AI-ASSISTED CODING

ASSIGNMENT-6.4

ROLL NO:2403A52084

TASK: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.

PROMPT:

# Create a Python class named Student with attributes: name, roll\_number, and marks.

# Add a method display\_details() to print all student information.

# Add a method is\_passed() that checks if marks are above average (e.g., 40) and prints "Passed" or "Failed".

class Student.

CODE:

A screen shot of a computer program

AI-generated content may be incorrect.

OUT PUT:

A computer screen shot of numbers

AI-generated content may be incorrect.

OBSERVATOIN:

The error occurred because the constructor in your Student class was incorrectly named \_init\_ instead of \_\_init\_\_. Python requires double underscores for special methods like constructors. After correcting it to \_\_init\_\_, the code works as expected, allowing you to input marks separated by spaces and displaying the student's details and result.

TASK2:

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.

PROMPT:

Write a code to print the square of the even number only.

CODE:A computer screen with colorful text

AI-generated content may be incorrect.

OUTPUT:

A screenshot of a computer program

AI-generated content may be incorrect.

OBSERVATION:

The code prompts the user to enter numbers separated by spaces, converts them to integers, and iterates through each number. For every even number, it prints the square of that number. Odd numbers are ignored and not printed. There is a minor typo in the print statement: it repeats "is" twice.

TASK3:

Create a class called BankAccount with attributes account\_holder and balance. Use Copilot to  
complete methods for deposit(), withdraw(), and check for insufficient balance.

PROMPT:

Create a Python class called BankAccount with attributes account\_holder and balance. Implement methods deposit(amount), withdraw(amount), and get\_balance(). The withdraw method should check for insufficient funds and prevent overdrawing. Include print statements to show transaction results and current balance.

CODE:

A screen shot of a computer program

AI-generated content may be incorrect.

OUTPUT:

A black screen with colorful text

AI-generated content may be incorrect.

OBSERVATION:

The code defines a BankAccount class with methods to deposit and withdraw money, check for insufficient funds, and display the current balance. It shows transaction results using print statements. An example demonstrates creating an account and performing transactions.

TASK4:

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.

PROMPT:

Define a list of student dictionaries with keys 'name' and 'score'. Write a complete while loop to print the names of students who scored more than 75, with proper condition checks and formatted output.

CODE:

A computer screen shot of a program code

AI-generated content may be incorrect.

OUTPUT:A black screen with yellow and blue text

AI-generated content may be incorrect.

OBSERVATOIN:

The code defines a list of student dictionaries with names and scores. It uses a while loop to check each student's score, and prints the name of students who scored more than 75. The loop continues until all students are checked.

TASK5:

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.

PROMPT:

write a python code to generate methods to add\_item, remove\_item, and use a loop to calculate the total bill using conditional discounts.

CODE:

A screen shot of a computer program

AI-generated content may be incorrect.

OUTPUT:

A black screen with white text

AI-generated content may be incorrect.

OBERVATOIN:

The code defines a ShoppingCart class with methods to add and remove items, and calculate the total bill. If the total exceeds 500, a 10% discount is applied. Example usage demonstrates adding and removing items, then calculating the final bill.