ASSIGNMENT 6

HTNO:2403A51319

BATCH:15

**Task Description #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.

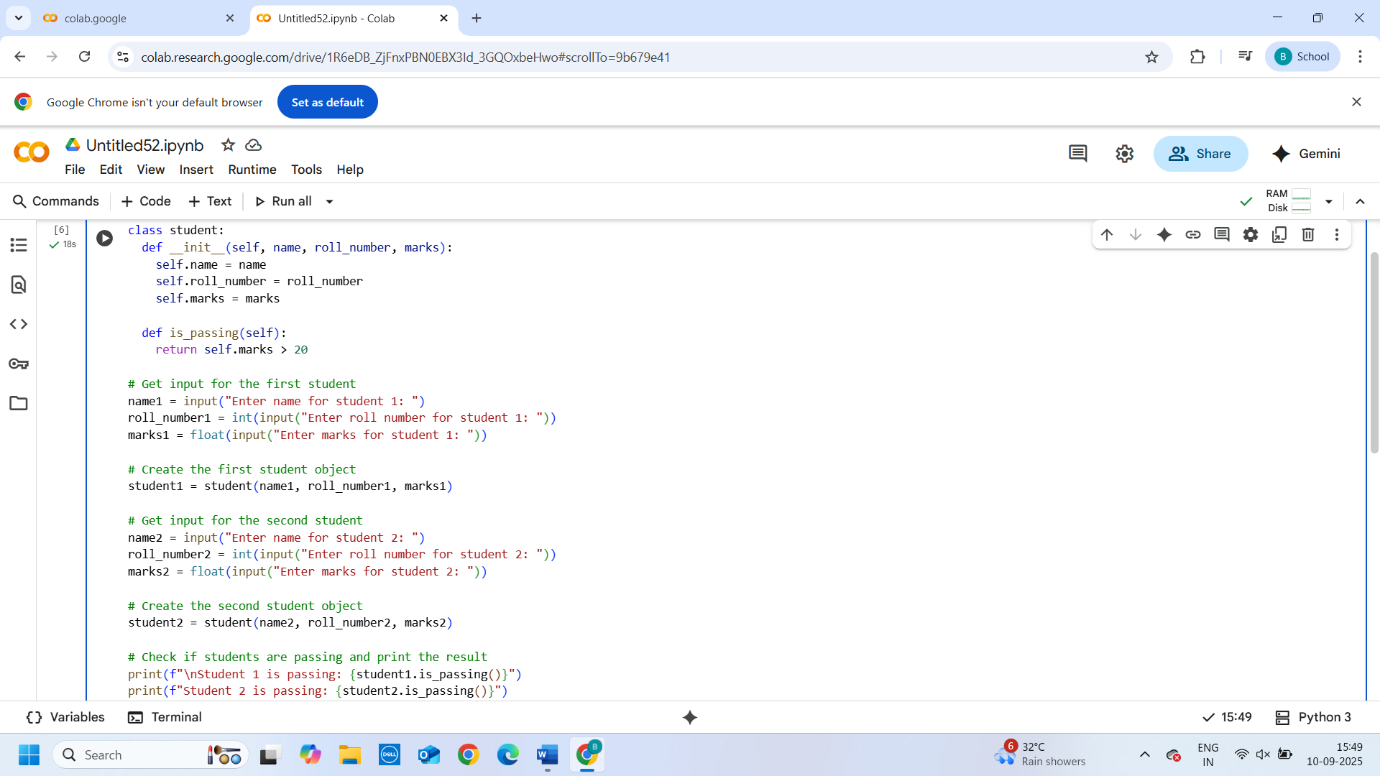
PROMPT:

write a python program to create a class name as student with attributes name,roll\_number,marks

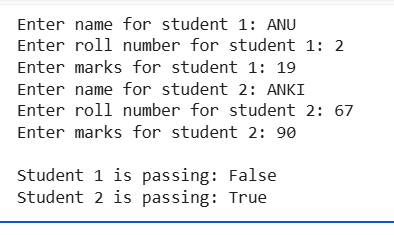
modify the code with user input with two student

check whether the student is pass or fail if average is greater than 20 pass else fail in boolean

CODE:



OUTPUT:



EXPLANATION:

This code defines a class called student with attributes for name, roll\_number, and marks. It also includes a method is\_passing() which returns True if the student's marks are greater than 20, and False otherwise.

