**AI ASSISTED CODING LAB**

**ASSIGNMENT 4.4**

**ENROLLMENT NO :**2503A51L25

**BATCH NO:** 19

**NAME:** SRIRAMOJU AJAY

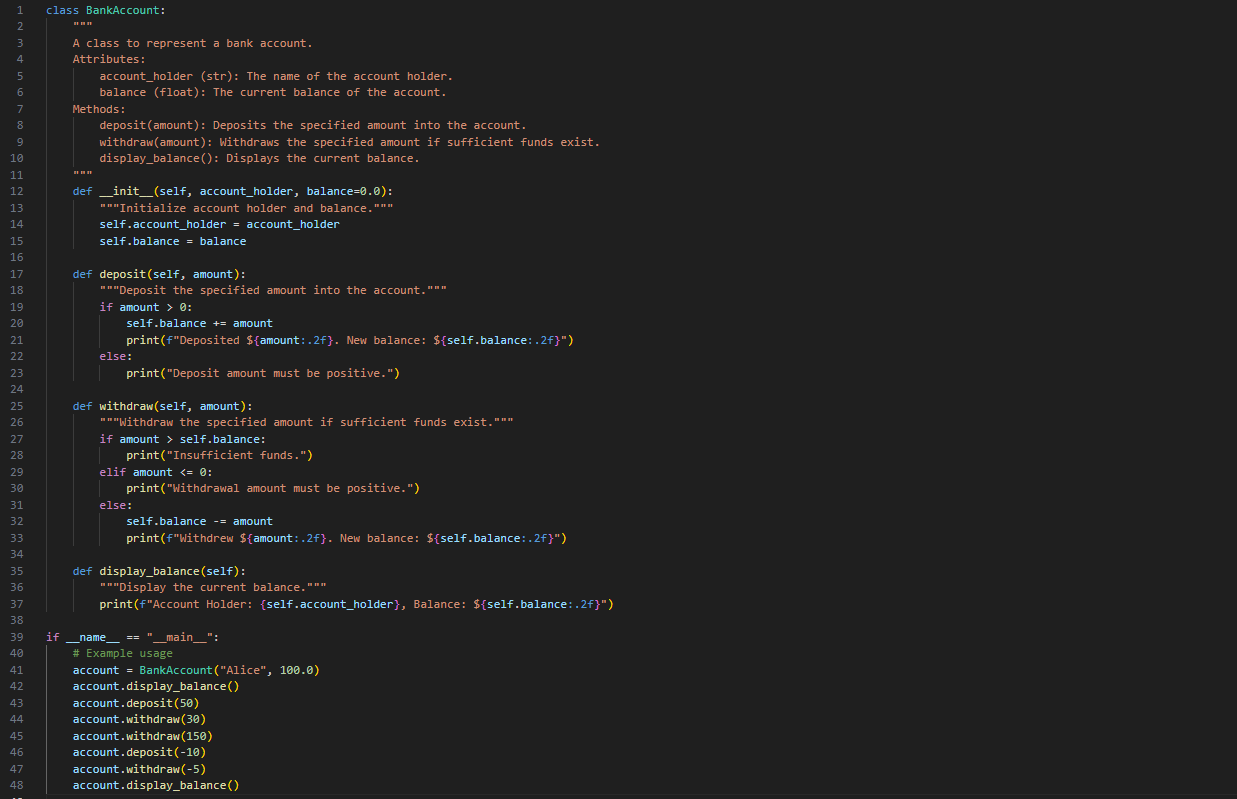
**TASK1**

**TASK1 DESCRIPTION** :- Auto-Complete a Python Class for Bank Account

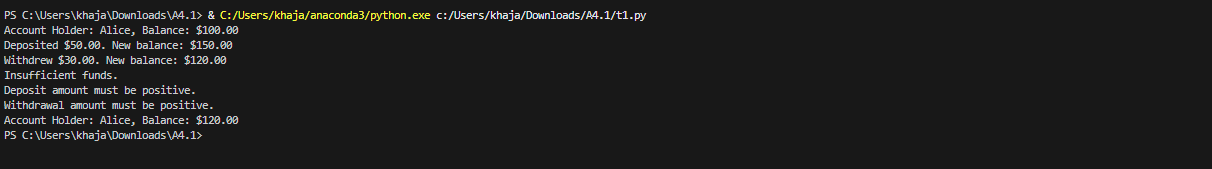
• Write a class definition comment and start the constructor for a class called Bank Account with account\_holder and balance attributes. Use GitHub Copilot to auto-complete the rest of the class, including methods to deposit, withdraw, and display balance.

