Assignment:6.4

Name:Aashutosh

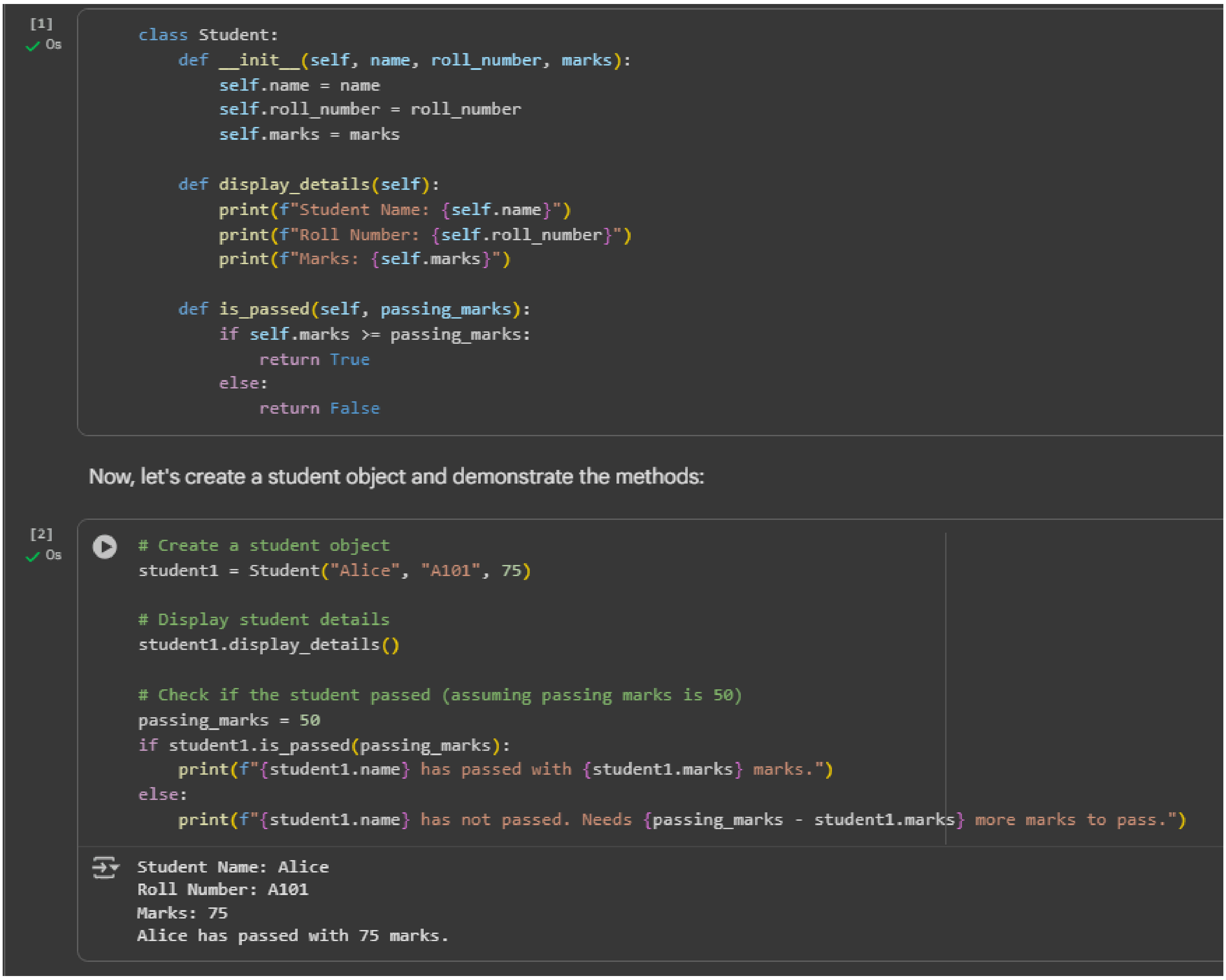
Hall Tkt:2403a51316

Task1: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.

