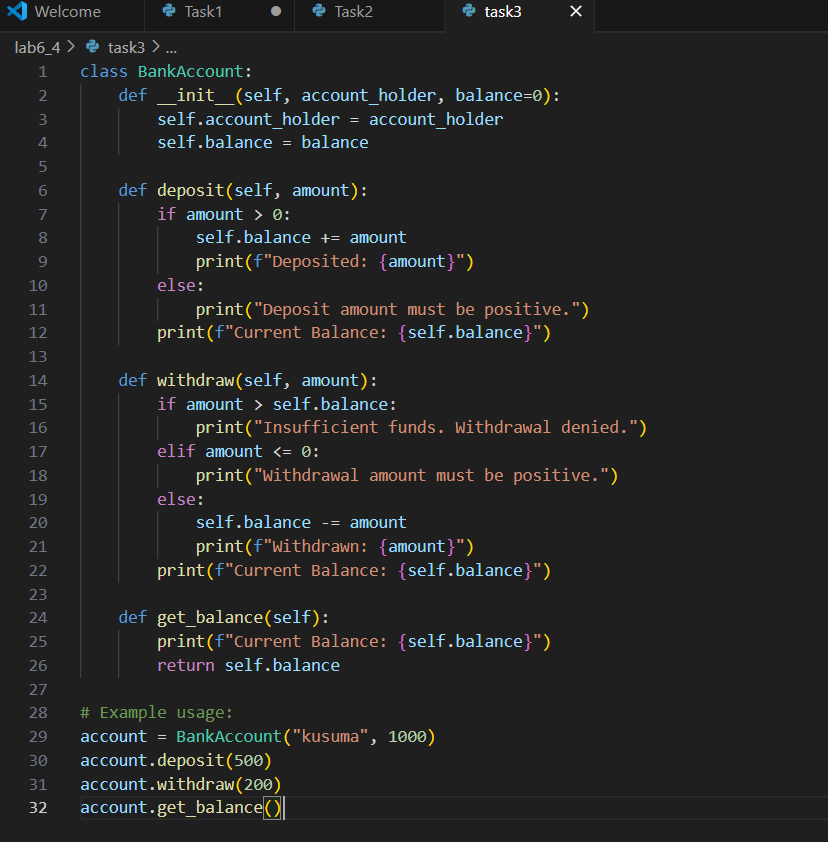
TASK3:

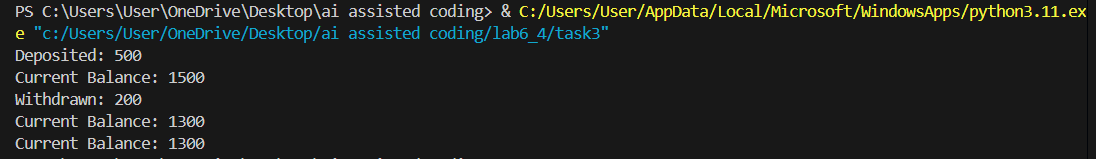
• Create a class called BankAccount with attributes account\_holder and balance. Use Copilot to complete methods for deposit(), withdraw(), and check for insufficient balance.

PROMPT:

Create a Python class called BankAccount with attributes account\_holder and balance. Implement methods deposit(amount), withdraw(amount), and get\_balance(). The withdraw method should check for insufficient funds and prevent overdrawing. Include print statements to show transaction results and current balance.

CODE:



OUTPUT:

OBSERVATION:

The code defines a BankAccount class with methods to deposit and withdraw money, check for insufficient funds, and display the current balance. It shows transaction results using print statements. An example demonstrates creating an account and performing transactions.

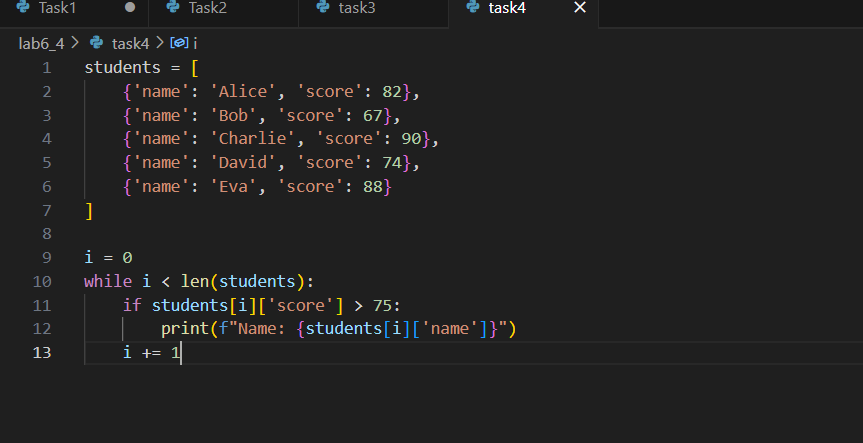
TASK4:

• 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.

