ASSIGNMENT – 4

NAME: A. MANIDEEPIKA

HT.NO: 2403A52052

BATCH NO: AIB03

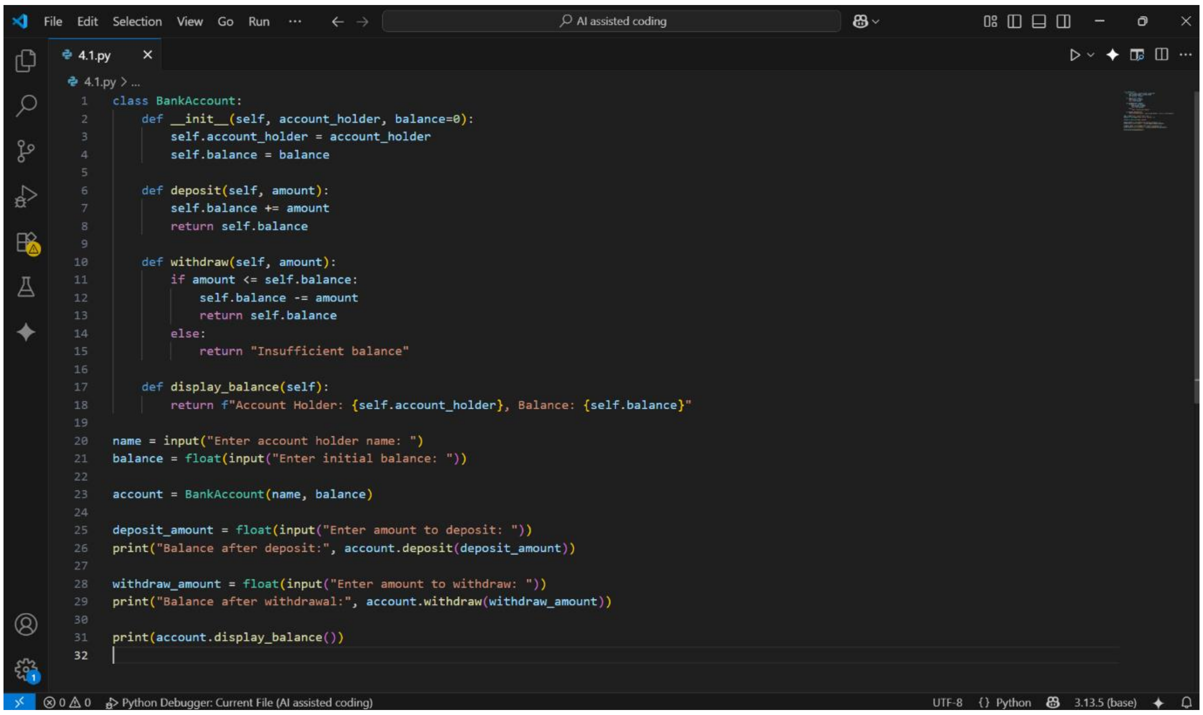
TASK 1:

Complete class with methods like:
deposit(self, amount)
withdraw(self, amount)
display_balance(self)

