ASSIGNMENT:6.5

NAME: CH. Divya

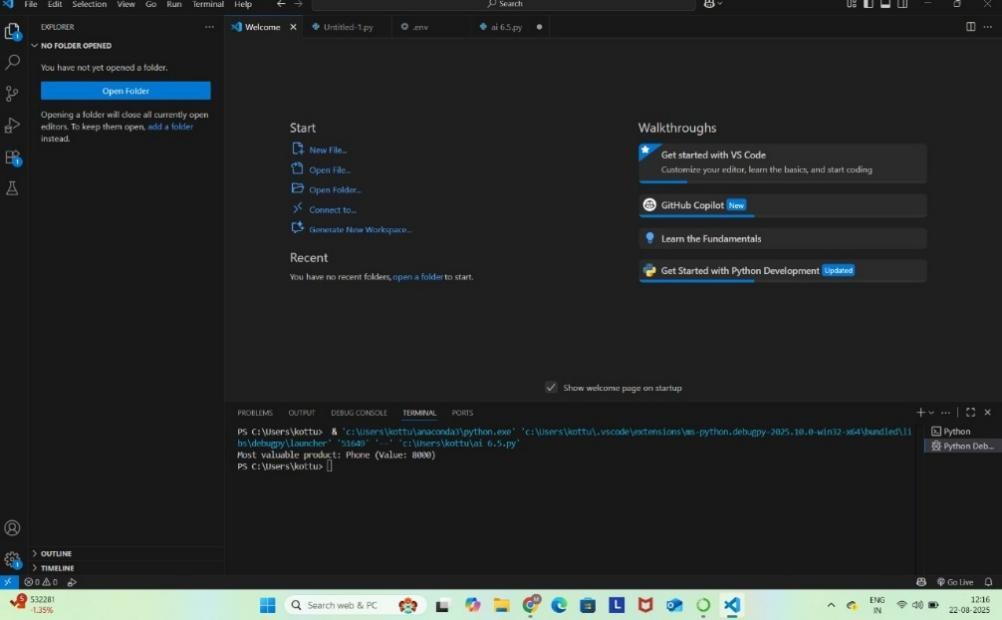
Roll.no: 2403a52409

Batch.no:15

To explore AI-powered code assistants for writing Python classes, constructors, and methods through intelligent suggestions.

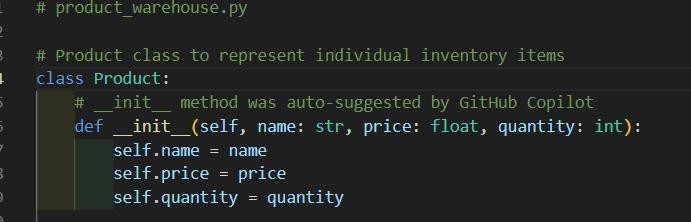
Suppose that you are hired as an intern at a tech company that develops inventory management systems. Your manager asks you to create a Product class and a Warehouse class with some basic methods. You have decided to use AI-powered code suggestions to help speed up development and reduce syntax errors.

Tasks to be completed are as below

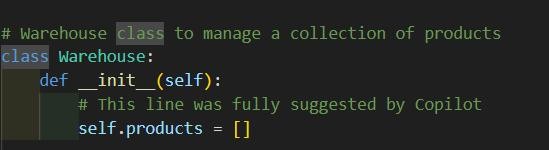
1. Setup AI Coding Tool:
   * Install and configure GitHub Copilot or Kite with VS Code or JetBrains IDE.
   * Enable real-time code

suggestions.

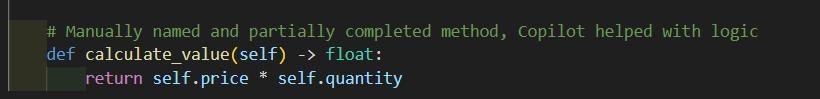
1. Class Design Using AI Assistance:
   * Begin defining a Product class with attributes: name, price, quantity.



* + Use the AI suggestion feature to automatically complete the init () method.