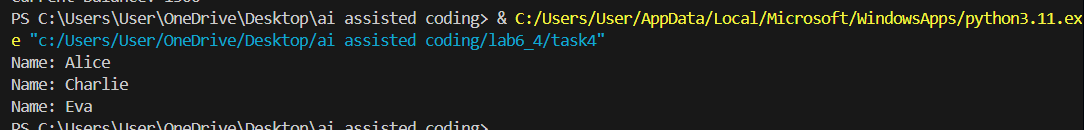
PROMPT:

Define a list of student dictionaries with keys 'name' and 'score'. Write a complete while loop to print the names of students who scored more than 75, with proper condition checks and formatted output.

CODE:



OUTPUT:



OBSERVATION:

The code defines a list of student dictionaries with names and scores. It uses a while loop to check each student's score, and prints the name of students who scored more than 75. The loop continues until all students are checked.

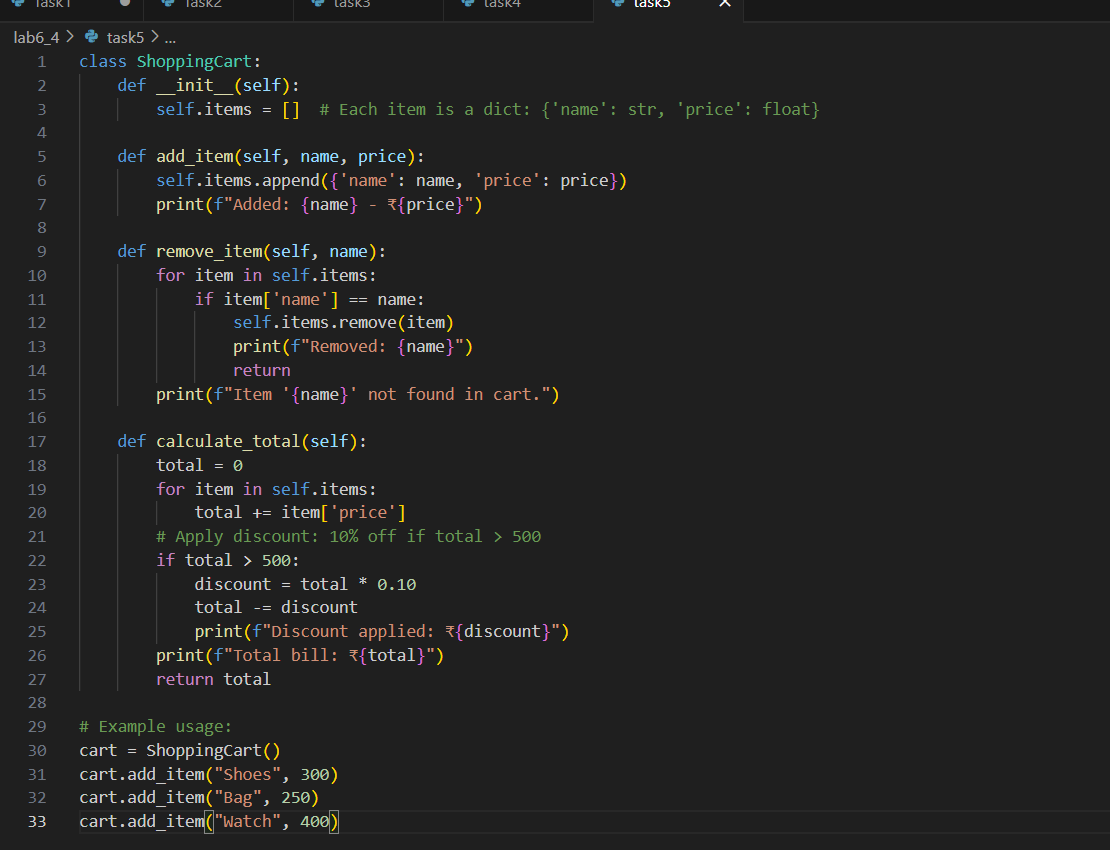
TASK5:

• 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.

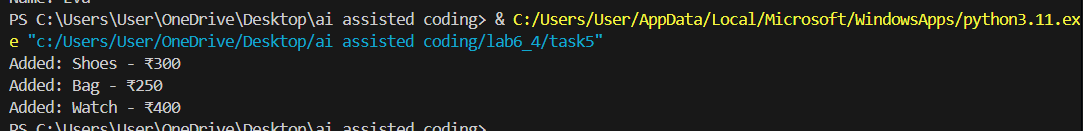
PROMPT:

write a python code to generate methods to add\_item, remove\_item, and use a loop to calculate the total bill using conditional discounts.

CODE:



OUTPUT:



OBSERVATION:

The code defines a ShoppingCart class with methods to add and remove items, and calculate the total bill. If the total exceeds 500, a 10% discount is applied. Example usage demonstrates adding and removing items, then calculating the final bill.