Lab 6: AI-Based Code Completion – Classes, Loops, and Conditionals

NAME – A. SHIVANI

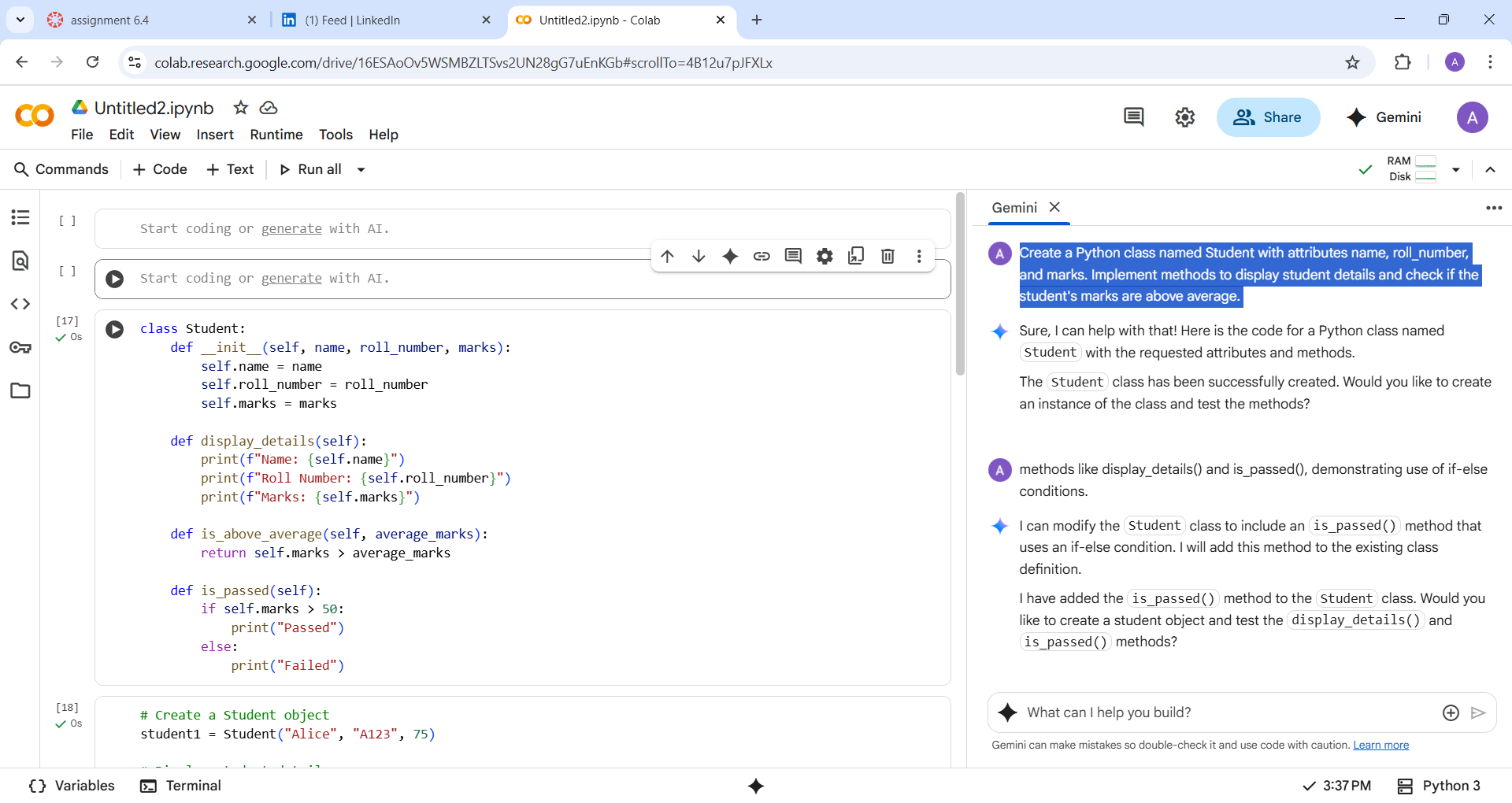
HTNO – 2403A51400

BATCH – 16

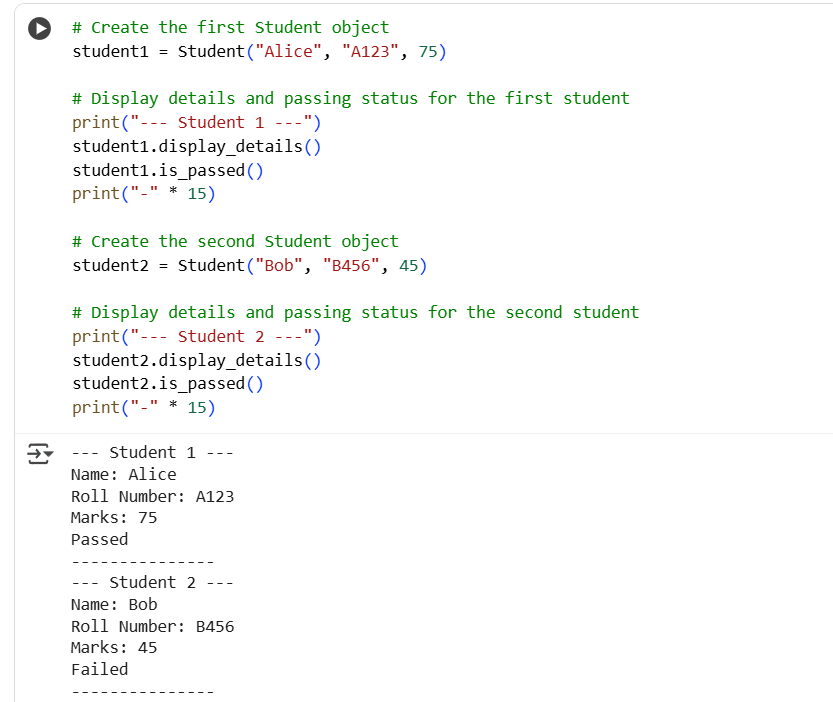
TASK 1

Create a Python class named Student with attributes name, roll\_number, and marks. Implement methods to display student details and check if the student's marks are above average.

CODE



OUTPUT



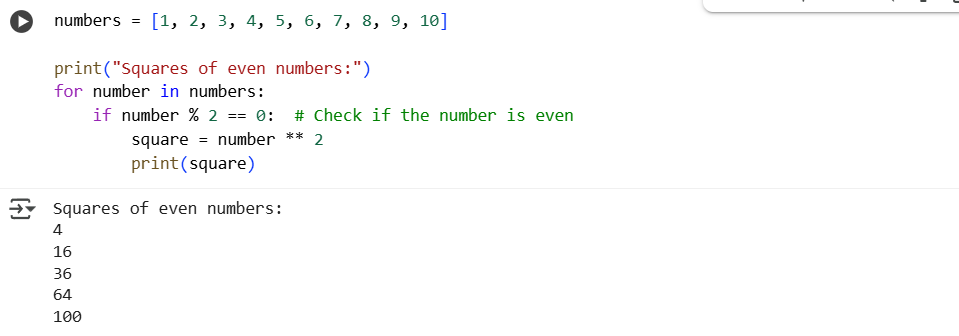
EXPLANATION

The code defines a Python class named Student. It has attributes: name, roll\_number, and marks. The \_\_init\_\_ method initializes these attributes when a new Student object is created. The display\_details method prints the student's name, roll number, and marks. The is\_above\_average method checks if the student's marks are greater than a given average. The is\_passed method checks if the student's marks are above 50. If marks are > 50, it prints "Passed"; otherwise, it prints "Failed". Two instances of the Student class are created: "Alice" with marks 75 and "Bob" with marks 45. The display\_details() and is\_passed() methods are called for both Alice and Bob. The output shows the details and passing status for each student.