Ans:-

Prompt:

Write a python class has attribute name,roll number and marks and methods like display_details and is_passed() using if-else conditons.



```
class Student:
    def __init__(self, name, roll_number, marks):
        self.name = name
        self.roll_number = roll_number
        self.marks = marks

    def display_details(self):
        print(f"Student Name: {self.name}")
        print(f"Roll Number: {self.roll_number}")
        print(f"Marks: {self.marks}")

    def is_passed(self, passing_marks):
        if self.marks >= passing_marks:
            return True
        else:
            return False
```

Now, let's create a student object and demonstrate the methods:

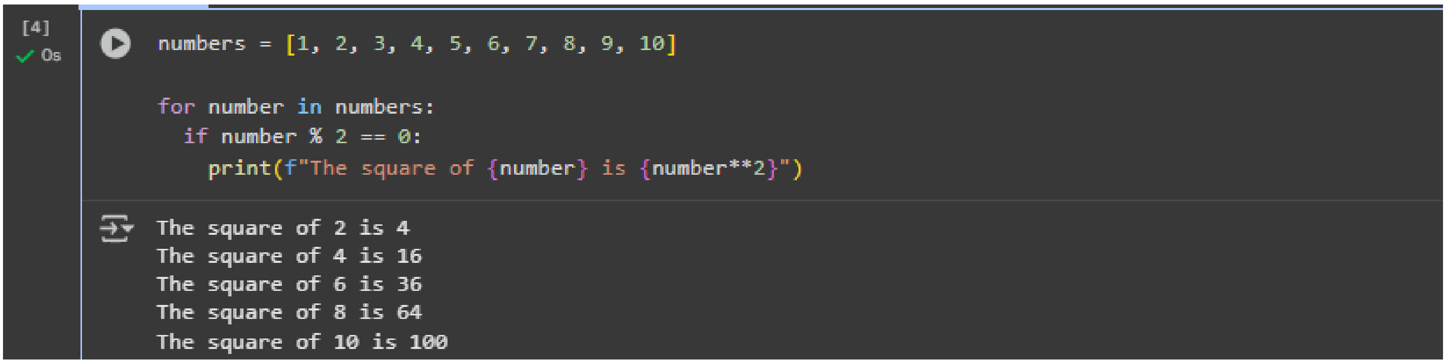
```
# Create a student object
student1 = Student("Alice", "A101", 75)

# Display student details
student1.display_details()

# Check if the student passed (assuming passing marks is 50)
passing_marks = 50
if student1.is_passed(passing_marks):
    print(f"{student1.name} has passed with {student1.marks} marks.")
else:
    print(f"{student1.name} has not passed. Needs {passing_marks - student1.marks} more marks to pass.")
```

```
Student Name: Alice
Roll Number: A101
Marks: 75
Alice has passed with 75 marks.
```

Task2: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 number only.

Ans:-

Prompt:-Write a python program that literate over a list of number if the number is even then display the square of even number only.



