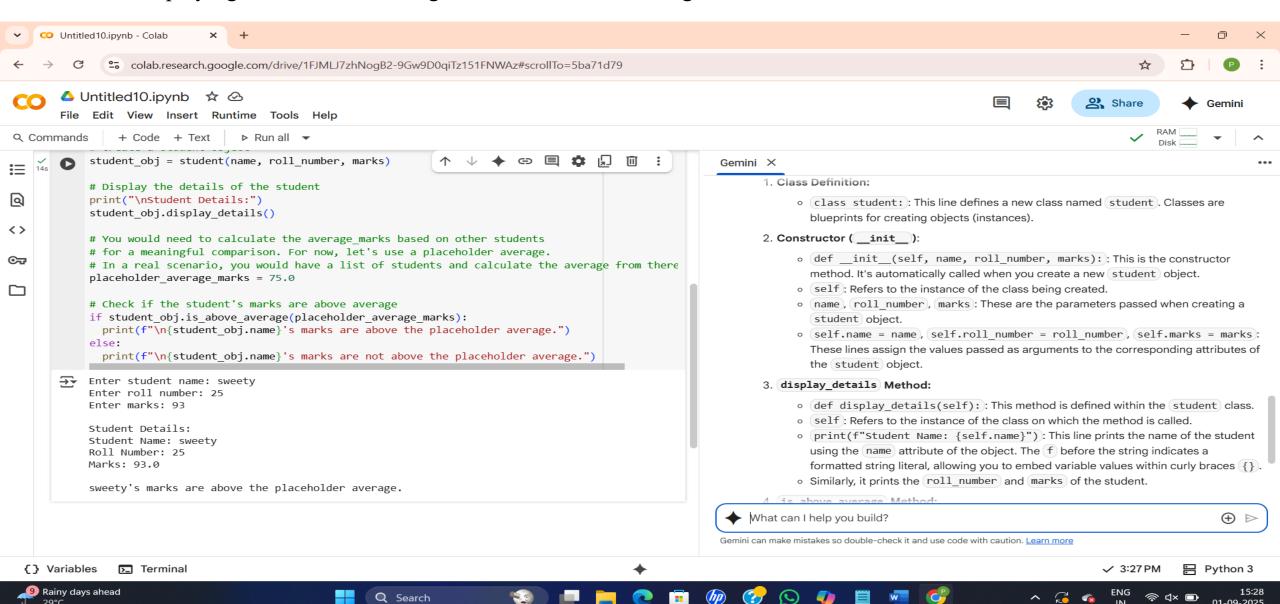
# AI ASSITED CODING

LAB-6

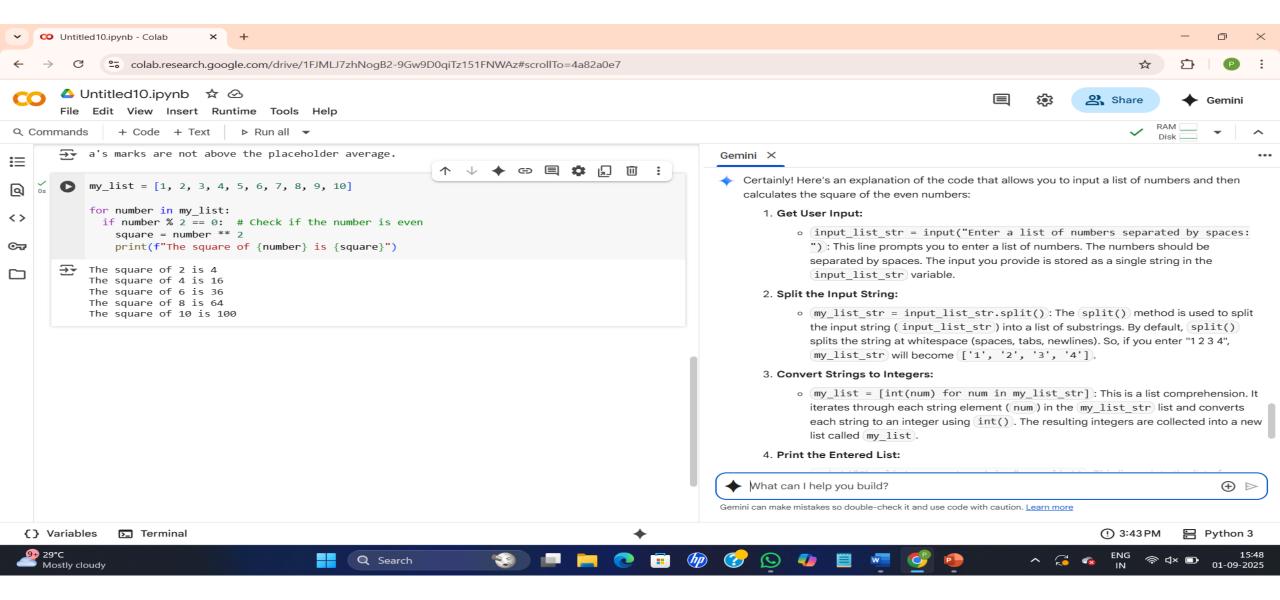
## **Task Description #1:**

• 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.



