## NAME:DHEVULAPALLY RASHMITHA

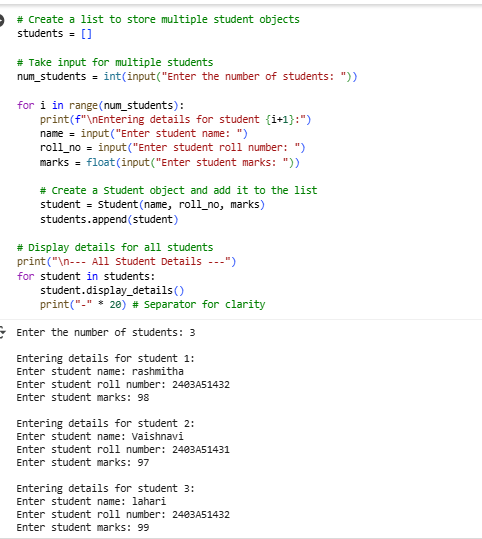
## ENROLLMENT\_NO:2403A51432

## BRANCH:CSE

## Task 1:

Prompt: write a program in python to create class name student with attributes names like name,roll\_no,marks by taking input from the user.complete methods for displaying details.

Screenshot:



Explanation:

Create a list: We make an empty list called students to hold all the student information.

Ask how many: We ask you how many students you want to enter.

Loop and collect: We repeat the process of asking for name, roll number, and marks for each student you specified.

Make student objects: For each set of details, we create a Student object using the blueprint we made earlier.

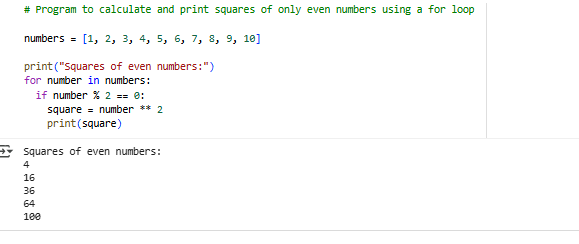
Add to list: We put each new Student object into the students list.

Show all: Finally, we go through the students list and use the display\_details() method for each student object to print their information.

## TASK 2:

Prompt: write a python program to iterate list of numbers using basic for loop give the logic code and modify the previous code for calculating and printing squares of only even numbers.

Screenshot:



Explanation:

numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]: This line creates a list of numbers from 1 to 10 and stores it in a variable called numbers.

print("Squares of even numbers:"): This line simply prints the text "Squares of even numbers:" to the output, so you know what the following output represents.

for number in numbers:: This is a loop that goes through each item in the numbers list one by one. In each turn of the loop, the current item is stored in the variable number.

if number % 2 == 0:: This is a condition that checks if the current number is even. The modulo operator (%) gives you the remainder when one number is divided by another. If a number divided by 2 has a remainder of 0, it means the number is even.

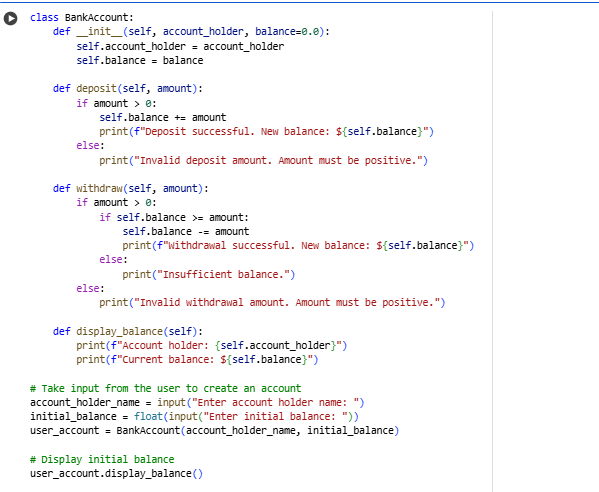
square = number \*\* 2: If the if condition is true (the number is even), this line calculates the square of the number (multiplies it by itself) and stores the result in a variable called square.

