LAB ASSINGNMENT – WEEK 6.8

NAME : KARTHIK RAM .D

HALL.NO : 2403A52384

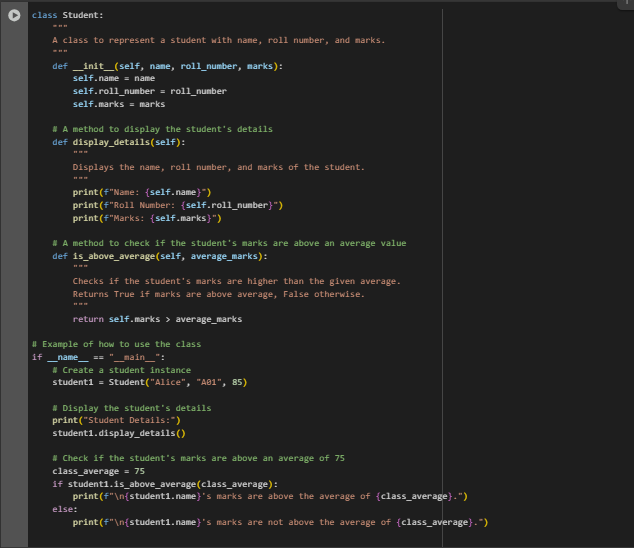
BATCH.NO : AI 14

TOOL USED : GEMINI AND PERPLEXITY AI

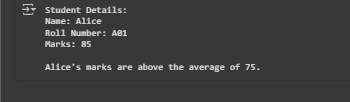
**PROMPT 01 :**

I need a python code to Start a Python class named Student with attributes name, roll\_number, and marks.

**Code :**

****

**OUTPUT :**

****

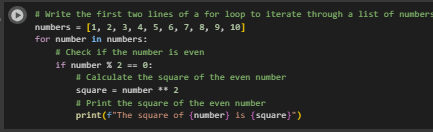
**CODE EXPLANATION :**

* Class as a Blueprint: The class Student: line defines a blueprint for creating "Student" objects. Each student object will have its own set of data (attributes) and behaviors (methods) based on this blueprint.
* The Constructor (\_\_init\_\_):
  + This special method is called automatically whenever a new Student object is created.
  + Its job is to initialize the object's attributes: name, roll\_number, and marks. The self keyword refers to the specific instance of the student being created.
* Methods (Behaviors):
  + display\_details(): This method defines a behavior for the Student object. When called, it prints out the student's information in a readable format.
  + is\_above\_average(average\_marks): This method takes an average\_marks value as input, compares it to the student's own marks, and returns either True or False.
* Docstrings (Documentation): The text enclosed in triple quotes ("""...""") at the beginning of the class and each method is a docstring. It explains the purpose of the code and is a best practice for making code understandable to others.
* Example Usage Block (if \_\_name\_\_ == "\_\_main\_\_":):
  + This block demonstrates how to use the Student class. It only runs when the script is executed directly (not when imported as a module).
  + It shows how to create a Student object (student1), call its methods, and use the results**.**

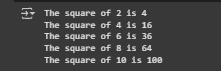
**PROMPT 02 :**

I NEED A PYTHON CODE THAT Write the first two lines of a for loop to iterate through a list of numbers AND how to calculate and print the square of even numbers only

**CODE :**

****

**OUTPUT :**

****

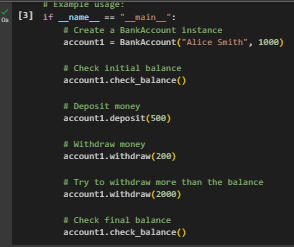
**CODE EXPLANATION :**

* Iteration: The for number in numbers: line iterates through each element in the numbers list, assigning the current element to the variable number in each loop.
* Even Number Check: The if number % 2 == 0: line checks if the current number is even by using the modulo operator (%). If the remainder of dividing the number by 2 is 0, the number is even.
* Square Calculation: If the number is even, square = number \*\* 2 calculates the square of the number by raising it to the power of 2.
* Printing the Result: Finally, print(f"The square of {number} is {square}") prints the calculated square along with the original even number using an f-string for formatted output.

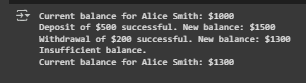
**PROMPT 03 :**i need a python code that Create a class called BankAccount with attributes account\_holder and balance and complete methods for deposit(), withdraw(), and check for insufficient balance.

**CODE :**

****

****

**OUT PUT :**

****

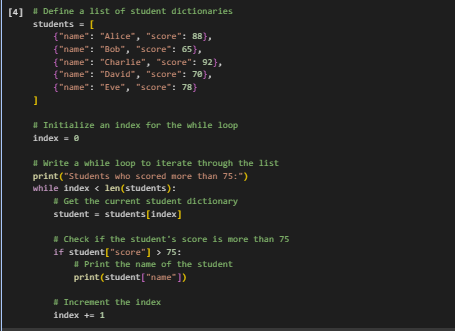
**CODE EXPLANATION :**

* Class Definition: The code defines a class named BankAccount to model a bank account.
* Constructor (\_\_init\_\_): This method is called when you create a new BankAccount object. It initializes the account\_holder and sets the initial balance (defaulting to 0 if not provided).
* deposit() Method: This method takes an amount and adds it to the balance. It includes a check to ensure the deposit amount is positive.
* withdraw() Method: This method takes an amount and subtracts it from the balance. It includes a check to ensure the withdrawal amount is positive and another check to prevent withdrawing more than the current balance (insufficient balance).
* check\_balance() Method: This method simply prints the current account\_holder's name and their balance.
* Example Usage (if \_\_name\_\_ == "\_\_main\_\_":): This block demonstrates how to create a BankAccount object and call its methods to perform operations like checking the balance, depositing, and withdrawing.