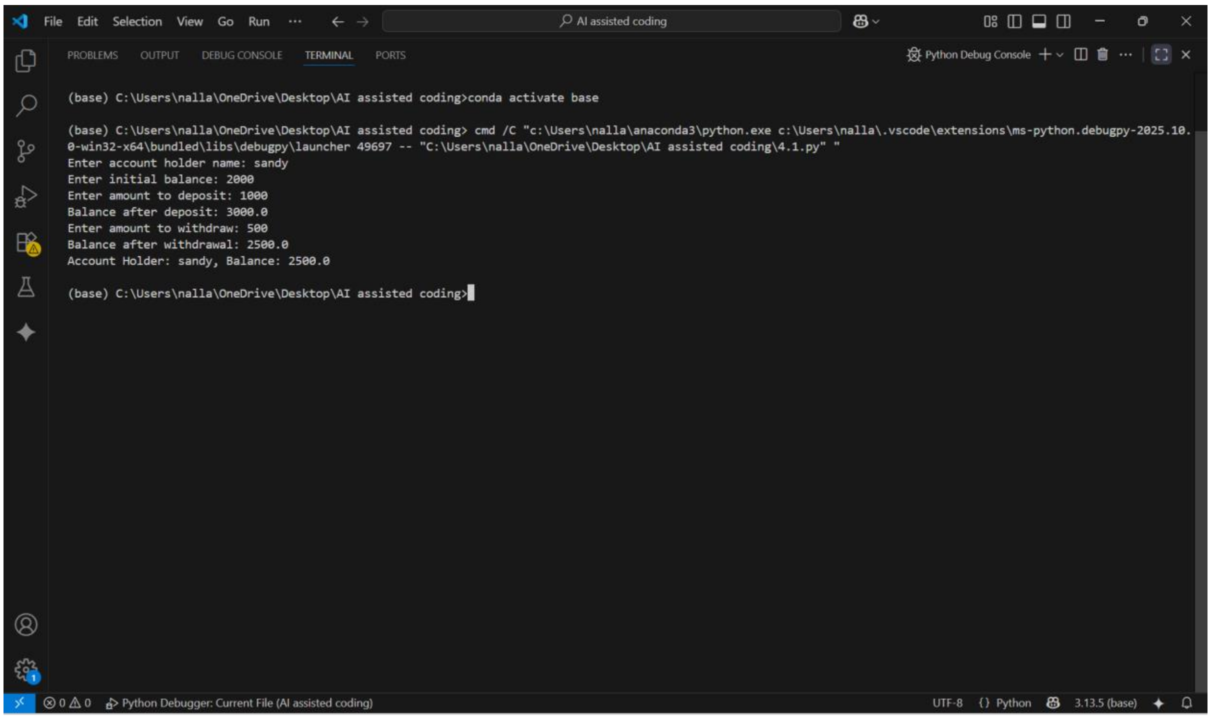
PROMPT:

Complete class with methods like:
deposit(self, amount)
withdraw(self, amount)
display_balance(self)

CODE:



```python
class BankAccount:
    def __init__(self, account_holder, balance=0):
        self.account_holder = account_holder
        self.balance = balance

    def deposit(self, amount):
        self.balance += amount
        return self.balance

    def withdraw(self, amount):
        if amount <= self.balance:
            self.balance -= amount
            return self.balance
        else:
            return "Insufficient balance"

    def display_balance(self):
        return f"Account Holder: {self.account_holder}, Balance: {self.balance}"

name = input("Enter account holder name: ")
balance = float(input("Enter initial balance: "))

account = BankAccount(name, balance)

deposit_amount = float(input("Enter amount to deposit: "))
print("Balance after deposit:", account.deposit(deposit_amount))

withdraw_amount = float(input("Enter amount to withdraw: "))
print("Balance after withdrawal:", account.withdraw(withdraw_amount))

print(account.display_balance())
```



```
(base) C:\Users\nalla\OneDrive\Desktop\AI assisted coding>conda activate base

(base) C:\Users\nalla\OneDrive\Desktop\AI assisted coding> cmd /C "c:\Users\nalla\anaconda3\python.exe c:\Users\nalla\.vscode\extensions\ms-python.debugpy-2025.10.
0-win32-x64\bundled\libs\debugpy\launcher 49697 -- "C:\Users\nalla\OneDrive\Desktop\AI assisted coding\4.1.py" "
Enter account holder name: sandy
Enter initial balance: 2000
Enter amount to deposit: 1000
Balance after deposit: 3000.0
Enter amount to withdraw: 500
Balance after withdrawal: 2500.0
Account Holder: sandy, Balance: 2500.0

(base) C:\Users\nalla\OneDrive\Desktop\AI assisted coding>
```

OBSERVATION:

This code defines a BankAccount class to manage a simple bank account with a holder's name and balance. It has methods to deposit money, withdraw money if enough balance is available, and display account details. The program asks the user for their name, initial balance, deposit, and withdrawal amounts, updates the account accordingly, and shows the final balance.