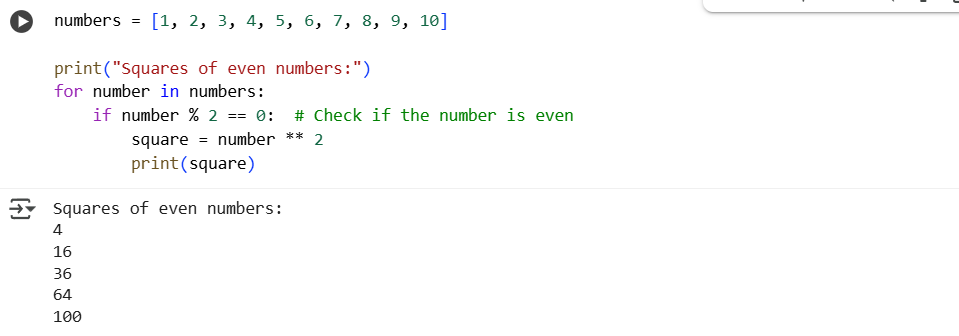
TASK2

write a python program to calculate and print the squares of even numbers only using the for loop to iterarate through a list of numbers

CODE



OUTPUT



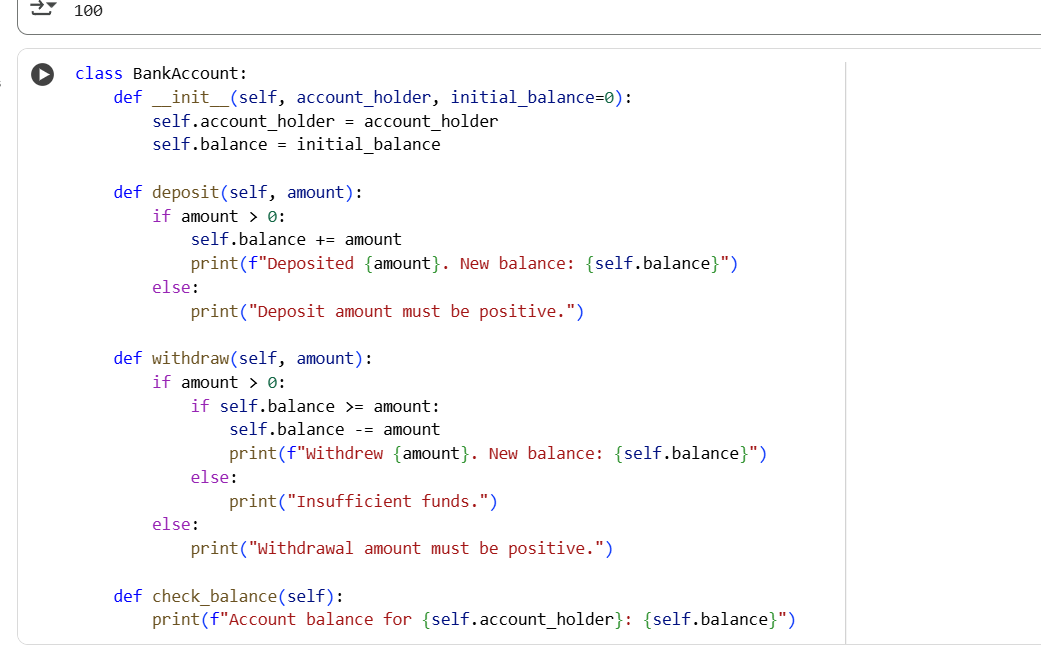
EXLANATION

This Python program iterates through a list of numbers using a for loop. For each number in the list, it checks if the number is even by using the modulus operator (%), which returns the remainder of the division of the number by 2. If the remainder is 0, the number is even, and the program calculates the square of the number by raising it to the power of 2 using the exponentiation operator (). It then prints out the number and its corresponding square. If the number is odd, the program simply moves on to the next number in the list without performing any calculations or printing anything. By doing so, the program efficiently filters out odd numbers and focuses solely on calculating and displaying the squares of even numbers, making it a concise and effective solution for this specific task. The output will include the squares of the even numbers 2, 4, 6, 8, and 10, which are 4, 16, 36, 64, and 100, respectively.

