# **AI ASSISTED CODING**

# **Lab 6 : AI-Based Code Completion – Classes, Loops, and Conditionals**

**Assignment number:** 6.4

**Roll No:** 2503A51L36

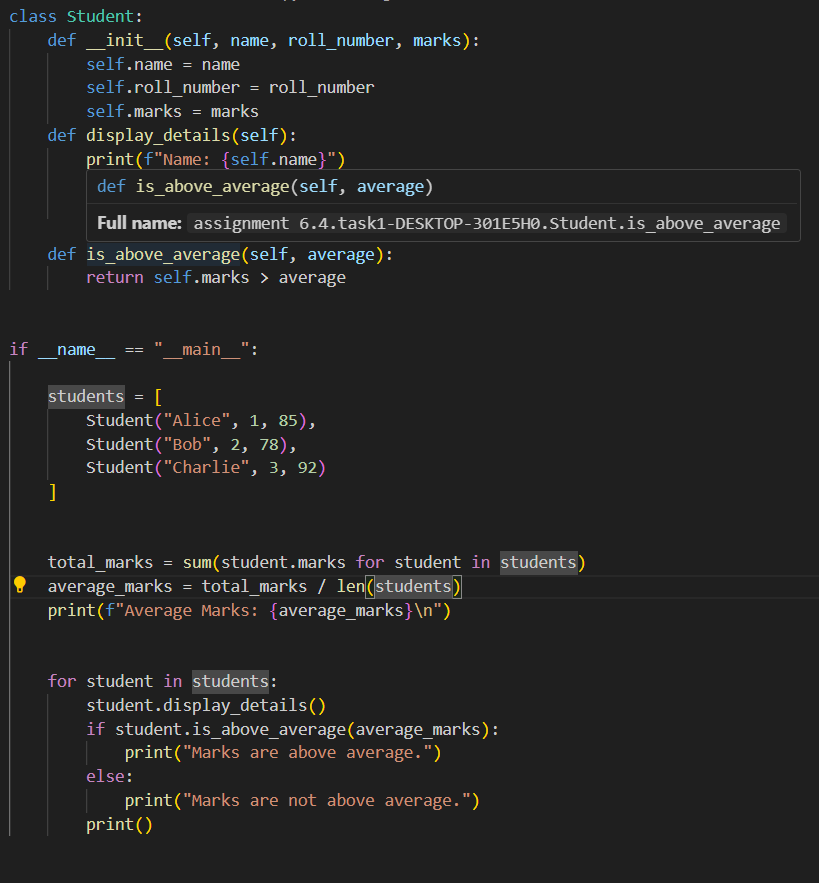
**Name:**  Suhana Rehan

**Batch:** 25TCAICSB20

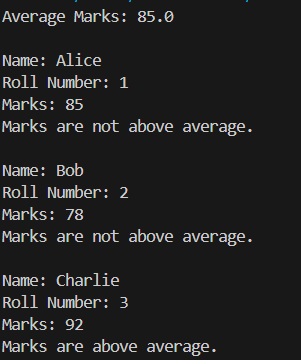
**Task 1:**

**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:**



**Output:**

****

**Observation:**

 1. The program defines a Student class with attributes for name, roll number, and marks.

 2. It calculates the average marks of all students in the list.

 3. For each student, it displays their details and checks if their marks are above the average.

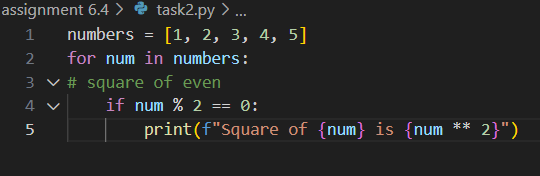
 4. The output clearly indicates which students have marks above or below the average.

 5. The code demonstrates basic class usage, list comprehension, and conditional logic in Python.

**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:**



**Output:**

****

**Observation:**

1. The program iterates through a list of numbers from 1 to 5.

  2. For each number, it checks if the number is even using the modulo operator.

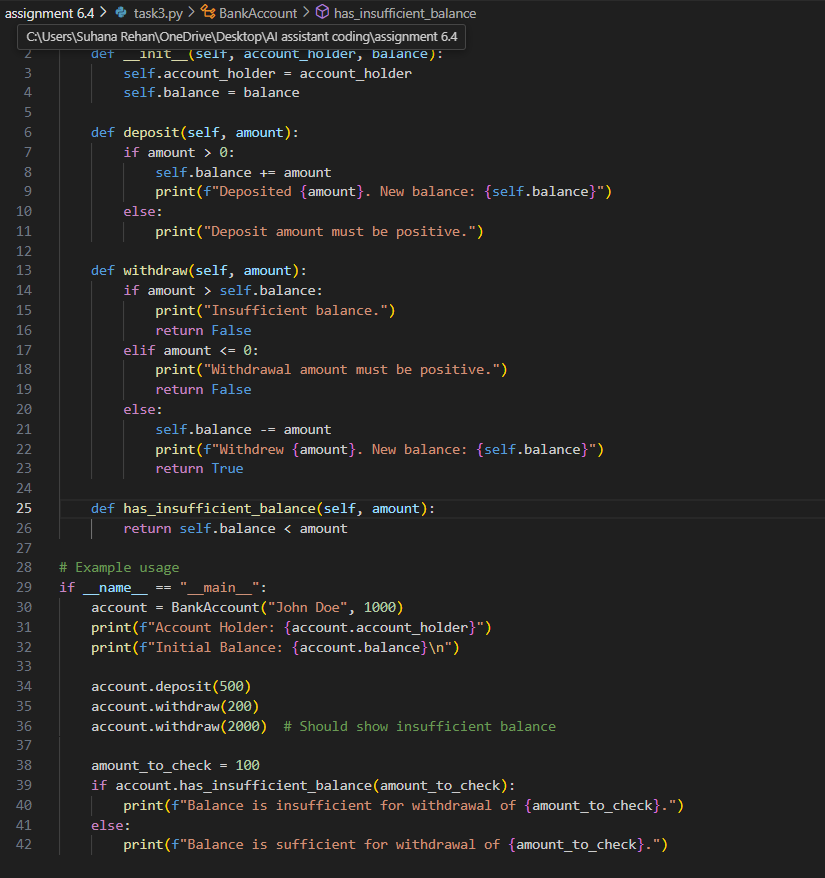
  3. If the number is even, it prints the square of that number.