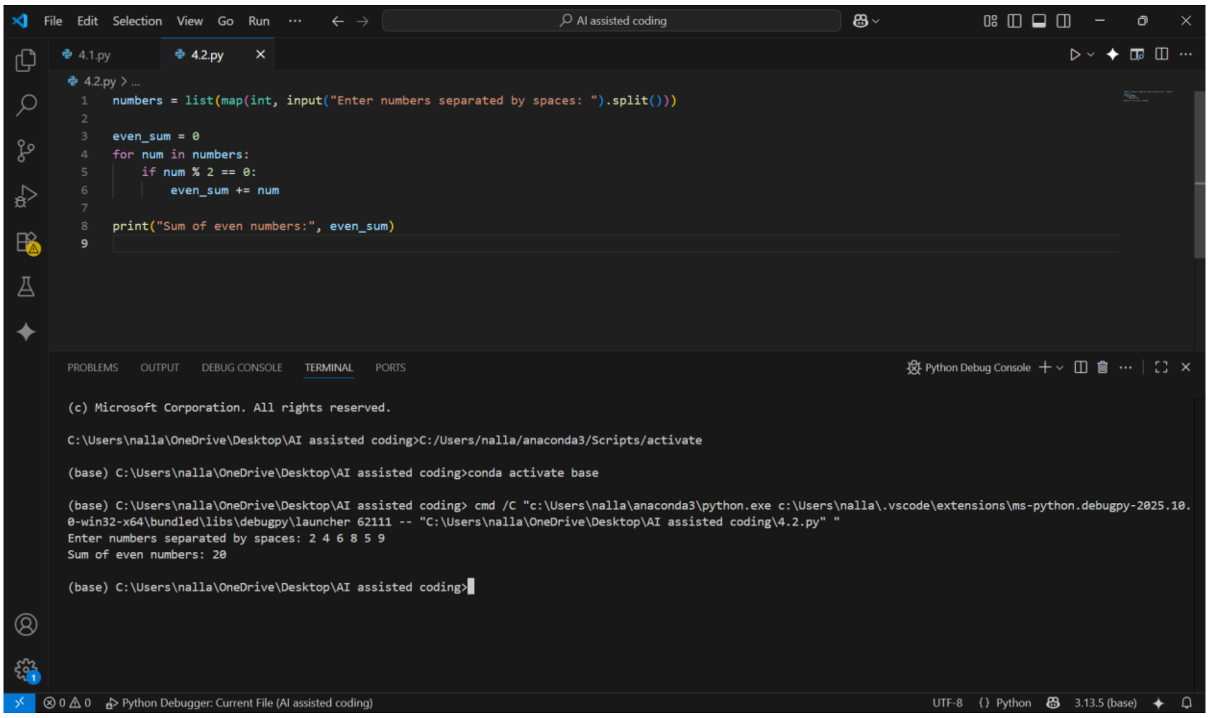
TASK 2:

Write a comment and the initial line of a loop to iterate over a list. Allow GitHub Copilot to complete the logic to sum all even numbers in the list.

PROMPT:

Write a comment and the initial line of a loop to iterate over a list. complete the logic to sum all even numbers in the list.