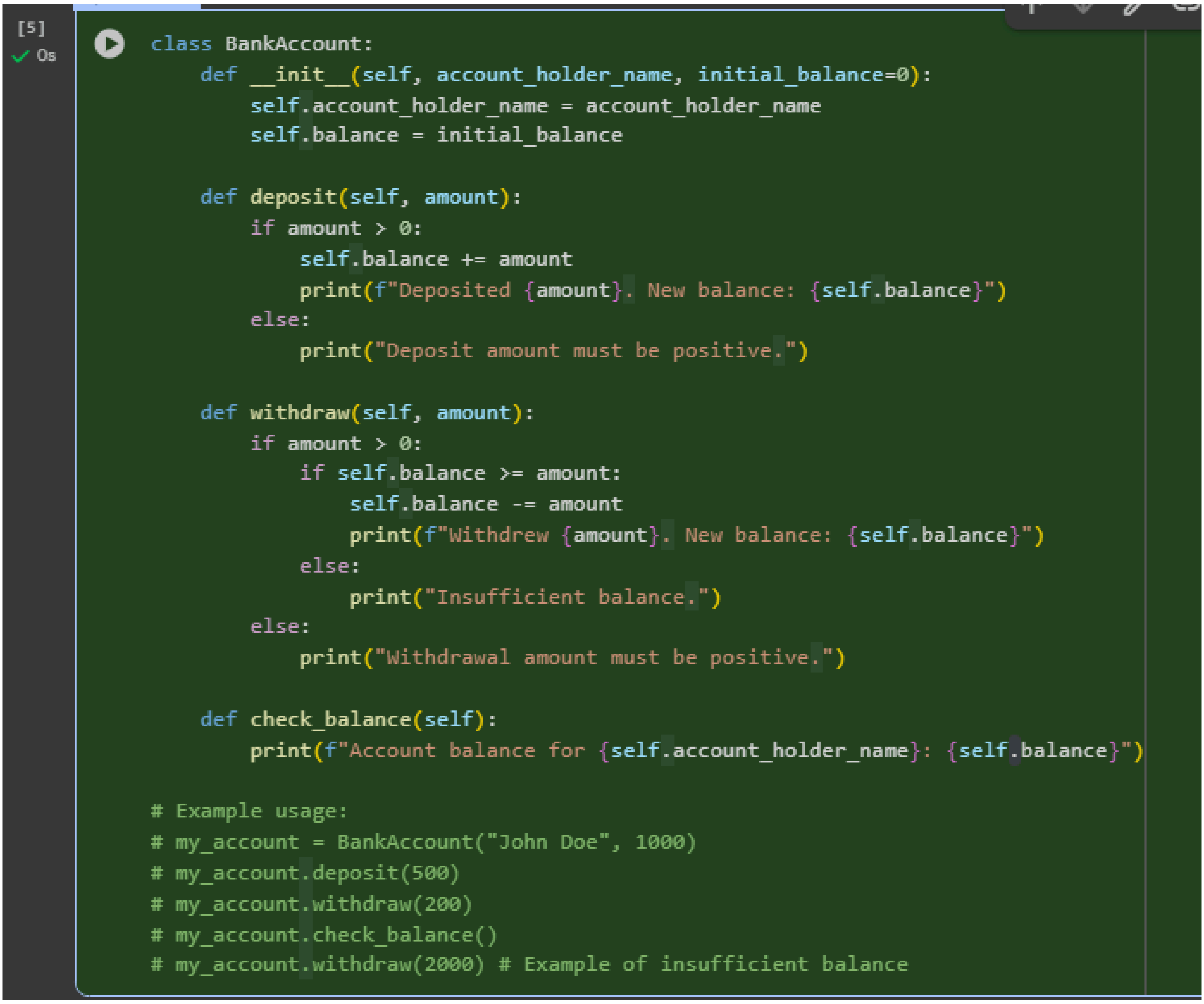
```
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

for number in numbers:
  if number % 2 == 0:
    print(f"The square of {number} is {number**2}")
```

```
The square of 2 is 4
The square of 4 is 16
The square of 6 is 36
The square of 8 is 64
The square of 10 is 100
```

Task3: Create a class BankAccount with attributes account_holder and balance. Use copilot to complete methods for diposit(),withdraw(), and check for insufficient balance.

Ans:-

Prompt:- Create a class bankaccount with attribute account_holder name and his balance and then add methods like deposit, withdraw and insufficient balance.

