

**AI ASSISTED CODING**

**Lab-6.4: *AI-Based Code Completion – Classes, Loops, and Conditionals***

**Roll No:** 2503A51L07

**Name:**  SAHASRA

**Batch:** 25TCAICSB19

**Task #1:**

**Prompt:** Start a Python class named Student with attributes name, roll\_number, and marks.Complete those methods for displaying details and checking if marks are above average. Also set 50 marks constant as average and ask for the other details from user.

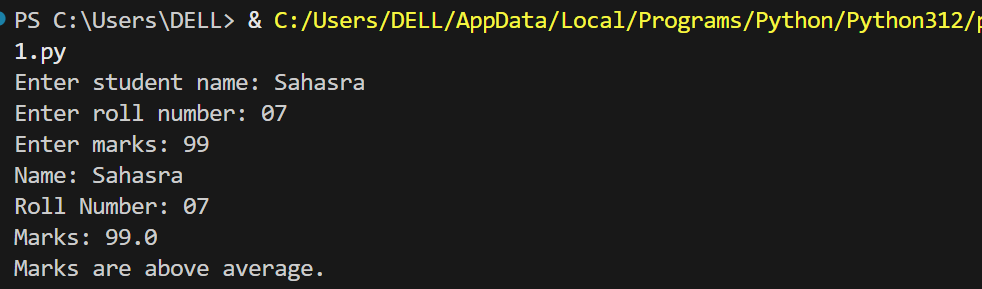


**Code Generated:**



**Output:**







**Observation:**

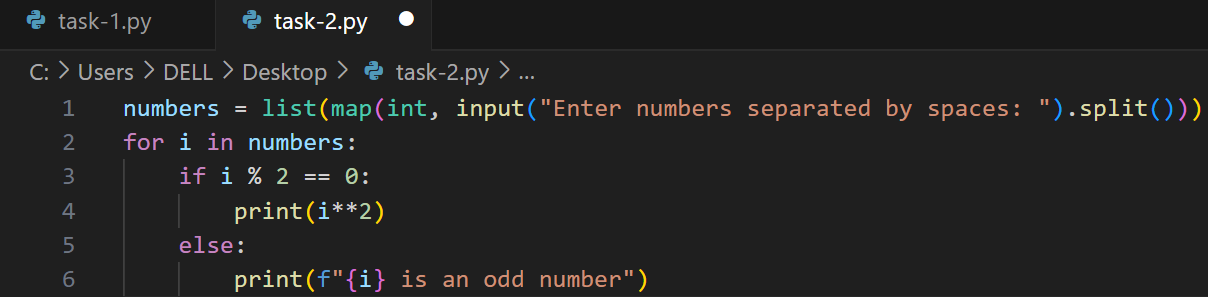
The code defines a Student class with attributes like name, roll number, and marks. It also includes methods to display details and check if the marks are above a constant average (50). User inputs are taken for all attributes, and the program prints whether the student has passed the average criteria. This task demonstrates basic class creation, methods, and conditionals.

**Task #2:**

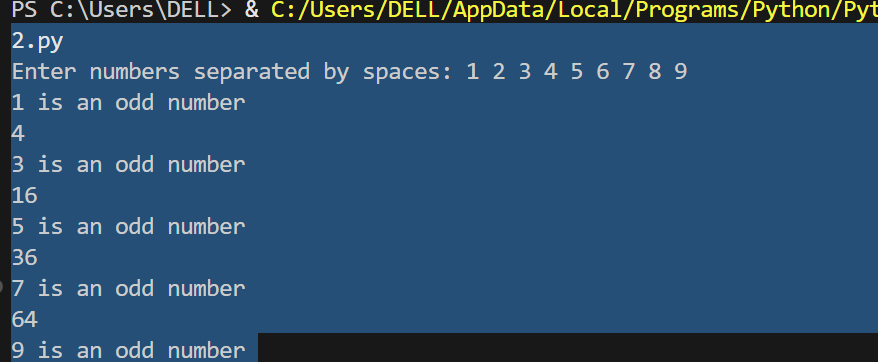
**Prompt:**

Write the first two lines of a for loop to iterate through a list of numbers. Calculate and print the square of even numbers only. Take the list from the users and if(i%2==0) only then square the number,otherwise for the odd numbers give the output as they are odd numbers.