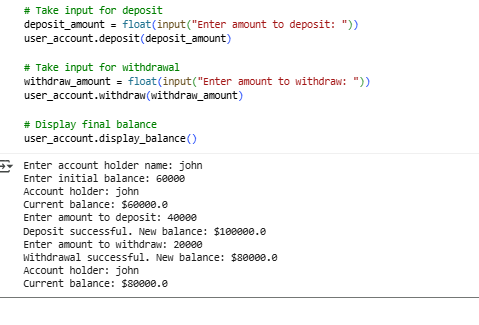
print(f"The square of {number} is {square}"): If the number is even, this line prints a message showing the original even number and its calculated square. The f"" is a way to easily include the values of variables (number and square) directly within the text.

## TASK 3:

Prompt: write a python program to Create a BankAccount class with attributes account\_holder and balance. Implement methods for deposit(amount), withdraw(amount), and handle insufficient balance cases."and give me as a user input.

Screenshot:





Explanation:

.BankAccount Blueprint: We create a blueprint called BankAccount that holds the account owner's name and their money (balance).

Deposit: A method to add money to the balance. It checks if the amount is positive.

Withdraw: A method to take money from the balance. It checks if the amount is positive and if there's enough money.

Display Balance: A method to show the account owner's name and current balance.

User Interaction: The code then asks you for the account holder's name and initial balance to create an account.

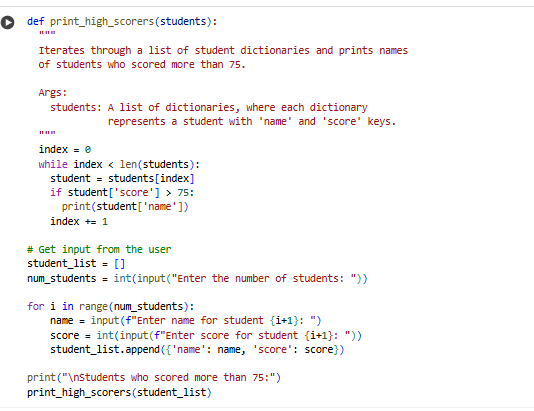
Transactions: It then asks you for amounts to deposit and withdraw, performing those actions on the account.

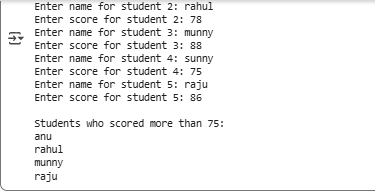
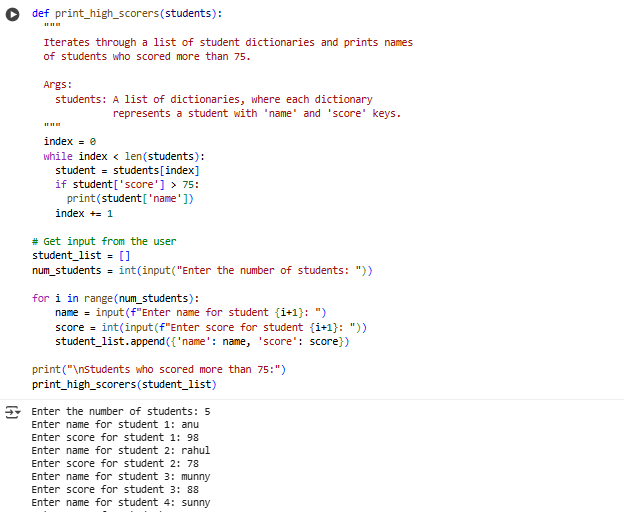
Show Final Balance: Finally, it displays the balance after the transactions.

## TASK 4:

Prompt: :write a python program and give me input as a user. A list of student dictionaries with keys 'name' and 'score', write a while loop to iterate through the list and print the names of students who scored more than 75.