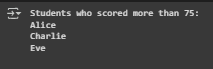
**PROMPT 04 :**

I need a python code that Define a list of student dictionaries with keys name and score and write a while loop to print the names of students who scored more than 75.

**CODE :**

****

**OUTPUT :**

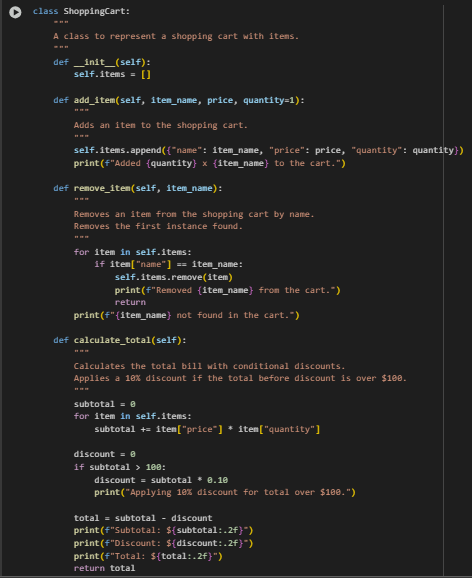
****

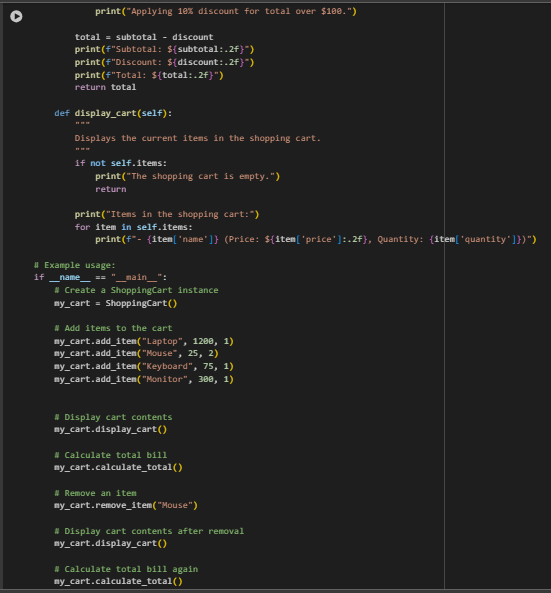
**CODE EXPPLANATION :**

* List of Dictionaries: The code starts by creating a list called students, where each item in the list is a dictionary representing a student with 'name' and 'score' keys.
* Initializing Index: An integer variable index is initialized to 0. This will be used to keep track of the current position in the students list.
* While Loop: The while index < len(students): line starts a while loop that will continue as long as the index is less than the total number of elements in the students list.
* Accessing Student Data: Inside the loop, student = students[index] retrieves the dictionary at the current index from the students list and assigns it to the variable student.
* Conditional Check: The if student["score"] > 75: line checks if the value associated with the 'score' key in the current student dictionary is greater than 75.
* Printing Name: If the score is greater than 75, print(student["name"]) prints the value associated with the 'name' key in the current student dictionary.
* Incrementing Index: index += 1 increases the value of index by 1 after each iteration, moving to the next student in the list.

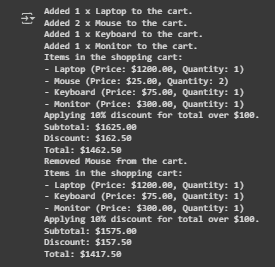
**PROMPT 05 :**I NEED A PYHTON CODE that Begin writing a class ShoppingCart with an empty items list and generate methods to add\_item, remove\_item, and use a loop to calculate the total bill using conditional discounts.

**CODE :**

****

****

**OUTPUT :**

****

**CODE EXPLANANTION :**

* Class Definition: The code defines a class named ShoppingCart to represent a shopping cart.
* Constructor (\_\_init\_\_): This method is called when you create a new ShoppingCart object. It initializes an empty list called items to store the items in the cart.
* add\_item() Method: This method takes item\_name, price, and optional quantity as input and appends a dictionary representing the item to the items list.
* remove\_item() Method: This method takes an item\_name and removes the first occurrence of that item from the items list if found.
* calculate\_total() Method: This method iterates through the items list to calculate the subtotal. It then checks if the subtotal is greater than $100 and applies a 10% discount if it is. Finally, it calculates and prints the total bill.
* display\_cart() Method: This method checks if the items list is empty. If not, it iterates through the list and prints the details of each item in the cart.
* Example Usage (if \_\_name\_\_ == "\_\_main\_\_":): This block demonstrates how to create a ShoppingCart object, add items, display the cart, calculate the total, remove an item, and then display the cart and calculate the total again.