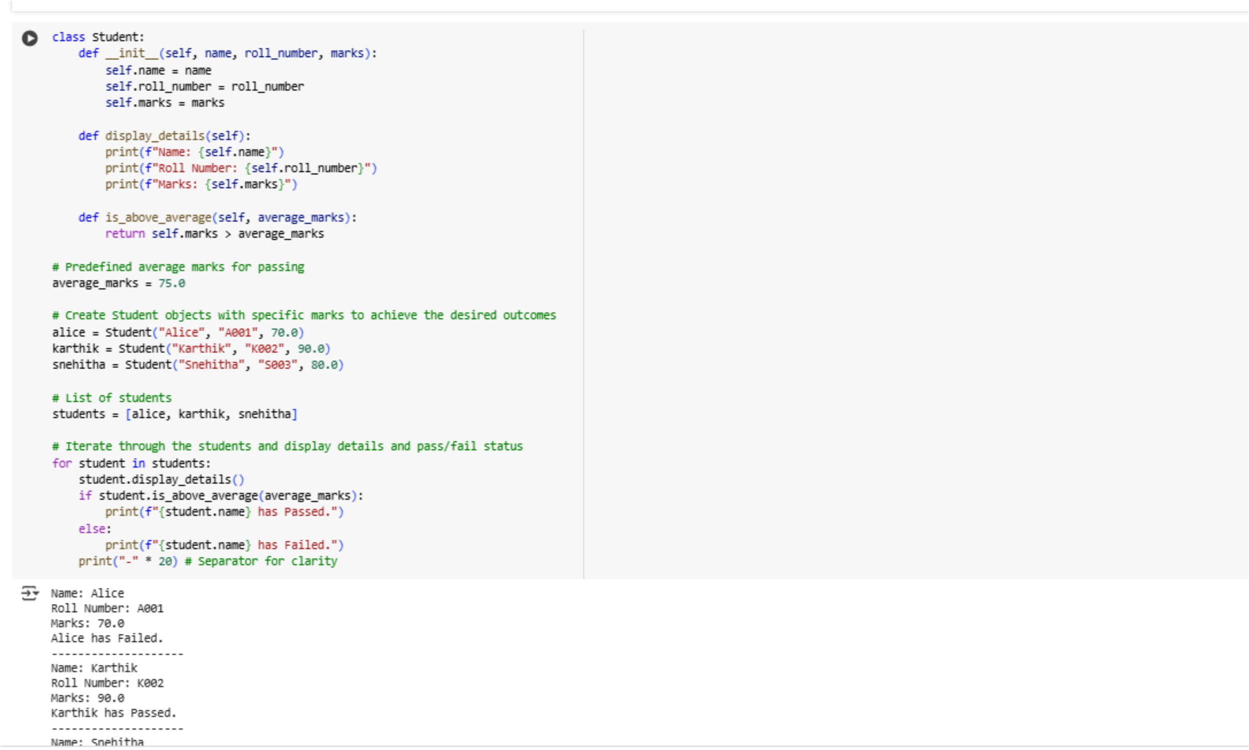
Assignment 6.4

Name: I. karthik

Roll no:2403A52200

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.



```python
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"Name: {self.name}")
        print(f"Roll Number: {self.roll_number}")
        print(f"Marks: {self.marks}")

    def is_above_average(self, average_marks):
        return self.marks > average_marks

# Predefined average marks for passing
average_marks = 75.0

# Create Student objects with specific marks to achieve the desired outcomes
alice = Student("Alice", "A001", 70.0)
karthik = Student("Karthik", "K002", 90.0)
snehitha = Student("Snehitha", "S003", 80.0)

# List of students
students = [alice, karthik, snehitha]

# Iterate through the students and display details and pass/fail status
for student in students:
    student.display_details()
    if student.is_above_average(average_marks):
        print(f"{student.name} has Passed.")
    else:
        print(f"{student.name} has Failed.")
    print("-" * 20) # Separator for clarity
```

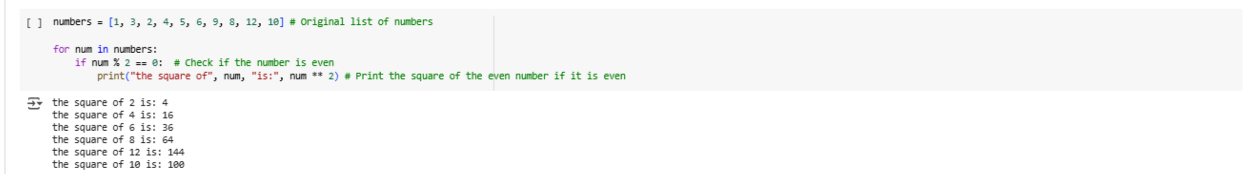
```
Name: Alice
Roll Number: A001
Marks: 70.0
Alice has Failed.
--------------------
Name: Karthik
Roll Number: K002
Marks: 90.0
Karthik has Passed.
--------------------
Name: Snehitha
```

Prompt:

Generate a python program for class named Student with attributes name, roll_number, and marks and displaying details and checking if marks are above average.