**PROMPT**:-Write a Python class called BankAccount with a class definition comment, a constructor that takes account\_holder and balance, and methods deposit(self, amount), withdraw(self, amount) with error handling for insufficient funds, and display\_balance(self); provide the complete implementation with example.

CODE:-



OUTPUT:-



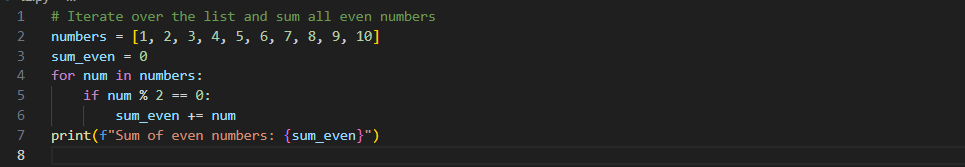
**TASK2**

**TASK2 DESCRIPTION**:- Auto-Complete a For Loop to Sum Even Numbers in a List

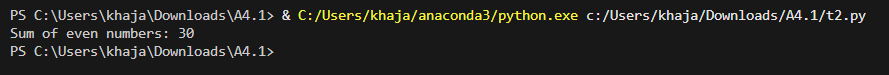
Write a comment and the initial line of a loop to iterate over a list. Allow GitHub Copilot to complete the logic to sum all even numbers in the list.

**PROMPT**:-Write a comment and the initial line of a Python loop to iterate over a list, then let GitHub Copilot auto-complete the logic to sum all even numbers in the list and implement with example.

CODE:-



OUTPUT:-



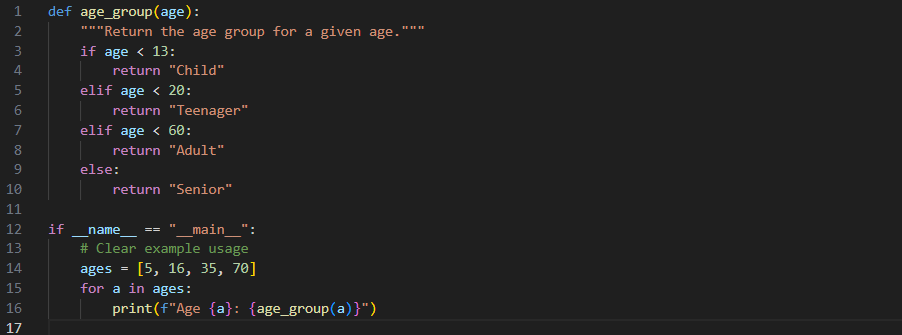
**TASK3**

**TASK3 DESCRIPTION:- Auto-Complete Conditional Logic to Check Age Group**

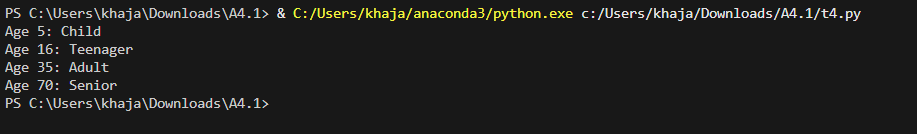
**Start a function that takes age as input and returns whether the person is a child, teenager,adult, or senior using if-elif-else. Use Copilot to complete the conditionals**

**PROMPT:-Generate a python function that takes age as input and returns whether the person is a child, teenager,adult, or senior using if-elif-else.implement with clear example.**