CODE:


File Edit Selection View Go Run ···
AI assisted coding
4.1.py
4.2.py
4.2.py > ...
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python Debug Console
⊗ 0 ⚠ 0 Python Debugger: Current File (AI assisted coding)
UTF-8 {} Python 3.13.5 (base)

```python
numbers = list(map(int, input("Enter numbers separated by spaces: ").split()))

even_sum = 0
for num in numbers:
    if num % 2 == 0:
        even_sum += num

print("Sum of even numbers:", even_sum)
```

```
(c) Microsoft Corporation. All rights reserved.

C:\Users\nalla\OneDrive\Desktop\AI assisted coding>C:/Users/nalla/anaconda3/Scripts/activate

(base) C:\Users\nalla\OneDrive\Desktop\AI assisted coding>conda activate base

(base) C:\Users\nalla\OneDrive\Desktop\AI assisted coding> cmd /C "c:\Users\nalla\anaconda3\python.exe c:\Users\nalla\.vscode\extensions\ms-python.debugpy-2025.10.
0-win32-x64\bundled\libs\debugpy\launcher 62111 -- "C:\Users\nalla\OneDrive\Desktop\AI assisted coding\4.2.py" "
Enter numbers separated by spaces: 2 4 6 8 5 9
Sum of even numbers: 20

(base) C:\Users\nalla\OneDrive\Desktop\AI assisted coding>
```

OBSERVATION:

This Python code calculates the sum of even numbers from a list entered by the user. It first prompts the user to input numbers separated by spaces, then converts that input into a list of integers using map and split. It initializes a variable even_sum to zero, then iterates through each number in the list. If a number is divisible by 2 (i.e., it’s even), it adds that number to even_sum. Finally, it prints the total sum of all even numbers entered by the user.

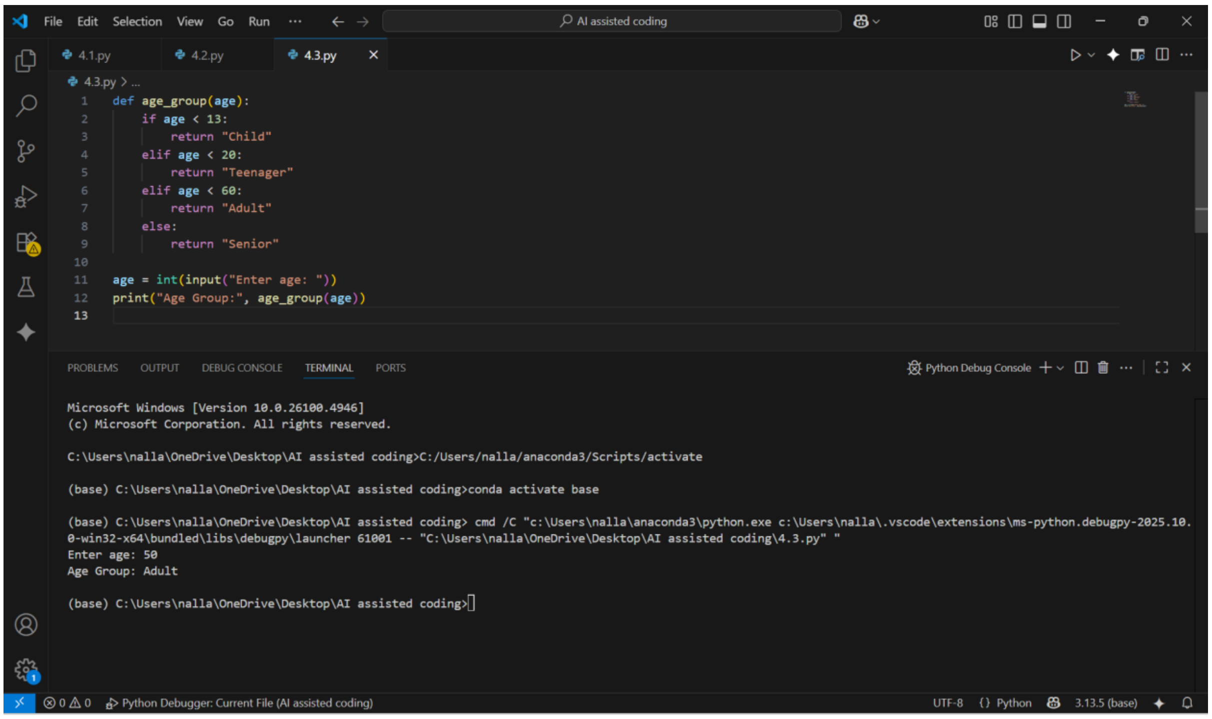
TASK 3:

Start a function that takes age as input and returns whether the person is a child, teenager, adult, or senior using if-elif-else. Use Copilot to complete the conditionals.