Screenshot:





Explanation:

Get Student Count: Ask the user how many students they want to enter.

Create Empty List: Initialize an empty list to store student data.

Input Loop: Loop for the number of students specified, asking for each student's name and score and storing them as a dictionary in the list.

Check and Print: Use a while loop to go through the list. For each student, check if their score is greater than 75 and print their name if it is.

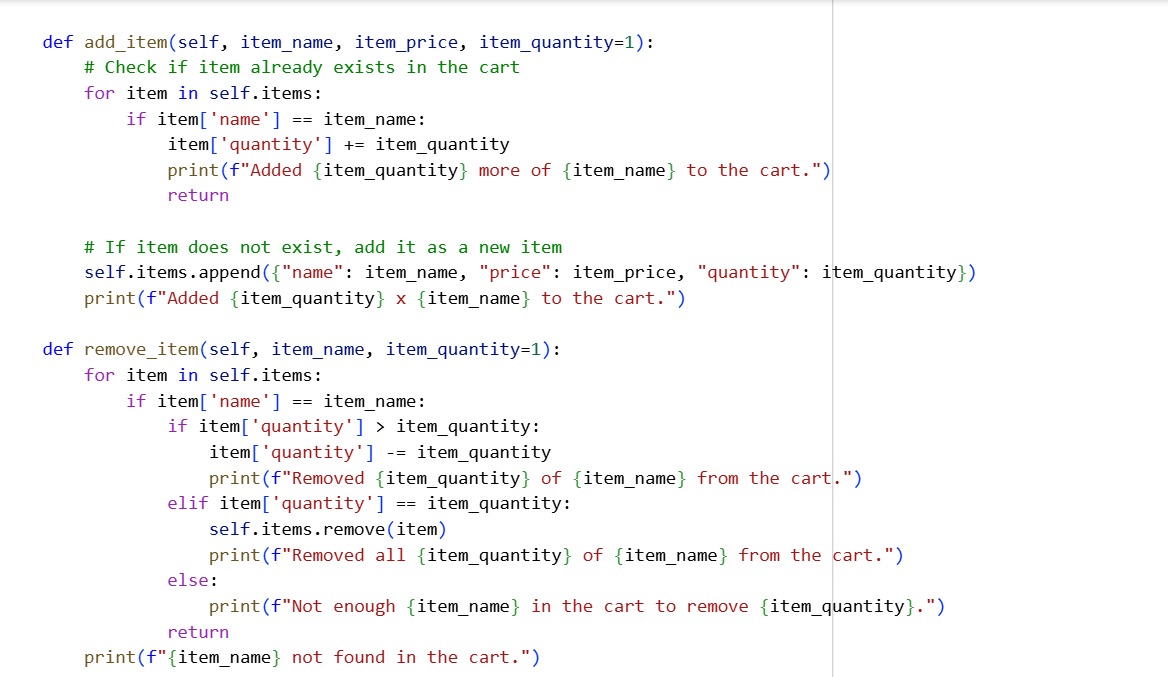
## TASK 5:

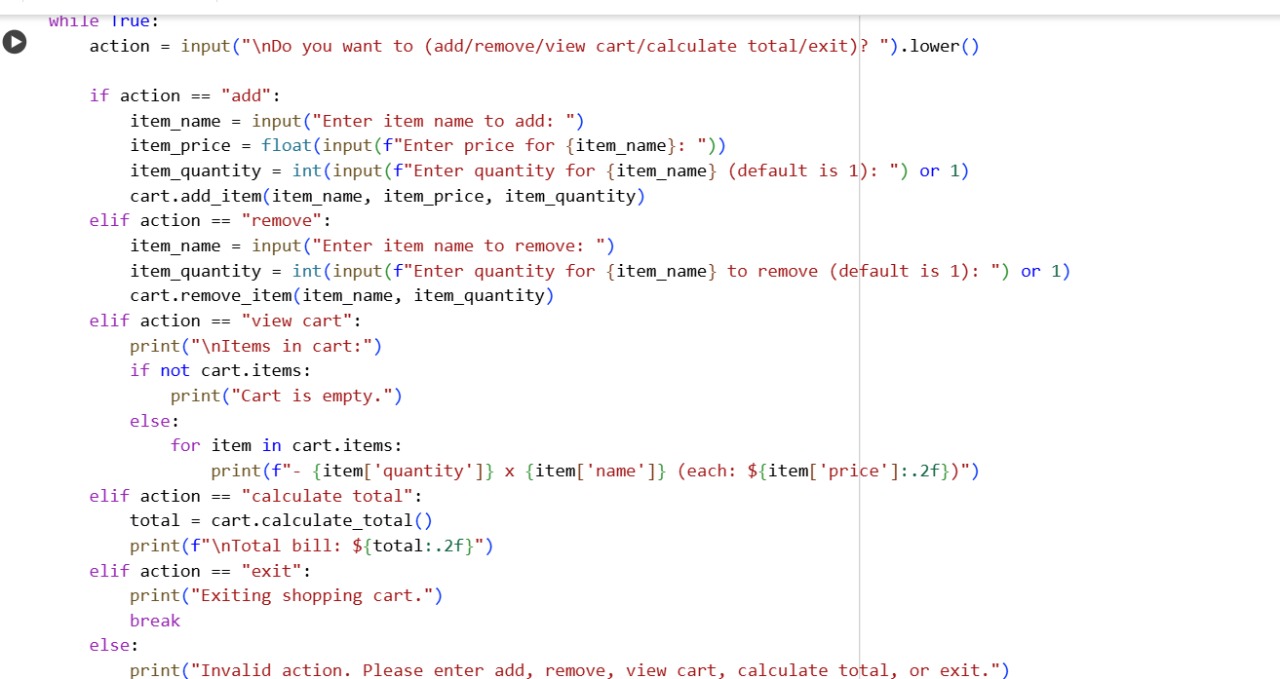
Prompt: write a program in python to create a class named ShoppingCart with an empty items list.