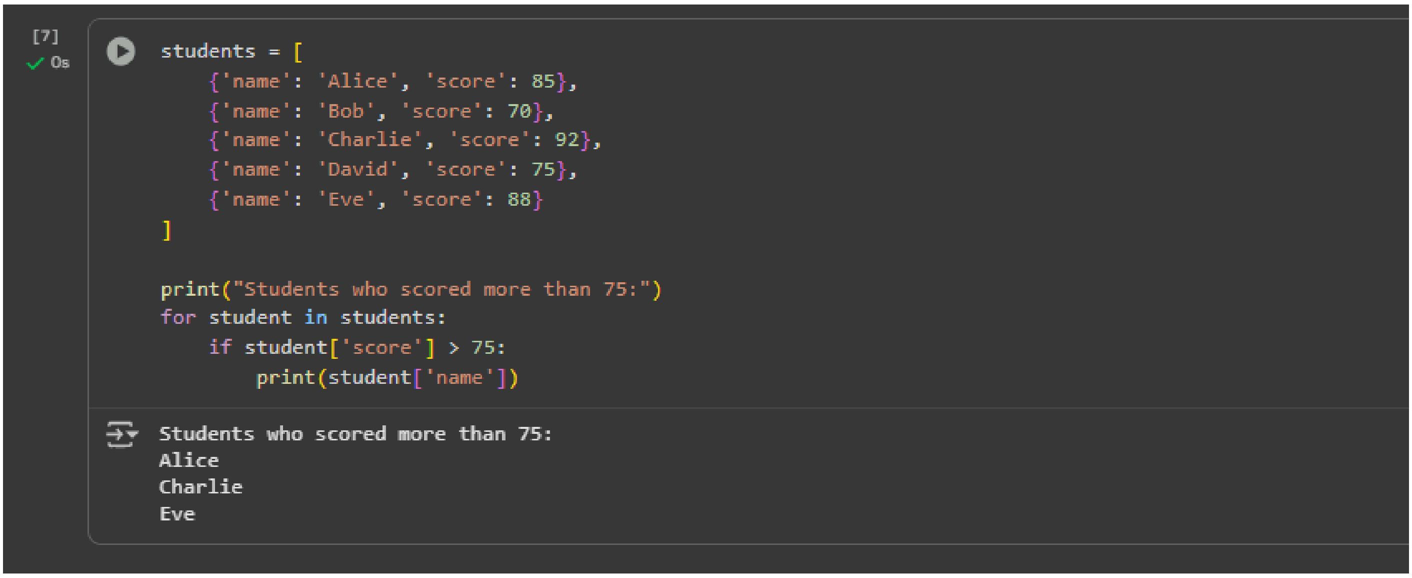


```
class BankAccount:
    def __init__(self, account_holder_name, initial_balance=0):
        self.account_holder_name = account_holder_name
        self.balance = initial_balance

    def deposit(self, amount):
        if amount > 0:
            self.balance += amount
            print(f"Deposited {amount}. New balance: {self.balance}")
        else:
            print("Deposit amount must be positive.")

    def withdraw(self, amount):
        if amount > 0:
            if self.balance >= amount:
                self.balance -= amount
                print(f"Withdrew {amount}. New balance: {self.balance}")
            else:
                print("Insufficient balance.")
        else:
            print("Withdrawal amount must be positive.")

    def check_balance(self):
        print(f"Account balance for {self.account_holder_name}: {self.balance}")

# Example usage:
# my_account = BankAccount("John Doe", 1000)
# my_account.deposit(500)
# my_account.withdraw(200)
# my_account.check_balance()
# my_account.withdraw(2000) # Example of insufficient balance
```

Task4:Define a list of student dictionaries with keys name and score. Ask copilot to write a while loop to print the name of students who scored more than 75.

Ans:-

Prompt:-create a list of student dictionary with keys name and score and print the names of the student who scored more than 75.



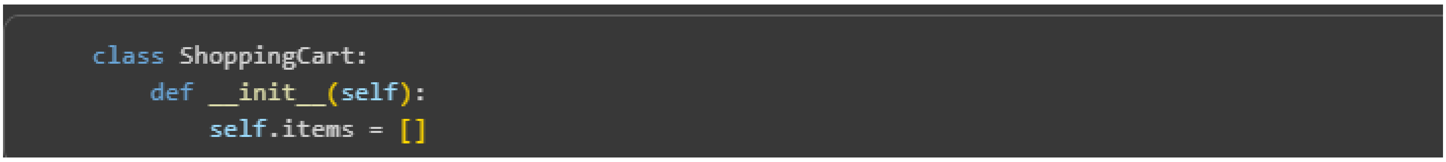
```
students = [
    {'name': 'Alice', 'score': 85},
    {'name': 'Bob', 'score': 70},
    {'name': 'Charlie', 'score': 92},
    {'name': 'David', 'score': 75},
    {'name': 'Eve', 'score': 88}
]

print("Students who scored more than 75:")
for student in students:
    if student['score'] > 75:
        print(student['name'])
```

```
Students who scored more than 75:
Alice
Charlie
Eve
```