**Code Generated:**



**Output:**

****

**Observation:**

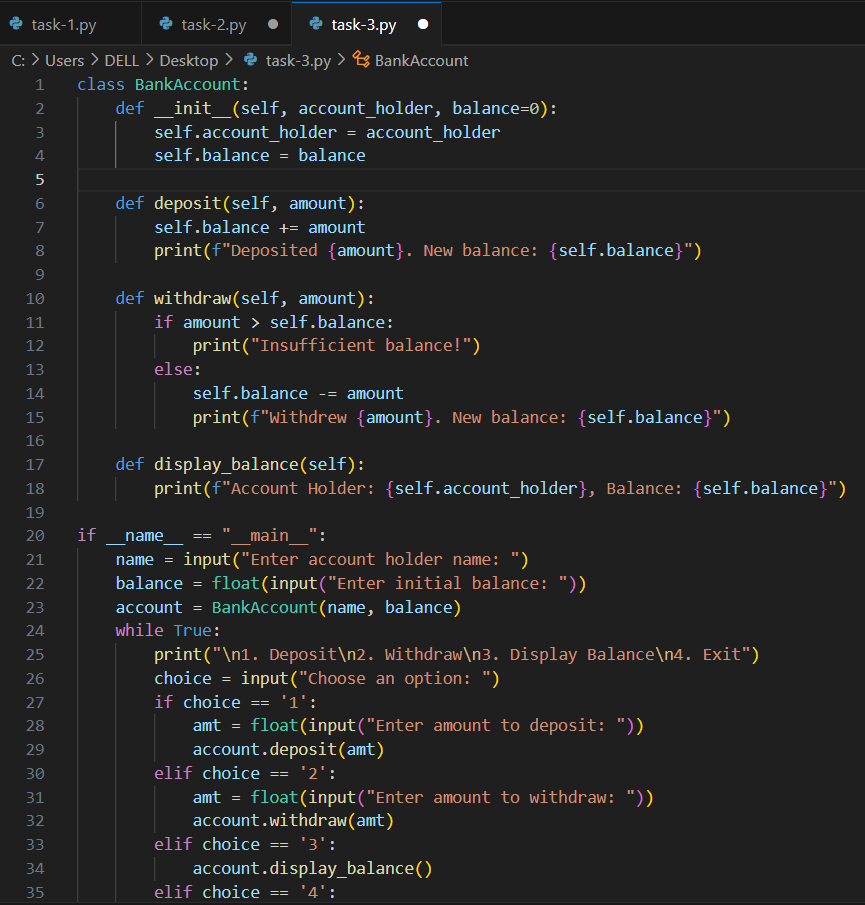
The program takes a list of numbers from the user and iterates using a for loop. It checks if each number is even using the modulus operator. If even, the square is calculated and printed; otherwise, it outputs that the number is odd. This highlights loop control and conditional branching in Python.

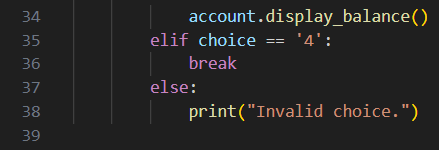
**Task #3:**

**Prompt:**

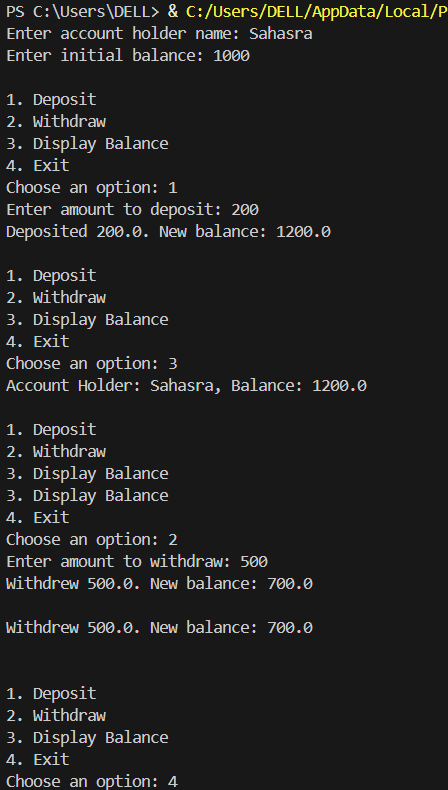
Create a class called BankAccount with attributes account\_holder and balance. Complete methods for deposit(), withdraw(), and check for insufficient balance. Take the inputs for the methods from the user

**Code Generated:**

****

****

**Output:**

****

**Observation:**

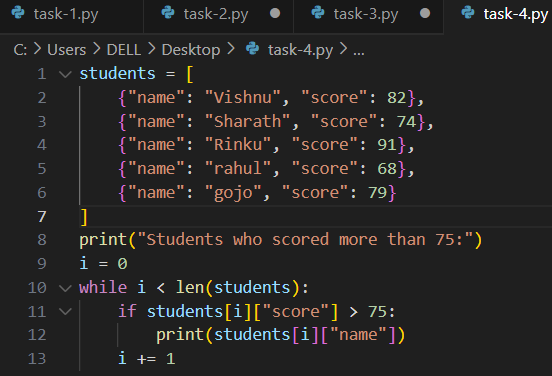
A BankAccount class is created with attributes for account holder and balance. It defines deposit and withdraw methods, while also checking for insufficient balance during withdrawals. The program takes user input for transactions and displays updated balances accordingly. This task demonstrates object-oriented principles with method functionality and conditional checks.

**Task #4:**

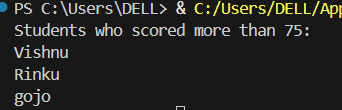
**Prompt:**

Define a list of student dictionaries with keys name and score. Write a while loop to print the names of students who scored more than 75.