  4. Only even numbers (2 and 4) are processed and displayed in the output.

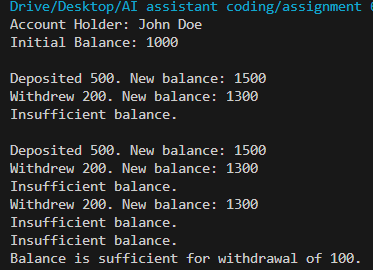
**Task 3:**

**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:**

****

**Output:**

****

**Observation:**

1. The program defines a BankAccount class with methods for deposit, withdrawal, and balance checking.

 2. It validates deposit and withdrawal amounts, ensuring only positive values are processed.

 3. The withdrawal method checks for sufficient balance before allowing the transaction.

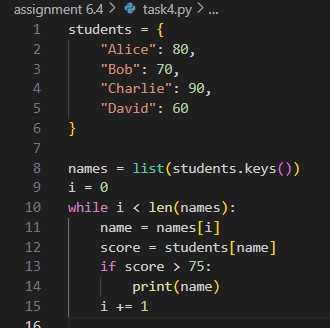
 4. The example usage demonstrates account creation, deposit, withdrawal, and balance

sufficiency 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 :**

****

**Output:**

****

**Observation:**

 1. The program uses a dictionary to store student names and their scores.

 2. It iterates through the dictionary using a while loop and a list of names.

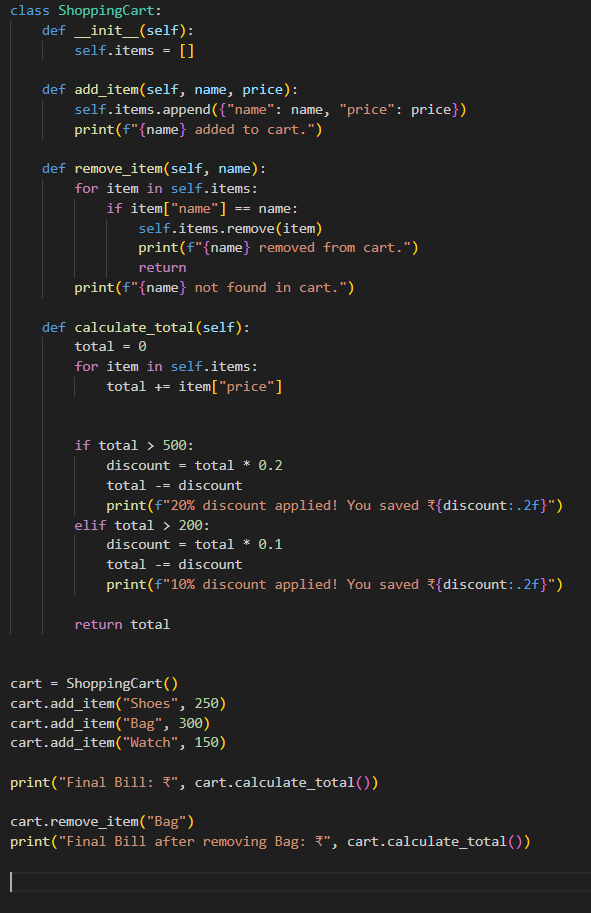
 3. For each student, it checks if their score is greater than 75 and prints their name if true.

 4. Only students with scores above 75 (Alice and Charlie) are displayed in the output.

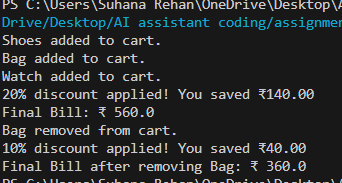
**Task #5:**

**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 :**

****

**Output:**

****

**Observation:**

1. The program defines a ShoppingCart class to manage items and calculate the total bill.

2. Items can be added or removed from the cart, with feedback provided for each action.

3. The total bill calculation includes discount logic: 20% for totals above ₹500, 10% for totals above ₹200.

4. The example usage demonstrates adding items, calculating the bill, removing an item, and recalculating the bill.

5. The code illustrates object-oriented programming, list manipulation, and conditional logic in Python.