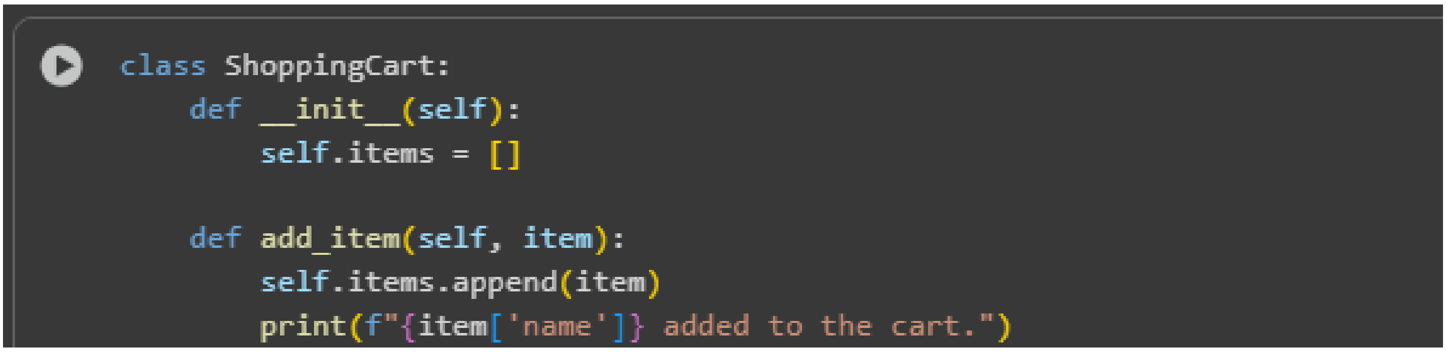
Task5:Begin writing a class shoppingcart with empty items list. Prompt copilot to generate methods to add_item, remove_item, and use a loop to calculate the total bill using conditonal discounts.

Ans:-

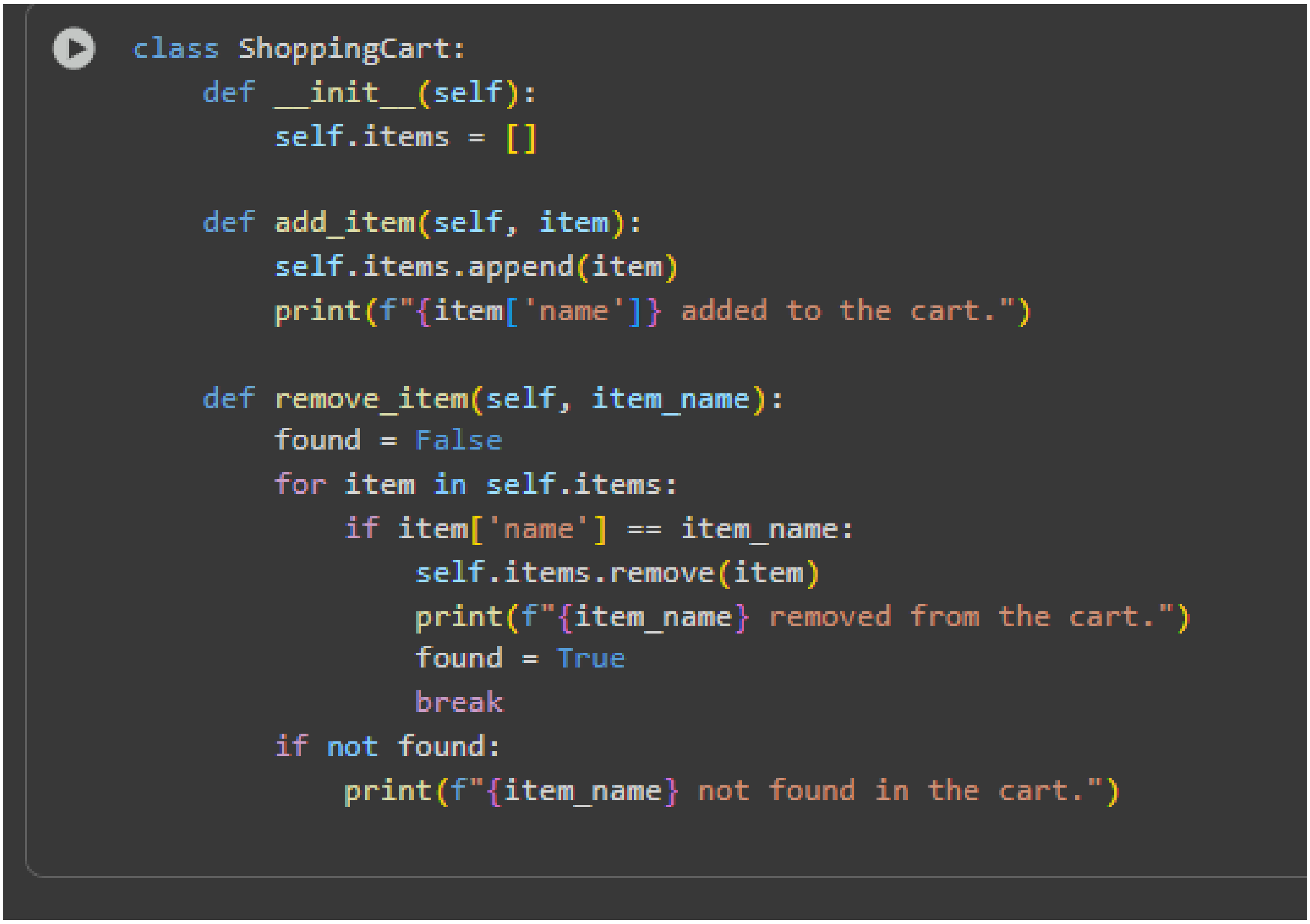
Prompt:-create a class shoppingcart with empty list and add methods add_item and remove_item and use a loop to calculate total bill using conditional discount.



