# ASSIGNMENT 6.4

**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.  
Expected Outcome #1:  
• Completed class with Copilot-generated methods like display\_details() and is\_passed(),  
demonstrating use of if-else conditions.

**Prompt**:

• Start a Python class named Student with attributes name, roll\_number, and marks. Prompt  
GitHub Copilot to complete methods for displaying details and checking ifmarksareaboveaveragExpectedOutcome#1:• Completed class with Copilot-generated methods like display\_details() and is\_passed(),  
demonstrating use of if-else conditions.

CODE AND OUTPUT:

A screen shot of a computer program

Description automatically generated

OBSERVATION: The code was generated in Copilot and executed in VS Code. It is related to student marks and passed out.

**Task 2**:  
• 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.  
Expected Outcome #2:  
• A complete loop generated by Copilot with conditional logic (if number % 2 == 0) and  
appropriate output.

**PROMPT:**

• Write the first two lines of a for loop to iterate through a list of numbers. Use a commentprompt to let Copilotsuggesthow to calculate and print the square of even numbers only.Expected Outcome #2:• A complete loop generated by Copilot with conditional logic (if number % 2 == 0) andappropriate output.

**CODE AND OUTPUT:**

**A screenshot of a computer

Description automatically generated**

**A screen shot of a computer

Description automatically generated**

**OBSERVATION:** The code was generated in Copilot and executed in VS Code. It produced improved code with the correct output.

**Task 3:**  
• Create a class called BankAccount with attributes account\_holder and balance. Use Copilot to  
complete methods for deposit(), withdraw(), and check for insufficient balance.  
Expected Outcome #3:  
• Functional class with complete method definitions using if conditions and self attributes. Code  
should prevent overdrawing.

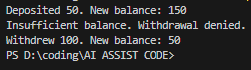
**PROMPT**:

• Create a class called BankAccount with attributes account\_holder and balance. Use Copilot to  
complete methods for deposit(), withdraw(), and check for insufficient balance.Expected Outcome #3:  
• Functional class with complete method definitions using if conditions and self attributes. Code should prevent overdrawing.

**CODE AND OUTPUT:**

**A computer screen shot of a code

Description automatically generated**

****

**OBSERVATION:** The BankAccount class allows depositing and withdrawing money. The methods update and display the account balance after each transaction.

**Task 4:**  
• 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.  
Expected Outcome #4:  
• A complete while loop generated by Copilot with proper condition checks and formatted  
output.

**PROMPT:**

• 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. Expected Outcome #4: • A complete while loop generated by Copilot with proper condition checks and formatted output.

**CODE AND OUTPUT:**

A computer screen shot of text

Description automatically generated

A black screen with white text

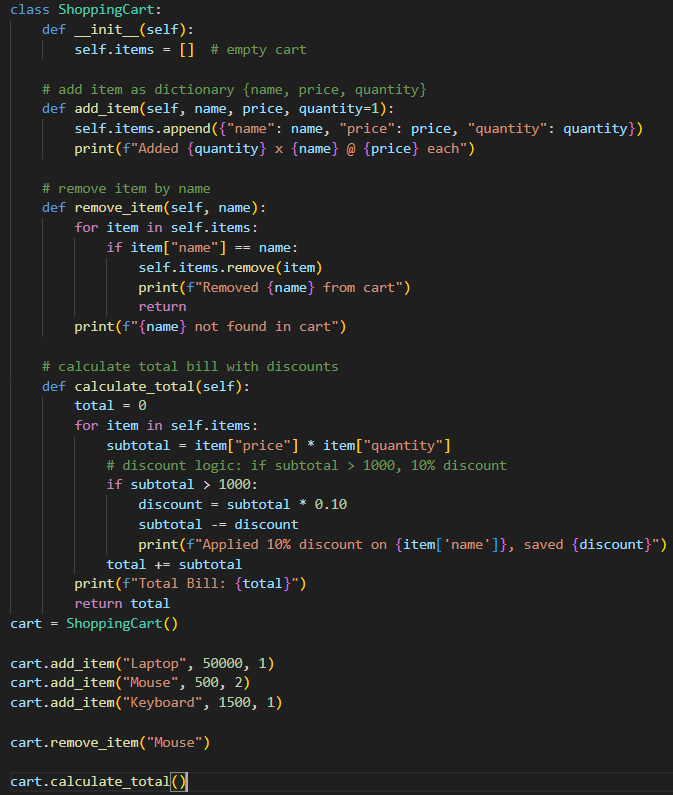
Description automatically generated

OBSERVATION: In this code we can observe that the code is checking that the students has scored more than 75 or not

**Task 5:**  
• 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.  
Expected Outcome #5:  
• A fully implemented ShoppingCart class with Copilot-generated loops and if-else statements  
handling item management and discount logic

Prompt: • 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.  
Expected Outcome #5:  
• A fully implemented ShoppingCart class with Copilot-generated loops and if-else statements  
handling item management and discount logic also print the Output

**Code and Output:**



A computer screen with white text

Description automatically generated

Observation:In this code we can see the code about the Shopping bill code consists of adding item, remove item, discount , total bill.