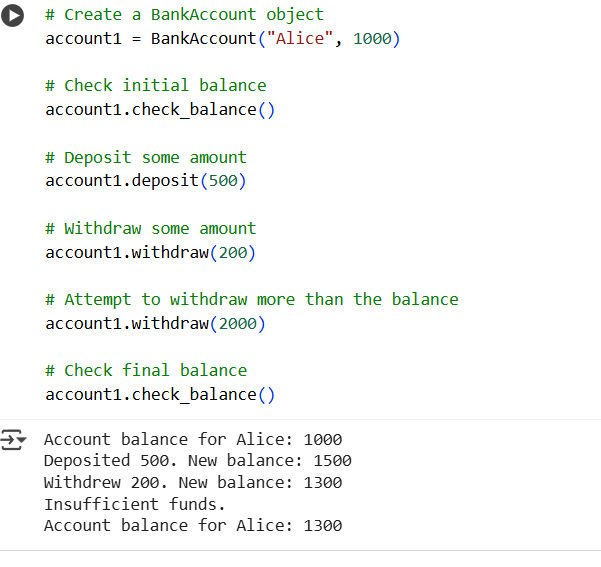
TASK3

Create a Python class named BankAccount with attributes account\_holder and balance. Implement methods deposit(), withdraw(), and check\_balance(). Ensure the withdraw() method checks for sufficient balance and prevents overdrawing.

Code



OUTPUT



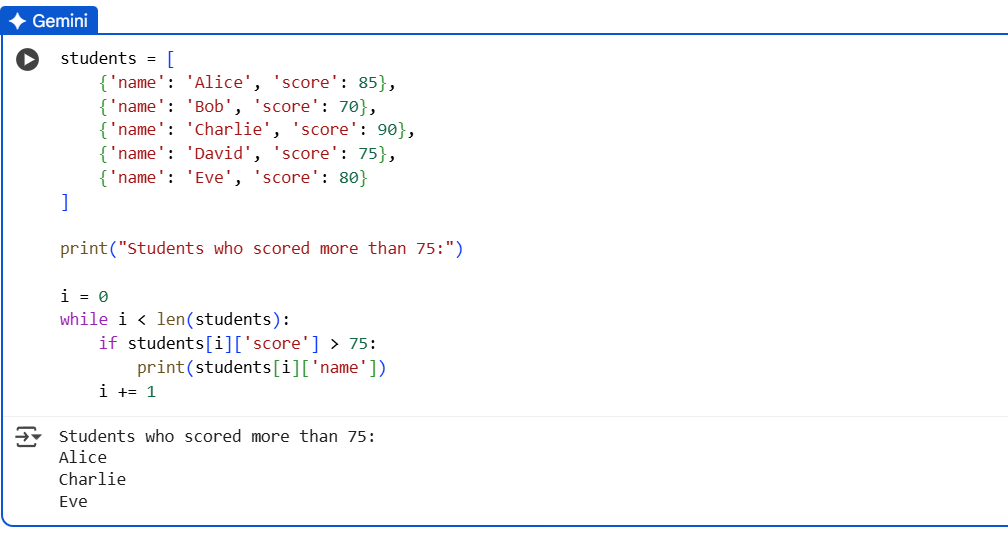
EXLANATION

The code you see defines a Python class called BankAccount, designed to simulate a basic bank account. The class is initialized with an account\_holder name and an optional initial\_balance, which defaults to 0 if not provided. Inside the class, there are three methods: deposit, withdraw, and check\_balance. The deposit method allows adding funds to the account; it checks if the deposit amount is positive before adding it to the current balance and then prints a confirmation message with the new balance. The withdraw method handles withdrawals; it first checks if the withdrawal amount is positive and then verifies if the account has sufficient funds (self.balance >= amount). If both conditions are met, the amount is deducted from the balance, and a confirmation message is printed. If there are insufficient funds, an "Insufficient funds" message is displayed. The check\_balance method simply prints the current balance for the account holder. The subsequent code demonstrates how to use this class: a BankAccount object named account1 is created for "Alice" with an initial balance of 1000. The initial balance is checked, then 500 is deposited, followed by a withdrawal of 200.