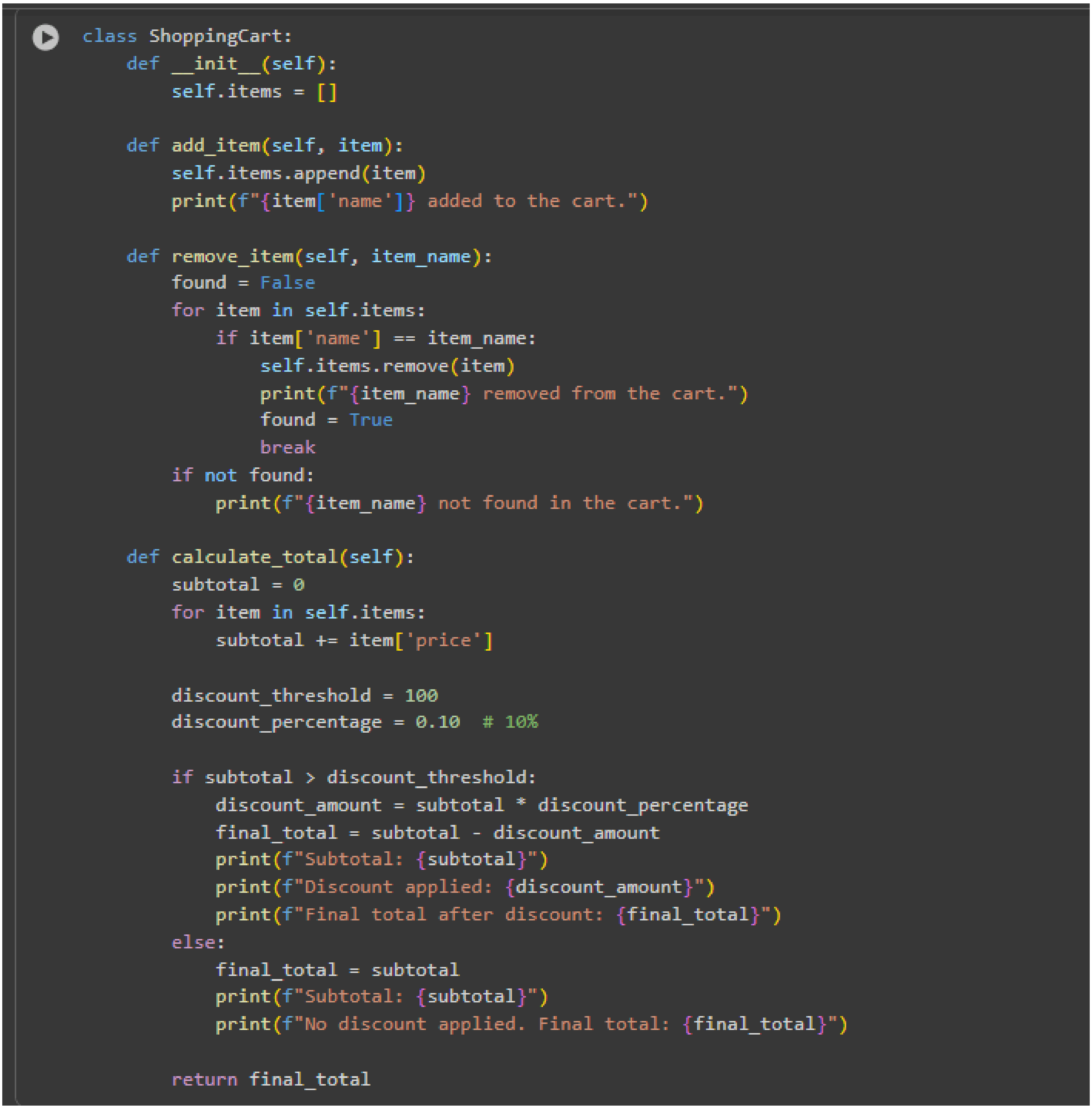
```
class ShoppingCart:
    def __init__(self):
        self.items = []
```