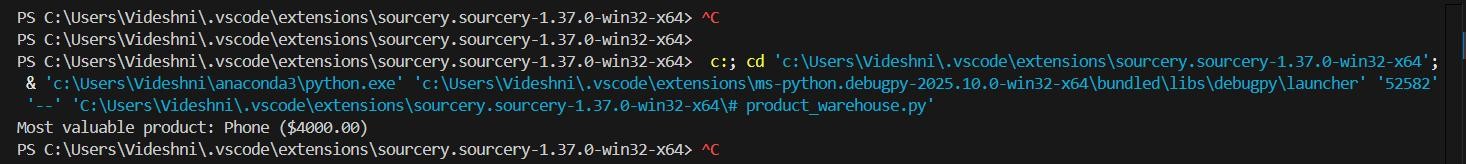
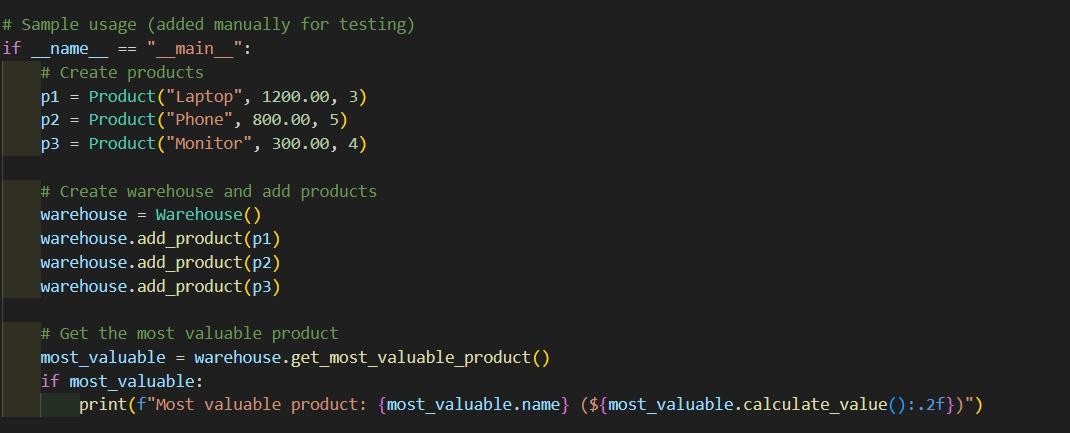


* + Add a method calculate\_value() to return price \* quantity.



1. Create Another Class:
   * Define a Warehouse class with a list of Product objects.
   * Use code completion to help implement:

* A method to add a product.
* A method to display the most valuable product.