The code then prompts the user to enter the name, roll number, and marks for two students, creates instances of the student class with this information, and finally prints whether each student is passing or failing based on the is\_passing() method.

**Task Description #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.

PROMPT:

write a python program to iterate a list of numbers through for loop and calculate,print square of even numbers

CODE:

A white background with numbers and symbols

AI-generated content may be incorrect.

OUTPUT:

A screenshot of a math test

AI-generated content may be incorrect.

EXPLANATION:

* numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]: This line creates a list named numbers containing integers from 1 to 10.
* for number in numbers:: This is a for loop that iterates through each element in the numbers list. In each iteration, the current element is assigned to the variable number.
* if number % 2 == 0:: This is an if statement that checks if the current number is even. The modulo operator (%) returns the remainder of a division. If number % 2 is 0, it means the number is perfectly divisible by 2 and is therefore even.
* square = number \*\* 2: If the if condition is true (the number is even), this line calculates the square of the number using the exponentiation operator (\*\*) and assigns the result to the variable square.
* print(f"The square of {number} is {square}"): This line prints the result to the console. The f before the string indicates a formatted string literal, which allows you to embed the values of variables directly within the string using curly braces {}.
* **Task Description #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.

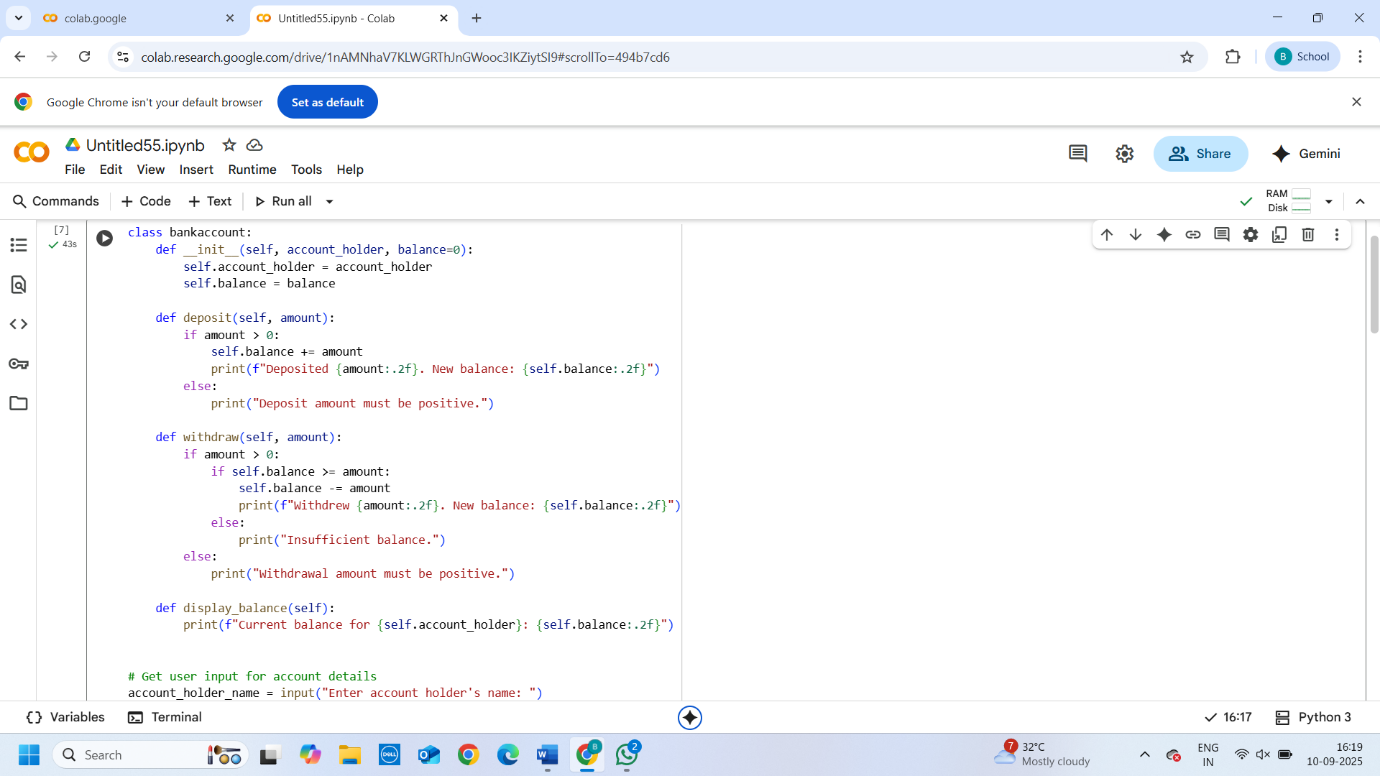
PROMPT:

write a python program to create a class called bankaccount with attributes account\_holder and balance

modify the code with user input

with methods for deposit(),withdraw(), and check for insufficient balance using if conditions and self attribute code should prevent overdrawing

CODE:



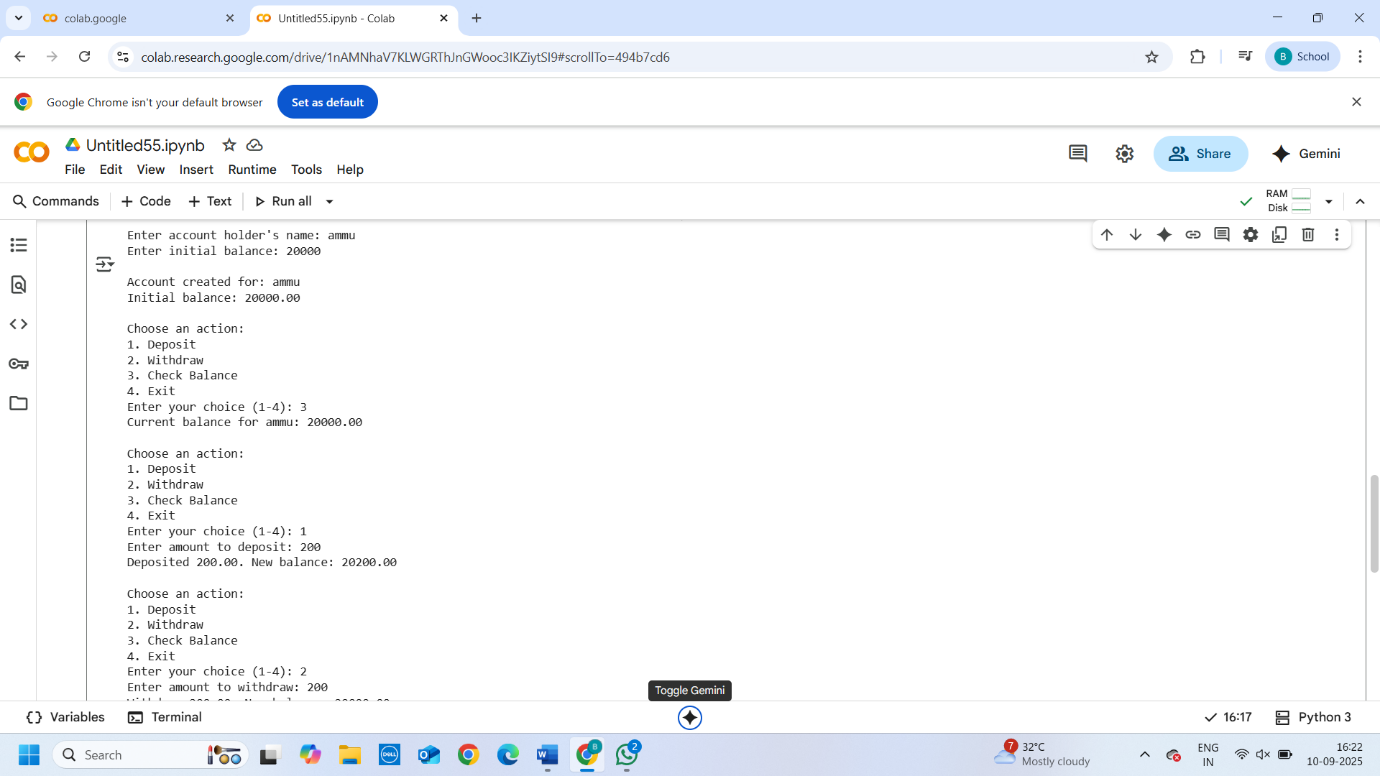
A screenshot of a computer

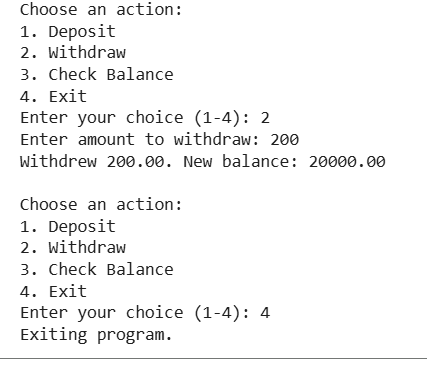
AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

OUTPUT:



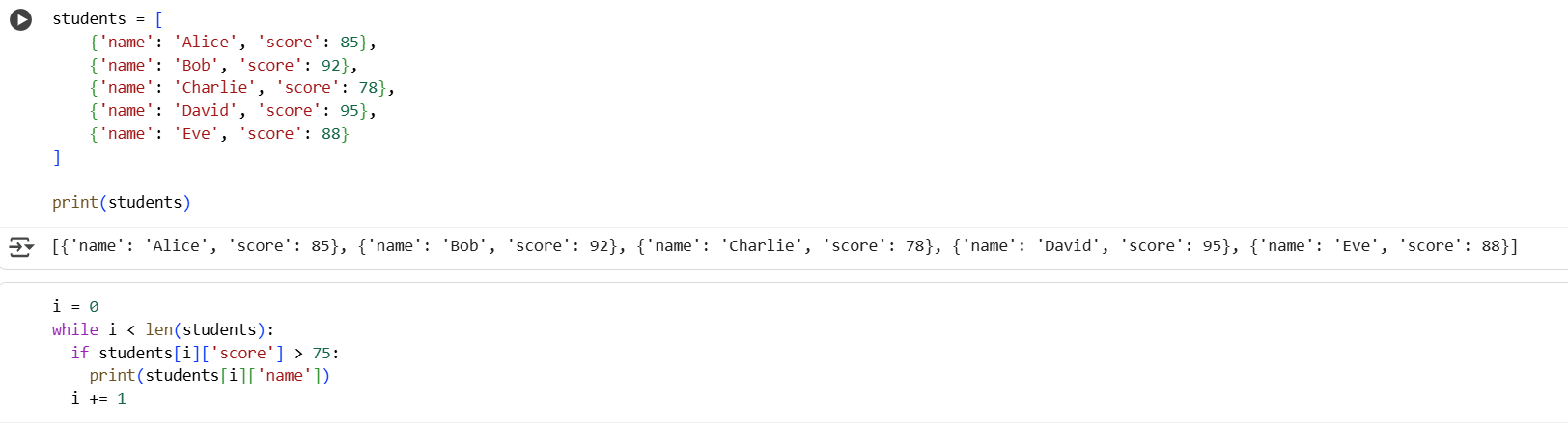


EXPLANATION:

**1. bankaccount Class:**

* **\_\_init\_\_(self, account\_holder, balance=0):** This is the constructor of the class. It's called when you create a new bankaccount object. It initializes two attributes:
  + account\_holder: Stores the name of the account holder.
  + balance: Stores the current balance in the account, defaulting to 0 if no initial balance is provided.
* **deposit(self, amount):** This method allows you to deposit money into the account.
  + It checks if the amount is positive.
  + If positive, it adds the amount to the self.balance.
  + It then prints a confirmation message with the deposited amount and the new balance.
* **withdraw(self, amount):** This method allows you to withdraw money from the account.
  + It checks if the amount is positive.
  + It then checks if the self.balance is greater than or equal to the amount to prevent overdrawing.
  + If there's enough balance, it subtracts the amount from the self.balance and prints a confirmation message.
  + If there's insufficient balance, it prints an "Insufficient balance" message.
* **display\_balance(self):** This method simply prints the current balance for the account holder.

**2. Interactive Program:**

* The code first prompts the user to enter the account holder's name and initial balance.
* It then creates an instance (an object) of the bankaccount class named user\_account using the provided input.
* It displays the details of the newly created account.
* The while True: loop creates an interactive menu:
  + It presents options for depositing, withdrawing, checking the balance, or exiting.
  + It takes the user's choice as input.
  + Based on the choice, it calls the corresponding method (deposit, withdraw, or display\_balance) of the user\_account object.
  + It includes error handling (try-except) to catch invalid input when the user is expected to enter a number for deposit or withdrawal amounts.
  + If the user chooses option 4, the loop breaks, and the program exits.
  + If the user enters an invalid choice, it prompts them to enter a number between 1 and 4.
* **Task Description #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.
  + PROMPT:
  + write a pytthon program list of a student dictionaries with key name and score.
* write while loop to print the names of students who scored more than 75.
  + CODE:
  + 

OUTPUT:

A group of people with their names

AI-generated content may be incorrect.

EXPLANATION:

1. It initializes a counter i to 0.
2. The while i < len(students): condition checks if i is less than the total number of students in the list. This ensures the loop continues as long as there are students to process.
3. Inside the loop, students[i] accesses the dictionary of the current student.
4. students[i]['score'] > 75 checks if the current student's score is greater than 75.
5. If the condition is true (the score is greater than 75), print(students[i]['name']) prints the name of that student.
6. Finally, i += 1 increments the counter i so that the loop moves to the next student in the list