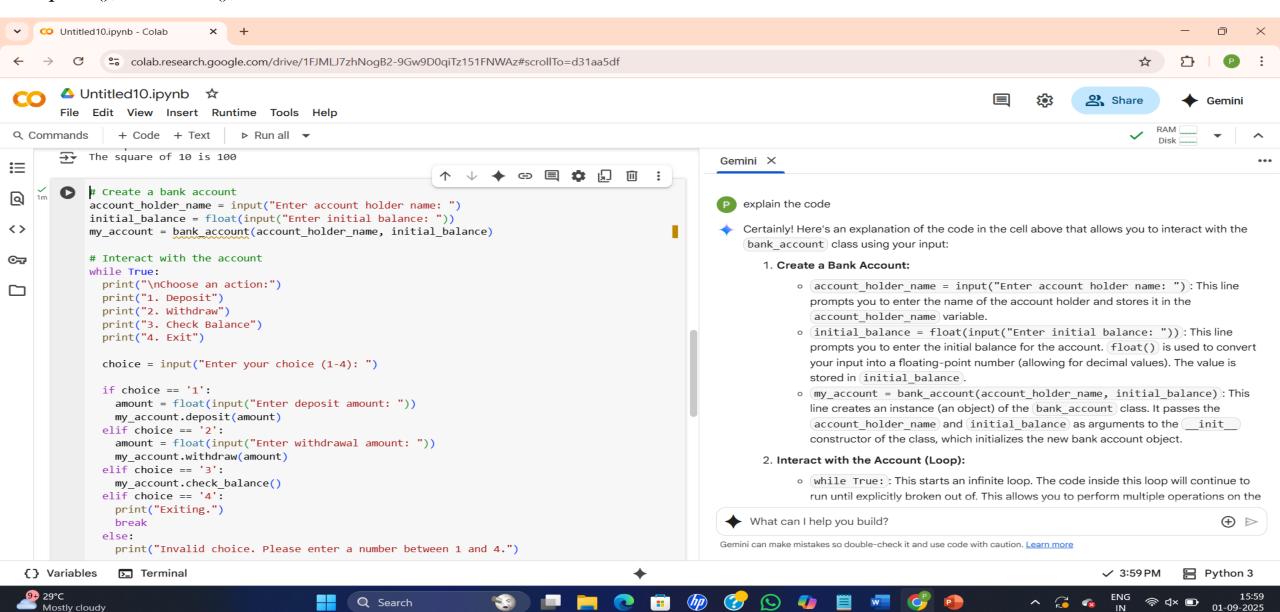
#### **Task Description #2:**

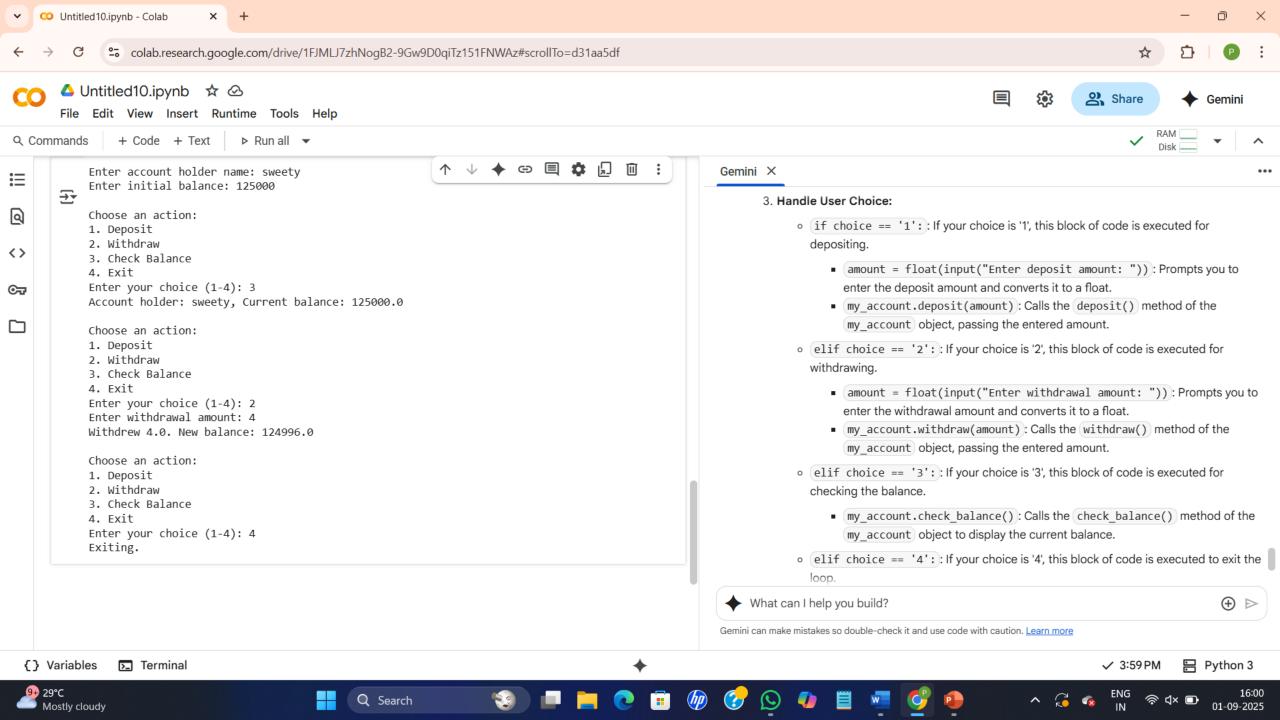
• Write the first two lines of a for loop to iterate through a list of numbers. Use a comment prompt to let Copilot suggest how to calculate and print the square of even numbers only.