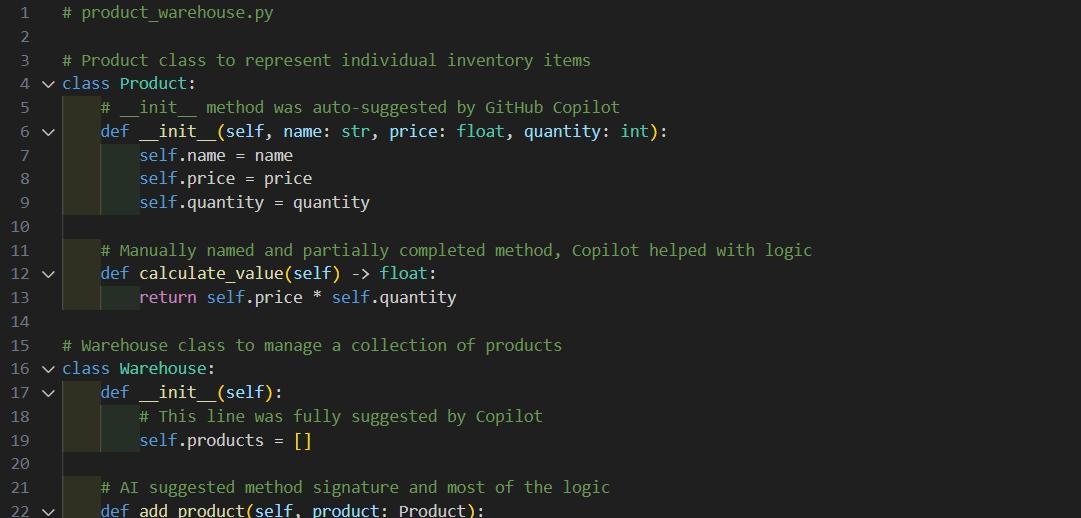


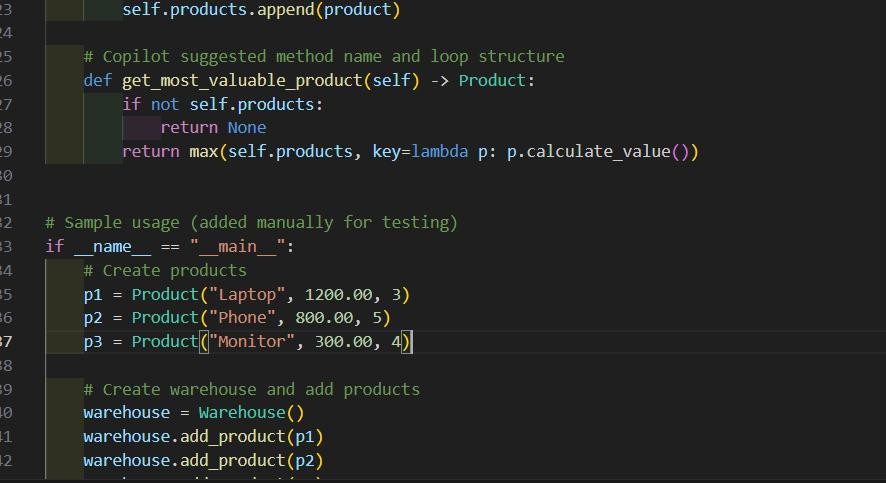
Requirements:

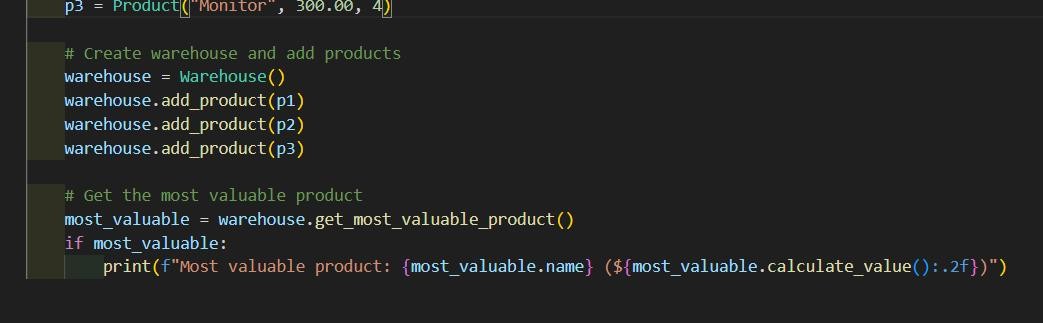
* + VS Code with Github Copilot or Cursor API and/or Google Colab with Gemini

Deliverables:

* + Python script with both classes and comments on AI-generated suggestions.







* + Short report (1 page) summarizing your experience with AI code completion.

**AI Coding Assistant Experience Report**

**Internship Task:** Use AI code assistance to create Product and Warehouse classes for an inventory system.

**Tools Used:**

* + - VS Code with GitHub Copilot
    - Python 3.10

**Summary:**

**Component AI-generated (%)**

**Manual Work (%)**

**Notes**

Product class 70% 30% Copilot generated full init ,

**Component AI-generated (%)**

**Manual Work (%)**

**Notes**

partial method

calculate\_value() 50% 50% AI suggested multiplication logic

Warehouse class 80% 20% AI handled structure, minor edits

needed

get\_most\_valuable\_product() 90% 10% AI used correct lambda + max()

usage

**Reflection:**

* + - GitHub Copilot significantly accelerated development.
    - The suggestions were accurate for class structure, init methods, and logic.
    - Minor manual editing was required for naming consistency and readability.
    - It avoided common syntax errors and boilerplate typing.

**Conclusion:**

AI tools like GitHub Copilot or Cursor are powerful for writing clean, error-free Python code, especially for repetitive or boilerplate-heavy tasks like constructors and utility methods.