PROMPT:

Start a function that takes age as input and returns whether the person is a child, teenager, adult, or senior using if-elif-else**.**

CODE:



File Edit Selection View Go Run
AI assisted coding
4.1.py
4.2.py
4.3.py
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python Debug Console
Python Debugger: Current File (AI assisted coding)
UTF-8 {} Python 3.13.5 (base)

```python
def age_group(age):
    if age < 13:
        return "Child"
    elif age < 20:
        return "Teenager"
    elif age < 60:
        return "Adult"
    else:
        return "Senior"

age = int(input("Enter age: "))
print("Age Group:", age_group(age))
```

```
Microsoft Windows [Version 10.0.26100.4946]
(c) Microsoft Corporation. All rights reserved.

C:\Users\nalla\OneDrive\Desktop\AI assisted coding>C:/Users/nalla/anaconda3/Scripts/activate

(base) C:\Users\nalla\OneDrive\Desktop\AI assisted coding>conda activate base

(base) C:\Users\nalla\OneDrive\Desktop\AI assisted coding> cmd /C "c:\Users\nalla\anaconda3\python.exe c:\Users\nalla\.vscode\extensions\ms-python.debugpy-2025.10.
0-win32-x64\bundled\libs\debugpy\launcher 61001 -- "C:\Users\nalla\OneDrive\Desktop\AI assisted coding\4.3.py" "
Enter age: 50
Age Group: Adult

(base) C:\Users\nalla\OneDrive\Desktop\AI assisted coding>
```

OBSERVATION:

This Python code defines a function age_group that categorizes a person based on their age. It takes an integer age as input and checks several conditions: if the age is less than 13, it returns "Child"; if it is between 13 and 19, it returns "Teenager"; if it is between 20 and 59, it returns "Adult"; and if it is 60 or above, it returns "Senior". The program prompts the user to enter their age, converts it to an integer, passes it to the age_group function, and then prints the resulting age category.