**Task Description #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.

**PROMPT:**

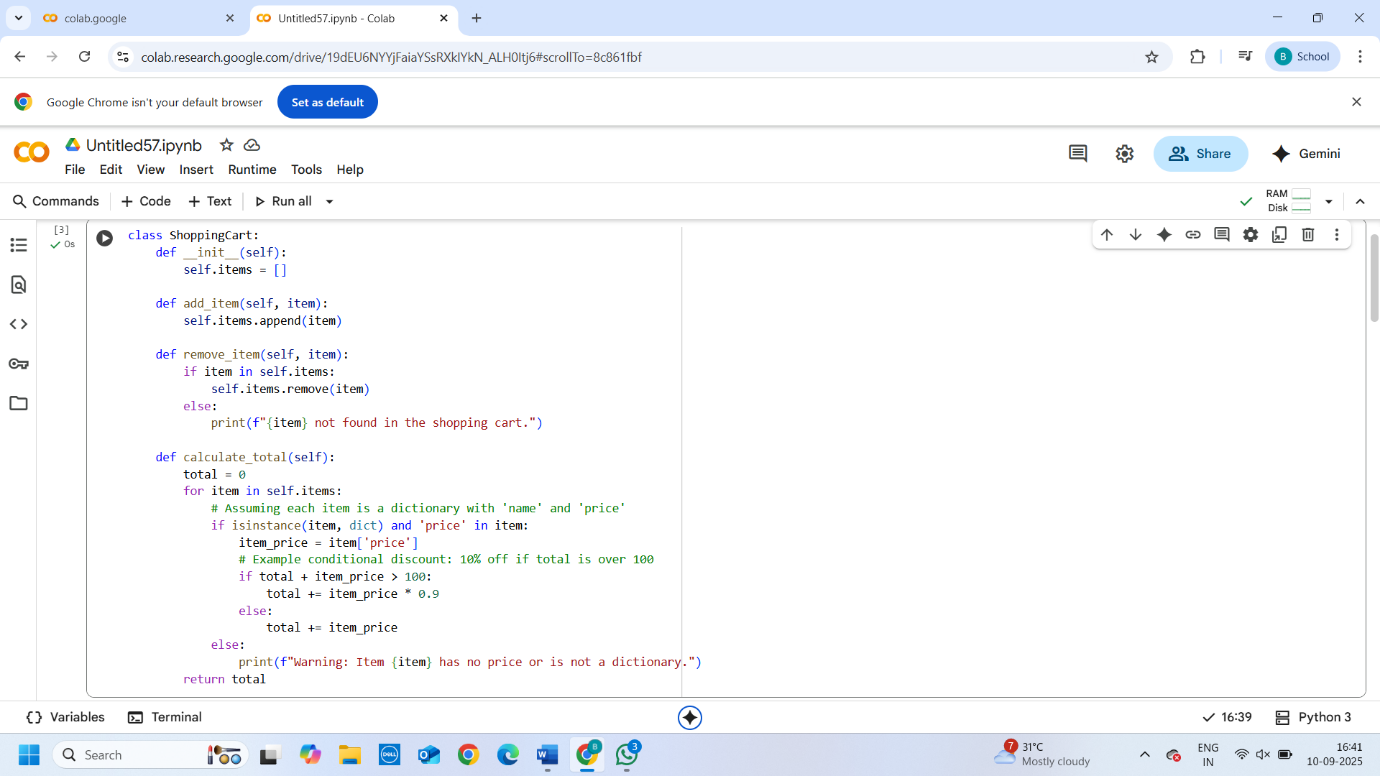
write a python program with class shoppingcart with an empty item list

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

generated loops and if-else statements handling item management and discount logic.

modify code with user input

CODE:



A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

OUTPUT:

A screenshot of a computer

AI-generated content may be incorrect.

EXPLANATION:

1. **cart = ShoppingCart()**: This line creates a new instance of your ShoppingCart class, essentially creating an empty shopping cart.
2. **while True:**: This starts an infinite loop, allowing the user to continuously interact with the shopping cart until they choose to exit.
3. **Printing Options**: Inside the loop, the code prints a menu of options to the user: add an item, remove an item, calculate the total, or exit.
4. **choice = input(...)**: This line prompts the user to enter their choice and stores it in the choice variable.
5. **if choice == '1': ... (Add Item)**: If the user enters '1', the code prompts for the item name and price. It includes error handling using a try-except block to ensure the price is a valid number. If valid, it creates a dictionary for the item and adds it to the cart.items list using the add\_item method.
6. **elif choice == '2': ... (Remove Item)**: If the user enters '2', the code prompts for the item name to remove. It then iterates through the items in the cart to find a matching item (based on name) and removes the first match found using the remove\_item method.
7. **elif choice == '3': ... (Calculate Total)**: If the user enters '3', the code calls the cart.calculate\_total() method to get the total bill (including any discounts) and prints the formatted total.
8. **elif choice == '4': ... (Exit)**: If the user enters '4', the code prints an exit message and the break statement exits the while loop, ending the program.
9. **else: ... (Invalid Choice)**: If the user enters anything other than '1', '2', '3', or '4', it prints an "Invalid choice" message.