**CODE:-**

****

**OUTPUT:-**

****

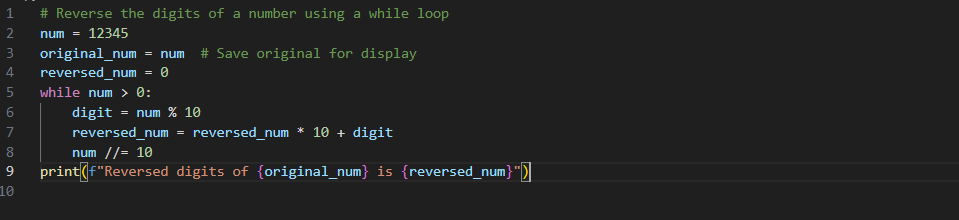
**TASK4**

**TASK4 DESCRIPTION:-** **Auto-Complete a While Loop to Reverse Digits of a Number**

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

**PROMPT:-Generate a Python comment and start a while loop to reverse the digits of a number. Then use GitHub Copilot to auto-complete the loop logic and implement it with example**

**CODE:-**

****

OUTPUT:-



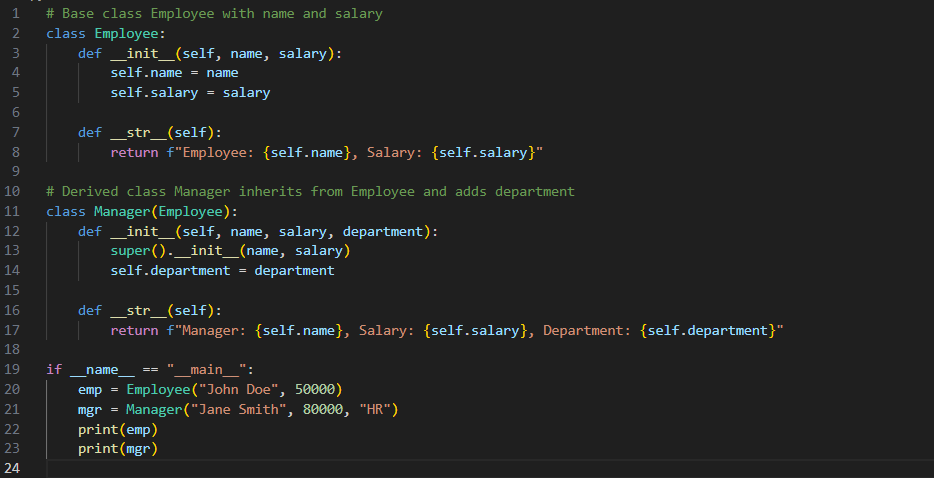
**TASK5**

**TASK5 DESCRIPTION:-** **Auto-Complete Class with Inheritance (Employee → Manager)**

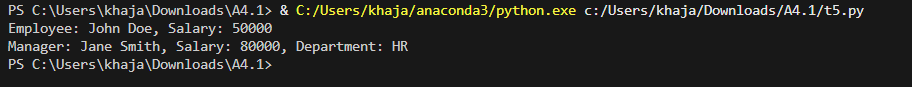
**Begin a class Employee with attributes name and salary. Then, start a derived class Manager that inherits from Employee and adds a department. Let GitHub Copilot complete the methods and constructor chaining**

**PROMPT:-** **Write the start of a Python class Employee with attributes name and salary, then begin a derived class Manager that inherits from Employee and adds a department attribute. Let GitHub Copilot auto-complete the constructors and any necessary methods, including proper constructor chaining implement with example.**

**CODE:-**

****

**OUTPUT:-**

**\**

**OBSERVATION:-** I observed how GitHub Copilot can be effectively used to auto-complete Python code when given the correct prompts. By providing only the initial structure such as a class definition, function header, or loop starter, Copilot was able to generate complete implementations with logical flow.

* In **Task 1**, I noticed that Copilot could generate a full Python class with constructor, methods, and proper error handling just from a descriptive prompt.
* In **Task 2**, it correctly completed the loop logic to filter and sum even numbers, showing its ability to understand conditional iteration.
* In **Task 3**, the function for age classification highlighted how Copilot handles nested conditionals and returns meaningful results.
* In **Task 4**, the while loop to reverse digits showed Copilot’s capability to handle mathematical logic inside loops.
* In **Task 5**, I observed how Copilot implements object-oriented concepts like inheritance and constructor chaining effectively.