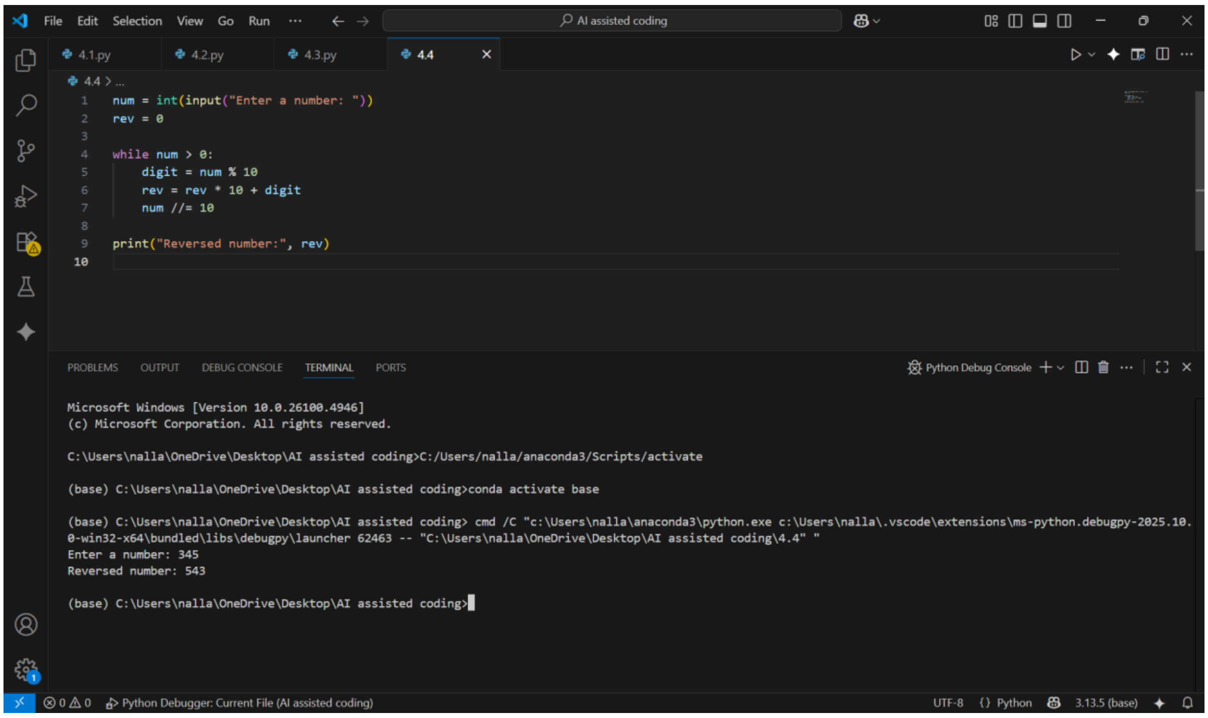
TASK 4:

Write a comment and start a while loop to reverse the digits of a number. Let Copilot complete
the loop logic

PROMPT:

Write a comment and start a while loop to reverse the digits of a number.

CODE:



```python
num = int(input("Enter a number: "))
rev = 0

while num > 0:
    digit = num % 10
    rev = rev * 10 + digit
    num //= 10

print("Reversed number:", rev)
```

```
Microsoft Windows [Version 10.0.26100.4946]
(c) Microsoft Corporation. All rights reserved.

C:\Users\nalla\OneDrive\Desktop\AI assisted coding>C:/Users/nalla/anaconda3/Scripts/activate

(base) C:\Users\nalla\OneDrive\Desktop\AI assisted coding>conda activate base

(base) C:\Users\nalla\OneDrive\Desktop\AI assisted coding> cmd /C "c:\Users\nalla\anaconda3\python.exe c:\Users\nalla\.vscode\extensions\ms-python.debugpy-2025.10.
0-win32-x64\bundled\libs\debugpy\launcher 62463 -- "C:\Users\nalla\OneDrive\Desktop\AI assisted coding\4.4" "
Enter a number: 345
Reversed number: 543

(base) C:\Users\nalla\OneDrive\Desktop\AI assisted coding>
```

OBSERVATION:

This Python code reverses the digits of a number entered by the user. It first prompts the user to input a number and stores it in num, while initializing rev to 0 to hold the reversed number. Inside a while loop that runs as long as num is greater than 0, it extracts the last digit using num % 10, adds it to rev after shifting the existing digits to the left (rev * 10 + digit), and removes the last digit from num using integer division num //= 10. After the loop finishes, it prints the reversed number stored in rev.