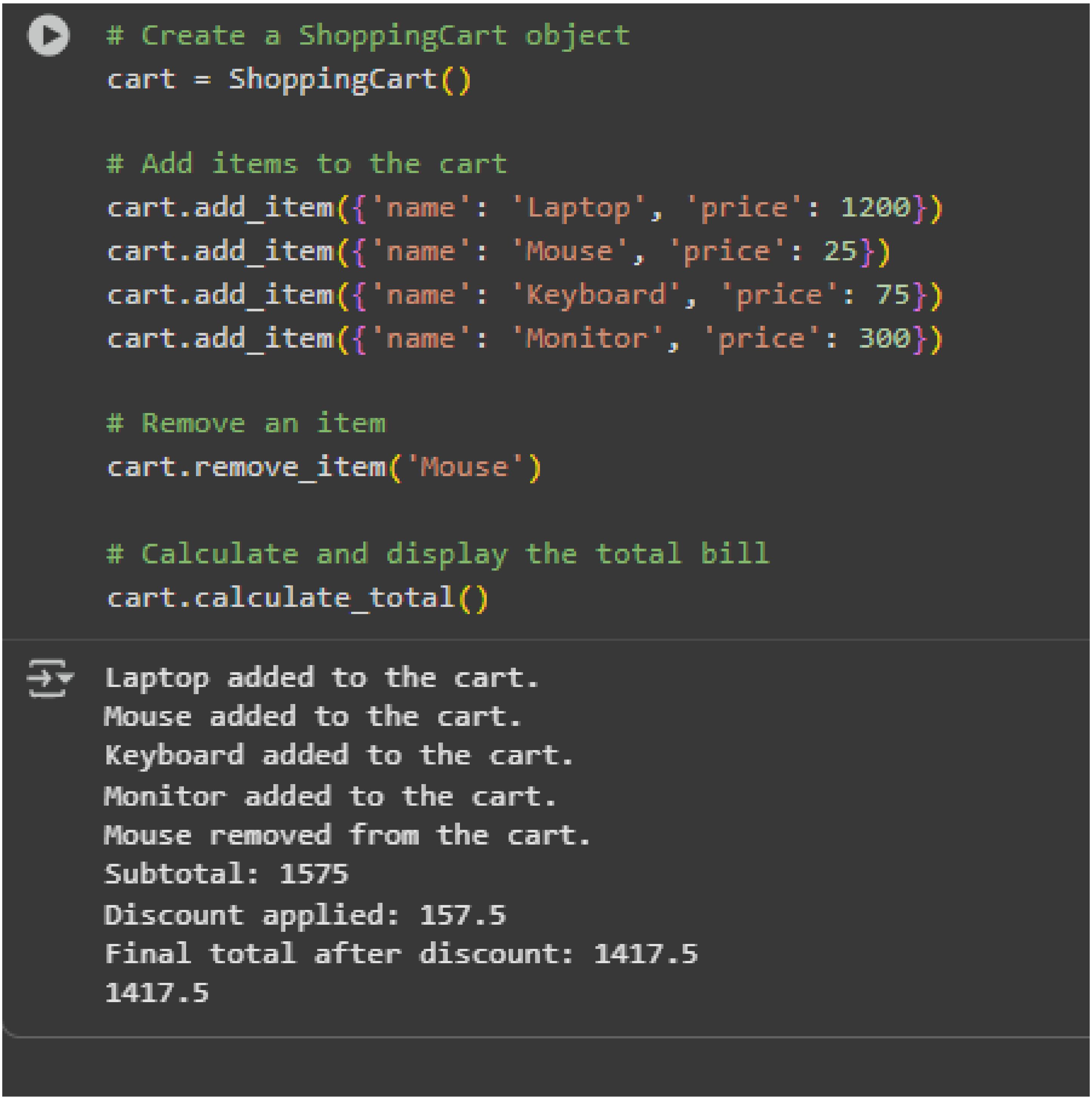
```
class ShoppingCart:
    def __init__(self):
        self.items = []

    def add_item(self, item):
        self.items.append(item)
        print(f"{item['name']} added to the cart.")
```



```python
class ShoppingCart:
    def __init__(self):
        self.items = []

    def add_item(self, item):
        self.items.append(item)
        print(f"{item['name']} added to the cart.")

    def remove_item(self, item_name):
        found = False
        for item in self.items:
            if item['name'] == item_name:
                self.items.remove(item)
                print(f"{item_name} removed from the cart.")
                found = True
                break
        if not found:
            print(f"{item_name} not found in the cart.")
```



```
class ShoppingCart:
    def __init__(self):
        self.items = []

    def add_item(self, item):
        self.items.append(item)
        print(f"{item['name']} added to the cart.")

    def remove_item(self, item_name):
        found = False
        for item in self.items:
            if item['name'] == item_name:
                self.items.remove(item)
                print(f"{item_name} removed from the cart.")
                found = True
                break
        if not found:
            print(f"{item_name} not found in the cart.")