**Code Generated:**

****

**Output:**

****

**Observation:**

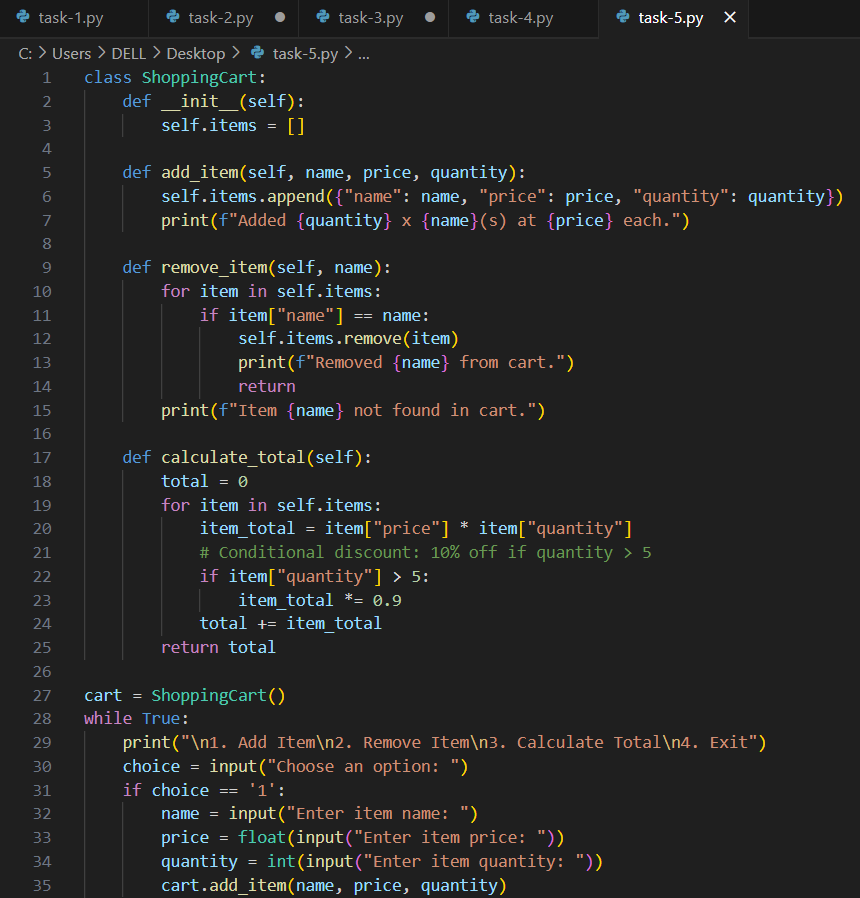
The code defines a list of student dictionaries containing names and scores. A while loop iterates over the list, checking which students scored above 75. Only those student names are displayed as output. This shows the practical use of loops with dictionary data structures and conditional filtering.

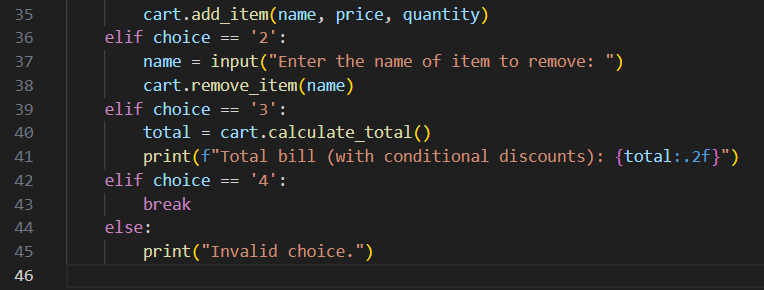
**Task #5:**

**Prompt:**

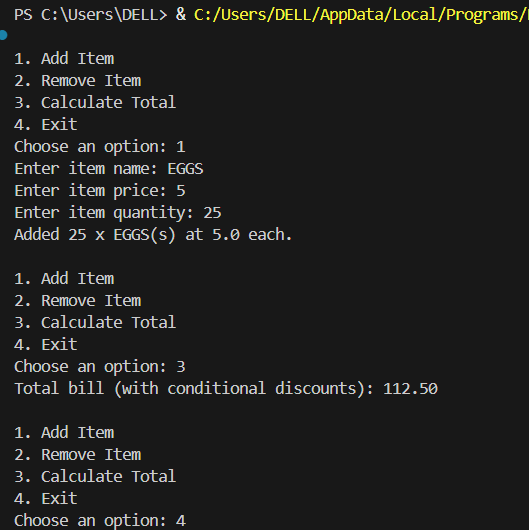
Begin writing a class ShoppingCart with an empty items list. Generate methods to add\_item, remove\_item, and use a loop to calculate the total bill using conditional discounts. And ask for the inputs from the user.

**Code Generated:**

****

****

**Output:**

****

**Observation:**

A ShoppingCart class is implemented with methods to add and remove items. A loop is used to calculate the total bill, and conditional statements apply discounts if applicable. The user inputs item details, and the program computes the final total. This demonstrates classes, list operations, loops, and conditions together in one program.