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.



```python
numbers = [1, 3, 2, 4, 5, 6, 9, 8, 12, 10] # Original list of numbers

for num in numbers:
    if num % 2 == 0:  # Check if the number is even
        print("the square of", num, "is:", num ** 2) # Print the square of the even number if it is even
```

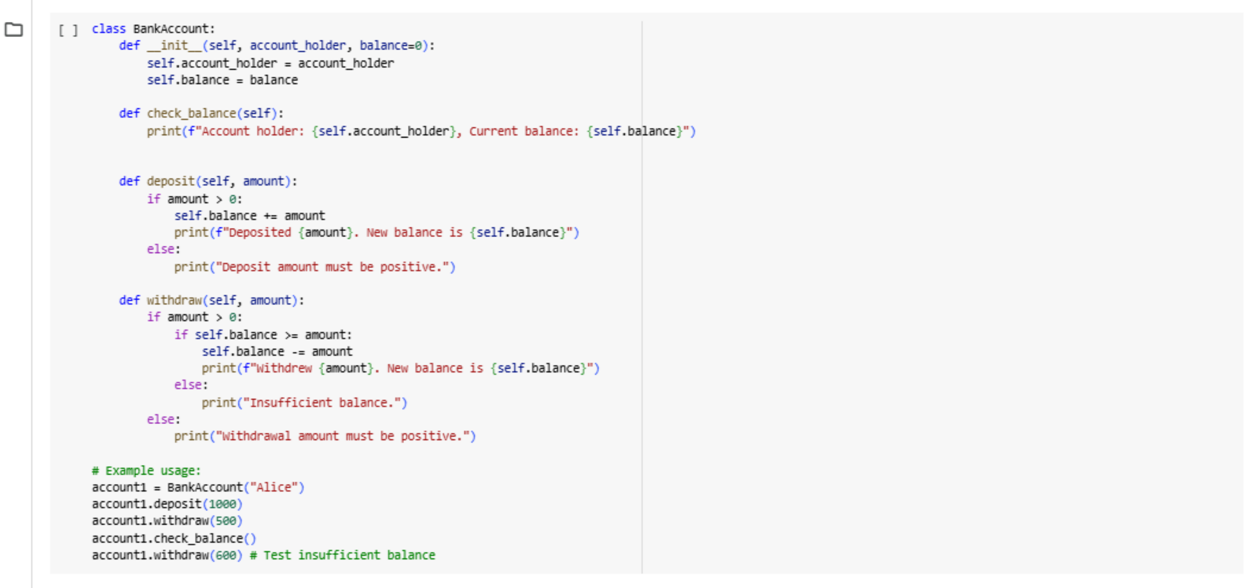
```
the square of 2 is: 4
the square of 4 is: 16
the square of 6 is: 36
the square of 8 is: 64
the square of 12 is: 144
the square of 10 is: 100
```

Prompt:

Generate a python program for to print the square of even numbers only by using conditional logic.

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.



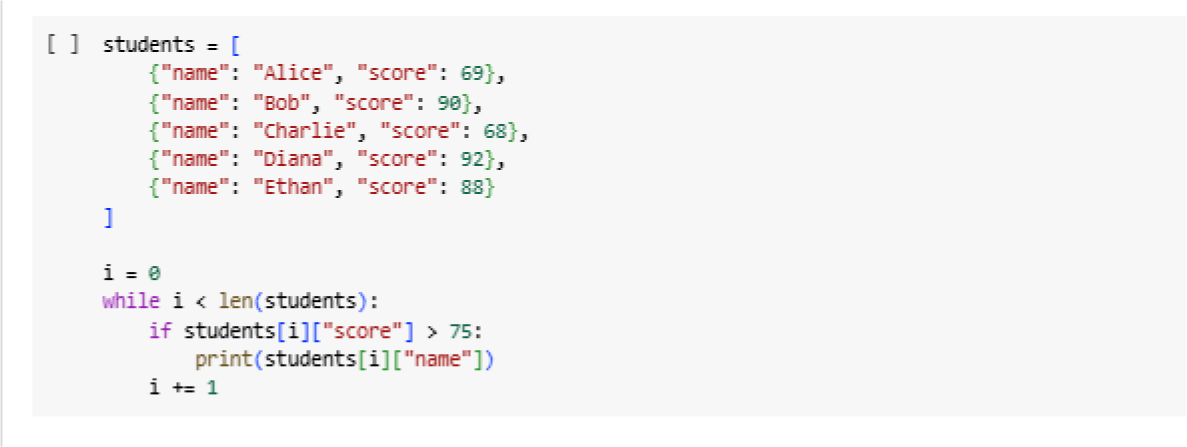
```python
class BankAccount:
    def __init__(self, account_holder, balance=0):
        self.account_holder = account_holder
        self.balance = balance

    def check_balance(self):
        print(f"Account holder: {self.account_holder}, Current balance: {self.balance}")


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

# Example usage:
account1 = BankAccount("Alice")
account1.deposit(1000)
account1.withdraw(500)
account1.check_balance()
account1.withdraw(600) # Test insufficient balance
```