    def calculate_total(self):
        subtotal = 0
        for item in self.items:
            subtotal += item['price']

        discount_threshold = 100
        discount_percentage = 0.10  # 10%

        if subtotal > discount_threshold:
            discount_amount = subtotal * discount_percentage
            final_total = subtotal - discount_amount
            print(f"Subtotal: {subtotal}")
            print(f"Discount applied: {discount_amount}")
            print(f"Final total after discount: {final_total}")
        else:
            final_total = subtotal
            print(f"Subtotal: {subtotal}")
            print(f"No discount applied. Final total: {final_total}")

        return final_total
```



```python
# Create a ShoppingCart object
cart = ShoppingCart()

# Add items to the cart
cart.add_item({'name': 'Laptop', 'price': 1200})
cart.add_item({'name': 'Mouse', 'price': 25})
cart.add_item({'name': 'Keyboard', 'price': 75})
cart.add_item({'name': 'Monitor', 'price': 300})

# Remove an item
cart.remove_item('Mouse')

# Calculate and display the total bill
cart.calculate_total()
```

```
Laptop added to the cart.
Mouse added to the cart.
Keyboard added to the cart.
Monitor added to the cart.
Mouse removed from the cart.
Subtotal: 1575
Discount applied: 157.5
Final total after discount: 1417.5
1417.5
```