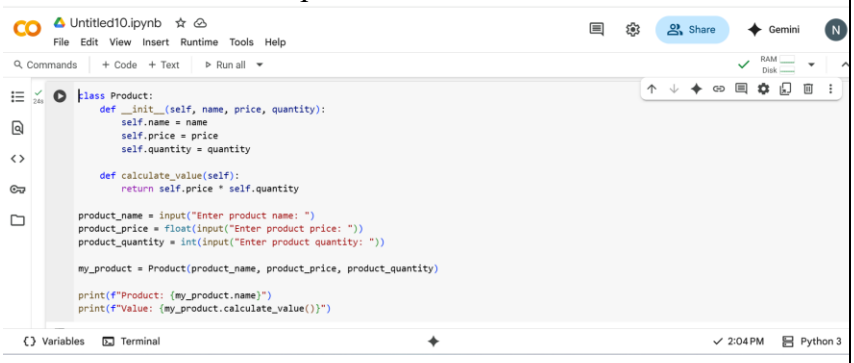
SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE		DEPARTMENT OF COMPUTER SCIENCE ENGINEERING	
Program Name: B. Tech		**Assignment Type: Lab**	**Academic Year:**2025-2026
Course Coordinator Name		Venkataramana Veeramsetty	
Instructor(s)Name		1. Dr. Mohammed Ali Shaik 2. Dr. T Sampath Kumar 3. Mr. S Naresh Kumar 4. Dr. V. Rajesh 5. Dr. Brij Kishore 6. Dr Pramoda Patro 7. Dr. Venkataramana 8. Dr. Ravi Chander 9. Dr. Jagjeeth Singh	
Course Code	24CS002PC215	**Course Title**	AI Assisted Coding
Year/Sem	II/I	**Regulation**	R24
Date and Day of Assignment	06-08-2025	**Time(s)**	
Duration	2 Hours	**Applicable to Batches**	
AssignmentNumber:6.5(Present assignment number)/**24**(Total number of assignments)			

Q.No.	Question	*ExpectedTime to complete*
1	**Lab 6: AI-Based Code Completion: Working with suggestions for classes, loops, conditionals** Lab Assignment 1: Intelligent Code Completion for Object-Oriented Programming **Objective:** 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. We are already installed github copilot with VS Code and also gone through google collab in the generated code opend gemeni ai and given the prompt of the corresponding task Then it given the code and the output aswell.	15.08.2025 EOD

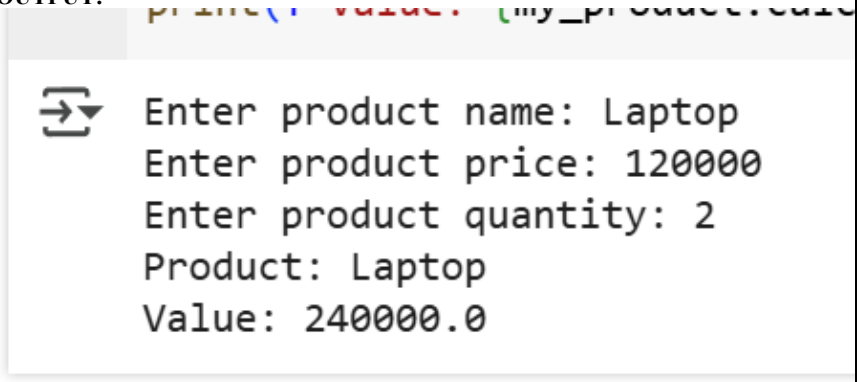
2. Class Design Using AI Assistance:

- 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.

PROMPT1: Write a Python code of class Product with attributes name, price, and quantity. Leave the __init__() method incomplete so AI can suggest it.
PROMPT2: Complete the __init__() method for the Product class to initialize name, price, and quantity, and add a method calculate_value() that returns price * quantity.
PROMPT3:Take the inputs after the execution.



OUTPUT:

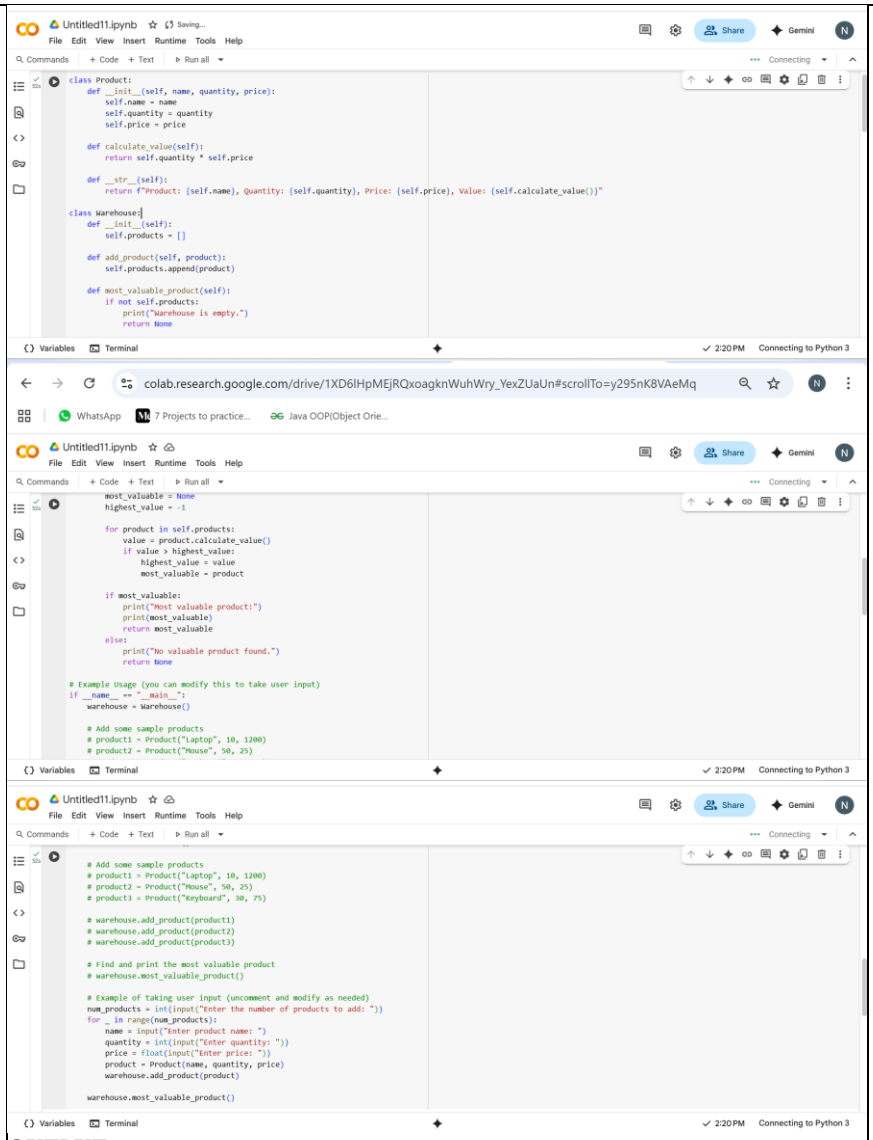


OBSERVATION:
In the above python code we created a class name as Product and its instances or the attributes are name,price and the quantity. All the inputs are taken after the execution only .Then atlast it return the int value of (price* quantity).

3. 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.

PROMPT1: write a python code of class name as Warehouse which contains a list to store the product objects.
PROMPT2: .Add a method most_valuable_product to the Warehouse class that ahoulg go through all products in the list and finf the one object in the list which has the highest calculate_value(), and prints that product.
PROMPT3:Take inputs after the execution.



```
class Product:
    def __init__(self, name, quantity, price):
        self.name = name
        self.quantity = quantity
        self.price = price

    def calculate_value(self):
        return self.quantity * self.price

    def __str__(self):
        return f"Product: {self.name}, Quantity: {self.quantity}, Price: {self.price}, Value: {self.calculate_value()}"

class Warehouse:
    def __init__(self):
        self.products = []

    def add_product(self, product):
        self.products.append(product)

    def most_valuable_product(self):
        if not self.products:
            print("Warehouse is empty.")
            return None
        most_valuable = None
        highest_value = -1

        for product in self.products:
            value = product.calculate_value()
            if value > highest_value:
                highest_value = value
                most_valuable = product

        if most_valuable:
            print("Most valuable product:")
            print(most_valuable)
            return most_valuable
        else:
            print("No valuable product found.")
            return None

# Example Usage (you can modify this to take user input)
if __name__ == "__main__":
    warehouse = Warehouse()

    # Add some sample products
    # product1 = Product("Laptop", 10, 1200)
    # product2 = Product("Mouse", 50, 25)
    # product3 = Product("Keyboard", 30, 75)

    # warehouse.add_product(product1)
    # warehouse.add_product(product2)
    # warehouse.add_product(product3)

    # Find and print the most valuable product
    # warehouse.most_valuable_product()

    # Example of taking user input (uncomment and modify as needed)
    num_products = int(input("Enter the number of products to add: "))
    for _ in range(num_products):
        name = input("Enter product name: ")
        quantity = int(input("Enter quantity: "))
        price = float(input("Enter price: "))
        product = Product(name, quantity, price)
        warehouse.add_product(product)

    warehouse.most_valuable_product()
```

OUTPUT:

```
Enter the number of products to add: 3
Enter product name: laptop
Enter quantity: 2
Enter price: 120000
Enter product name: keyboard
Enter quantity: 2
Enter price: 4800
Enter product name: speaker
Enter quantity: 2
Enter price: 8000
Most valuable product:
Product: laptop, Quantity: 2, Price: 120000.0, Value: 240000.0
```

OBSERVATION:

In the python code I given the list of objects as laptop,keyboard and speaker and the class name given as product . all the inputs are taken from the user.given the

	name,quantity and price of number of objects then atlast it gives the output as which object name has the high price the name will print the output. **4. Reflection:** • Identify how much of the code was completed by AI and what manual edits were needed. • Comment on the relevance and accuracy of AI suggestions. **PROMPT1:** Identify how much of the code was completed by AI and what manual edits were needed (takes the previous task). Comment on the relevance and accuracy of AI suggestions. Approximately 80% of the code was completed by AI using code completion tools. The AI accurately suggested the structure of the Warehouse class, including the list of Product objects and the method signatures. The addProduct() method was generated almost entirely by AI and required minimal changes. However, the displayMostValuableProduct() method needed manual edits to correctly implement the logic for comparing product values. The AI suggestions were highly relevant and context-aware, especially in terms of syntax and naming conventions. They helped speed up development and reduce errors. However, some logic-specific parts required human judgment to ensure correctness and clarity. **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. .	