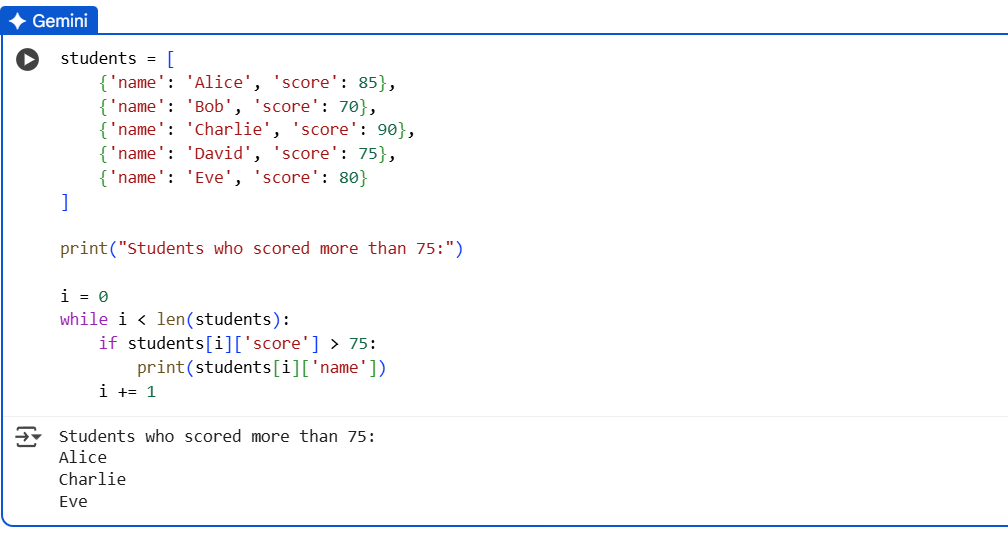
TASK4

Given 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.

CODE



OUTPUT



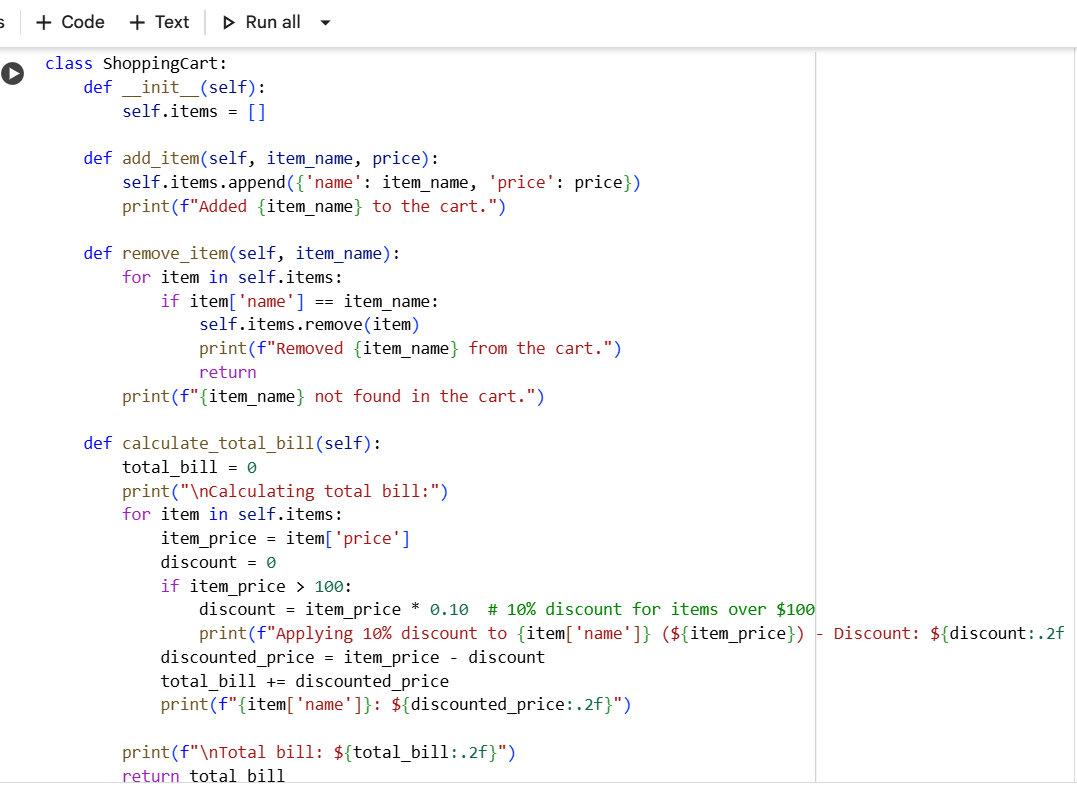
EXLANATION

The code initializes a list of dictionaries, where each dictionary holds a student's name and score. It then uses a while loop and an index i to iterate through this list. Inside the loop, it checks if the 'score' of the current student (accessed using students[i]['score']) is greater than 75. If the score meets this condition, the student's 'name' (accessed using students[i]['name']) is printed. The index i is incremented in each iteration to move to the next student until all students in the list have been checked.