## **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.

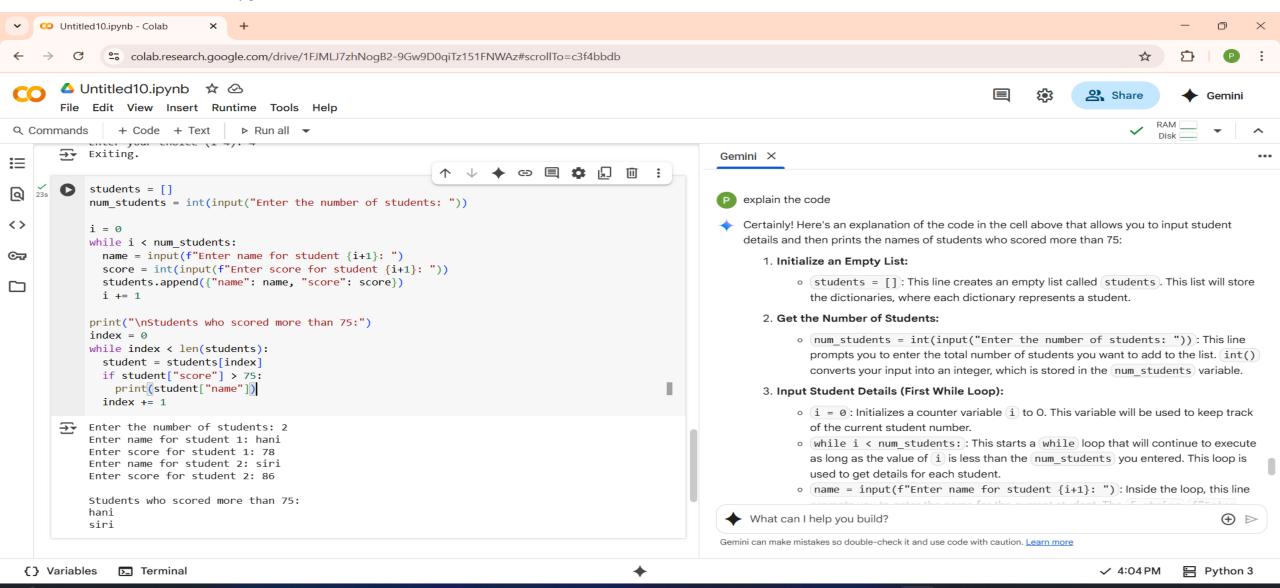




## **Task Description #4:**

Mostly cloudy

• 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.



01-09-2025

Q Search

#### **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.

```
# Create a shopping cart
     my_cart = shopping_cart()
     # Add items to the cart
     while True:
      item name = input("Enter item name to add (or 'done' to finish): ")
      if item name.lower() == 'done':
        break
       try:
        price = float(input(f"Enter price for {item name}: "))
         my cart.add item(item name, price)
       except ValueError:
         print("Invalid price. Please enter a number.")
     print("\nItems in your cart:")
     for item in my_cart.items:
      print(f"- {item['name']}: ${item['price']:.2f}")
Finter item name to add (or 'done' to finish): eraser
     Enter price for eraser: 10
     Added eraser to the cart.
     Enter item name to add (or 'done' to finish): pen
     Enter price for pen: 30
     Added pen to the cart.
     Enter item name to add (or 'done' to finish): sharpener
     Enter price for sharpener: 5
     Added sharpener to the cart.
     Enter item name to add (or 'done' to finish): done
     Items in your cart:
     - eraser: $10.00
     - pen: $30.00
     - sharpener: $5.00
```

- explain the code
- Certainly! Here's an explanation of the code in the cell above that allows you to add items to your shopping cart using your input:

#### 1. Create a Shopping Cart:

o (my\_cart = shopping\_cart()): This line creates an instance (an object) of the (shopping\_cart) class. This object will represent your shopping cart.

#### 2. Add Items to the Cart (Loop):

- while True: This starts an infinite loop, allowing you to add multiple items to the cart until you decide to stop.
- item\_name = input("Enter item name to add (or 'done' to finish): "): This line prompts you to enter the name of the item you want to add to the cart. It also tells you to type 'done' to finish adding items. Your input is stored in the item name variable.
- o (if item\_name.lower() == 'done': This checks if you entered 'done' (case-insensitive). If you did, the break statement is executed, which exits the while loop.
- try...except ValueError: This is a try-except block used for error handling. It attempts to execute the code within the try block. If a ValueError occurs (which would happen if you enter something that cannot be converted to a float for the price), the code within the except block is executed.
  - (price = float(input(f"Enter price for {item\_name}: ")): Inside the (try) block, this line prompts you to enter the price for the item you just named. (float()) converts your input to a floating-point number.
  - my\_cart.add\_item(item\_name, price): This line calls the add\_item() method of the my\_cart object, passing the item\_name and price you entered as arguments. This adds the item to the cart's internal list.
  - [print("Invalid price. Please enter a number."): Inside the [except] block, this line prints an error message if the price you entered was not a valid number.

#### 3. Print Items in the Cart:

- o print("\nItems in your cart:"): After the loop finishes (when you enter 'done'), this line prints a header indicating the items that are currently in your cart.
- o for item in my\_cart.items: This starts a for loop that iterates through each dictionary (representing an item) in the items list of the my\_cart object.
- print(f"- {item['name']}: \${item['price']:.2f}"): Inside this loop, this line prints the name and price of each item in the cart. item['name'] accesses the name from the dictionary, and item['price'] accesses the price. :.2f formats the price to two decimal places.

In summary, this code allows you to interactively build a shopping cart by repeatedly entering item names and prices until you type 'done'. It then prints a list of the items you added to the cart.