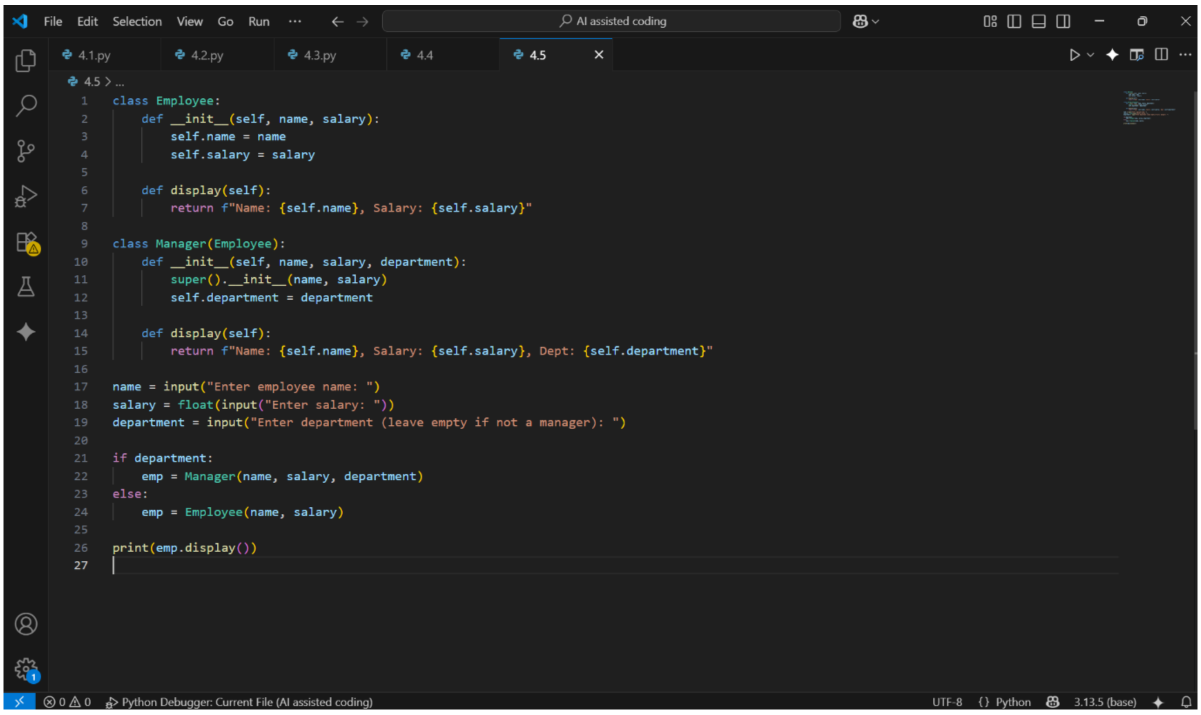
TASK 5:

**Begin a class Employee with attributes name and salary. Then, start a derived class Manager that inherits from Employee and adds a department. Let GitHub Copilot complete the methods
and constructor chaining**

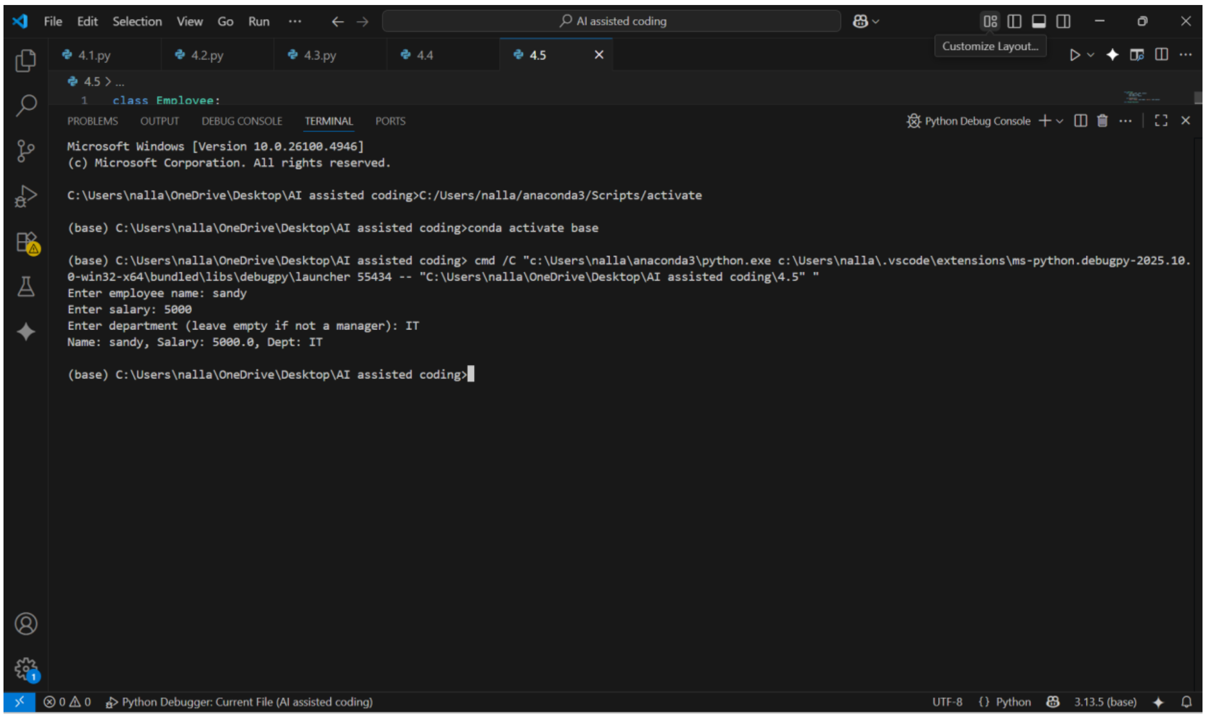
PROMPT:

Begin a class Employee with attributes name and salary. Then, start a derived class Manager that inherits from Employee and adds a department

CODE:



```python
class Employee:
    def __init__(self, name, salary):
        self.name = name
        self.salary = salary

    def display(self):
        return f"Name: {self.name}, Salary: {self.salary}"

class Manager(Employee):
    def __init__(self, name, salary, department):
        super().__init__(name, salary)
        self.department = department

    def display(self):
        return f"Name: {self.name}, Salary: {self.salary}, Dept: {self.department}"

name = input("Enter employee name: ")
salary = float(input("Enter salary: "))
department = input("Enter department (leave empty if not a manager): ")

if department:
    emp = Manager(name, salary, department)
else:
    emp = Employee(name, salary)

print(emp.display())
```


File Edit Selection View Go Run
AI assisted coding
4.1.py
4.2.py
4.3.py
4.4
4.5
Customize Layout...
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python Debug Console
Python Debugger: Current File (AI assisted coding)
UTF-8 {} Python 3.13.5 (base)

```
class Employee:
```

```
Microsoft Windows [Version 10.0.26100.4946]
(c) Microsoft Corporation. All rights reserved.

C:\Users\nalla\OneDrive\Desktop\AI assisted coding>C:/Users/nalla/anaconda3/Scripts/activate

(base) C:\Users\nalla\OneDrive\Desktop\AI assisted coding>conda activate base

(base) C:\Users\nalla\OneDrive\Desktop\AI assisted coding> cmd /C "c:\Users\nalla\anaconda3\python.exe c:\Users\nalla\.vscode\extensions\ms-python.debugpy-2025.10.
0-win32-x64\bundled\libs\debugpy\launcher 55434 -- "C:\Users\nalla\OneDrive\Desktop\AI assisted coding\4.5" "
Enter employee name: sandy
Enter salary: 5000
Enter department (leave empty if not a manager): IT
Name: sandy, Salary: 5000.0, Dept: IT

(base) C:\Users\nalla\OneDrive\Desktop\AI assisted coding>
```

OBSERVATION:

This Python code demonstrates **inheritance** in classes. It defines an Employee class with a constructor that stores the employee’s name and salary, and a display method that returns these details as a string. Then it defines a Manager class that inherits from Employee and adds a department attribute. The Manager class also overrides the display method to include the department in the output. The program asks the user to enter a name, salary, and optionally a department. If a department is provided, it creates a Manager object; otherwise, it creates a regular Employee object. Finally, it prints the details of the created object using the display method.