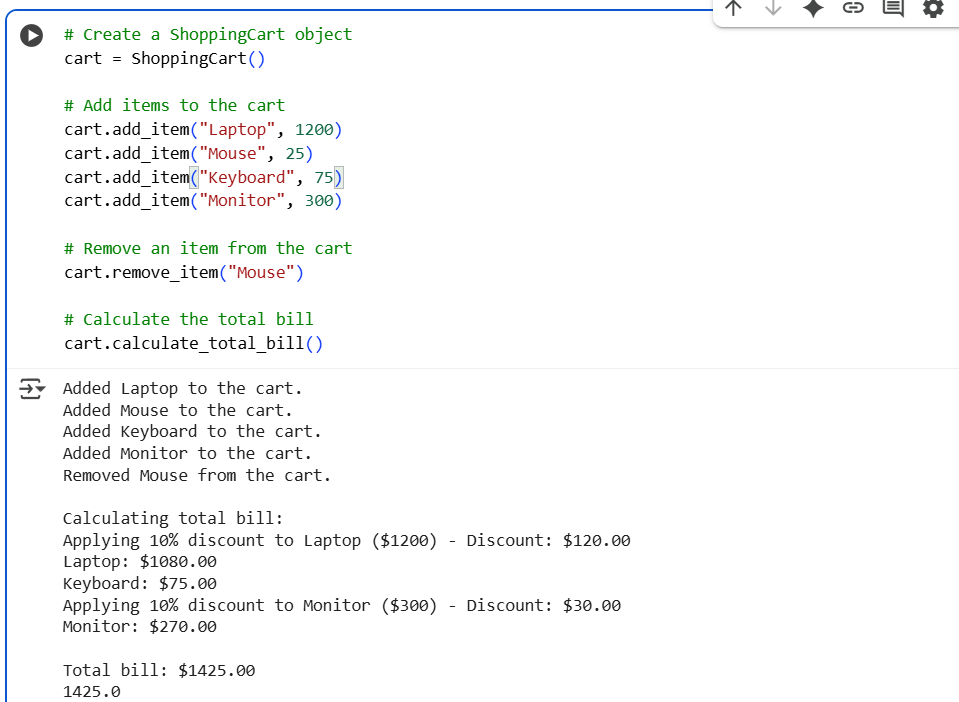
TASK5

Create a Python class ShoppingCart with an empty items list. Implement methods to add\_item, remove\_item, and calculate\_total\_bill. In calculate\_total\_bill, use a loop to iterate through items and apply conditional discounts (e.g., 10% off for items over $100).

CODE



OUTPUT



EXLANATION

The code I just ran demonstrates how to use the ShoppingCart class we defined earlier. It starts by creating an instance of the ShoppingCart class, effectively initializing an empty shopping cart. Then, it adds several items to the cart using the add\_item() method, specifying the item's name and price for each. This includes items both above and below the $100 discount threshold to showcase the conditional discount. After adding items, the code calls the `remove\_item()` method to take a specific item, "Mouse," out of the cart. Finally, it invokes the `calculate\_total\_bill()` method. This method iterates through the items currently in the cart, applies a 10% discount to any item priced over $$100 discount threshold to showcase the conditional discount. After adding items, the code calls the `remove\_item()` method to take a specific item, "Mouse," out of the cart. Finally, it invokes the `calculate\_total\_bill()` method. This method iterates through the items currently in the cart, applies a 10% discount to any item priced over $100, prints the details of the calculation for each item (including the applied discount), and then outputs the final total bill for all items remaining in the cart. The output you saw reflects these steps: confirming items added, confirming the item removed, showing the discounted prices for eligible items, and displaying the final calculated total.