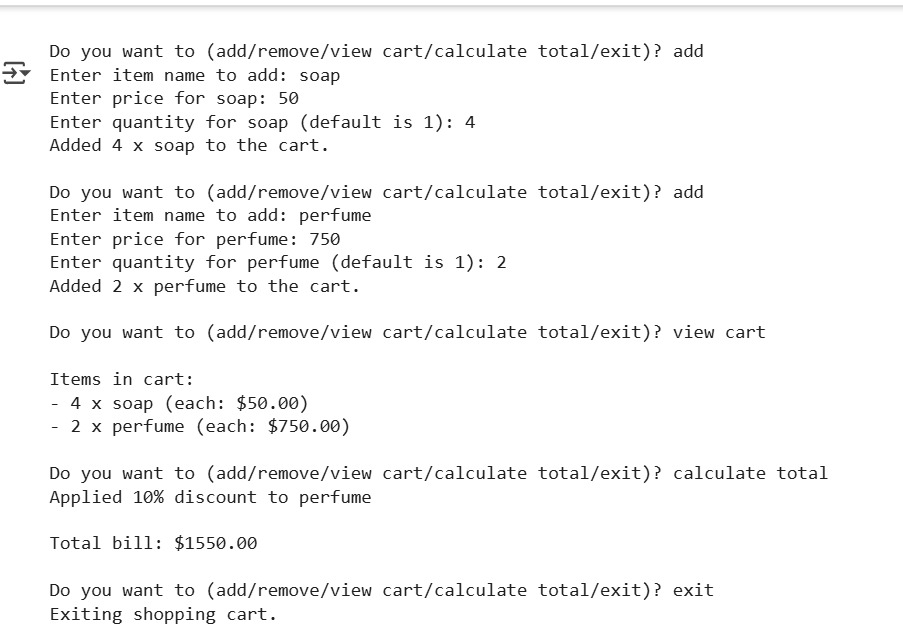
AI.6.4.51.png

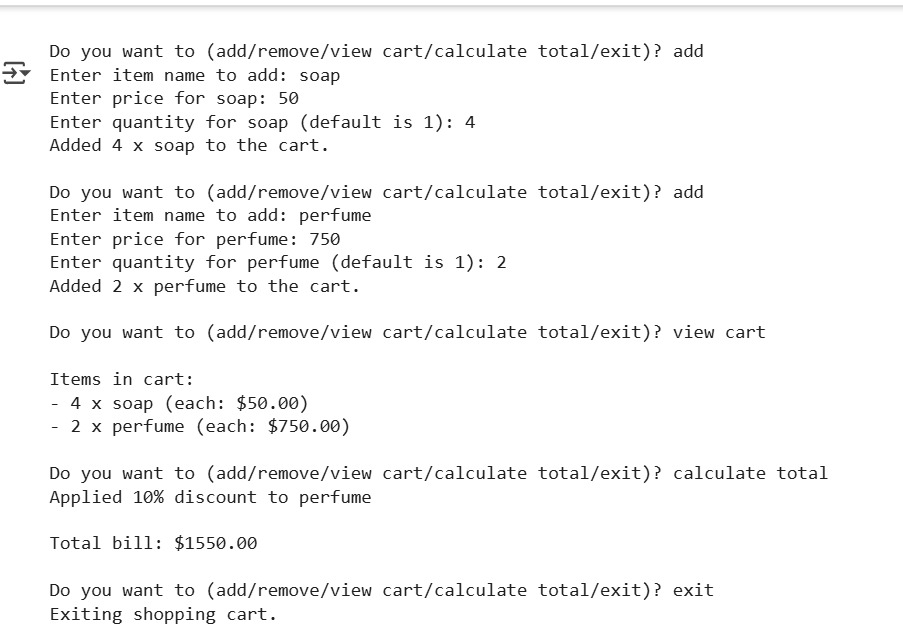
to the previous code generate methods to add\_item,remove\_item,use loop to calculate the total bill using conditional discounts taking user input.

Screeenshot:









Explanation:

class ShoppingCart:: This defines a blueprint for creating shopping cart objects. Each object will have an items list to store the items in the cart.

\_init\_(self):: This is the constructor. When you create a new ShoppingCart object, it initializes an empty list called self.items to hold the items.

add\_item(self, item\_name, item\_price, item\_quantity=1):: This method adds an item to the cart. It takes the item's name, price, and optionally the quantity. If the item is already in the cart, it updates the quantity. Otherwise, it adds the item as a new dictionary to the self.items list.

remove\_item(self, item\_name, item\_quantity=1):: This method removes an item from the cart. It finds the item by item\_name. If the quantity to remove is less than the current quantity, it reduces the quantity. If the quantity is equal, it removes the item entirely from the list. It also handles cases where the item or quantity is not found.

calculate\_total(self):: This method calculates the total cost of all items in the cart. It loops through the self.items list, calculates the cost for each item (price \* quantity), and applies a conditional discount (10% if the item price is over $50). It then adds the item's calculated total to the total\_bill.

cart = ShoppingCart(): This line creates an actual shopping cart object using the ShoppingCart blueprint.

while True:: This starts an infinite loop that keeps running until you specifically tell it to stop.

action = input(...).lower(): Inside the loop, this line asks you what you want to do (add, remove, view cart, calculate total, or exit) and stores your input in the action variable, converting it to lowercase for easier comparison.

if action == "add": ... elif action == "remove": ... elif action == "view cart": ... elif action == "calculate total": ... elif action == "exit": ... else: ...: This block of code checks the value of the action variable. Based on your input, it calls the corresponding method of the cart object (add\_item, remove\_item, calculate\_total), prints the items in the cart, or uses break to exit the loop if you type "exit". If you enter an invalid action, it prints an error message