Prompt:

Generate a python program for BankAccount with attributes account_holder and balance.And add methods for deposit(), withdraw(), and check for insufficient balance by using if conditions.

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.



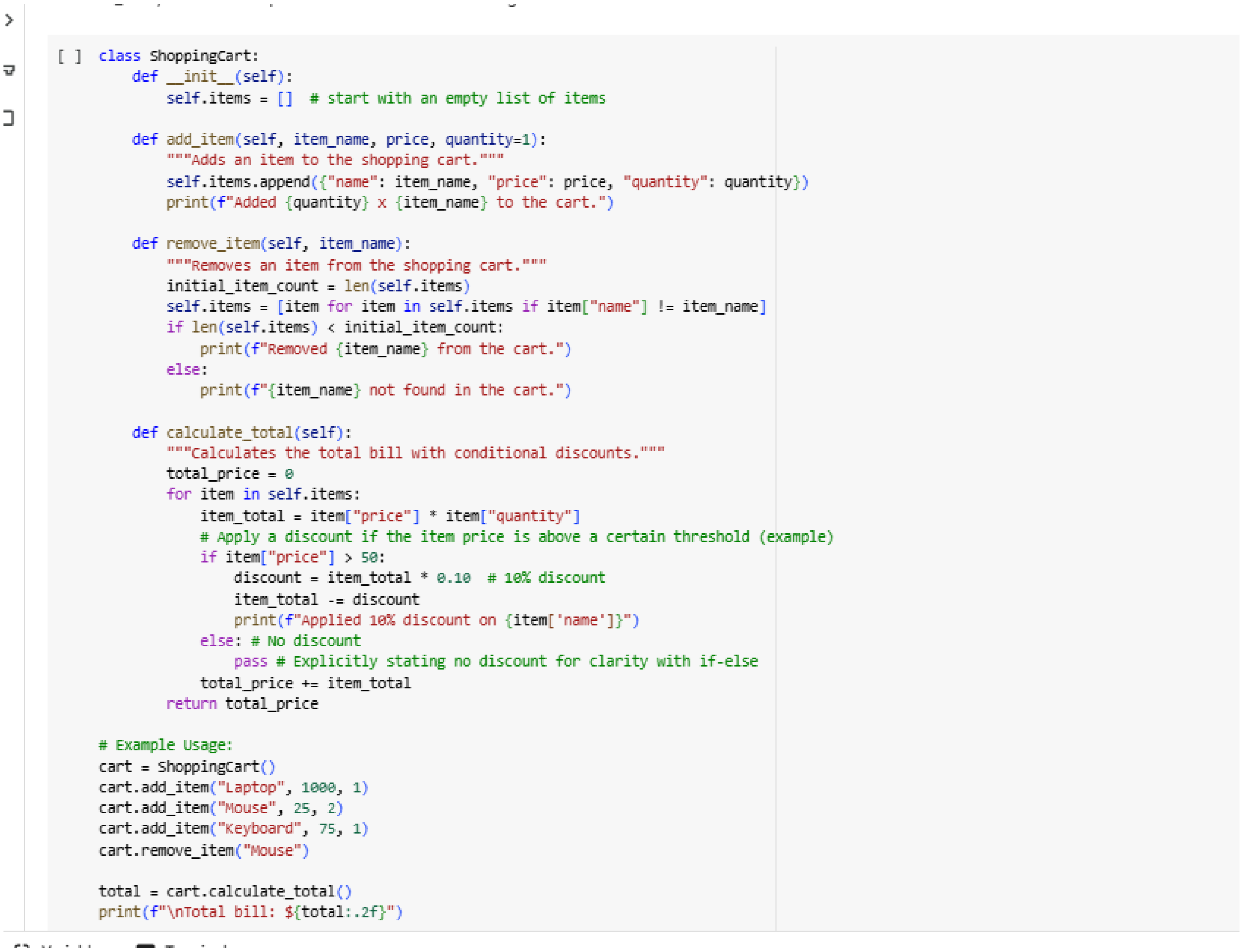
```
[ ]  students = [
         {"name": "Alice", "score": 69},
         {"name": "Bob", "score": 90},
         {"name": "Charlie", "score": 68},
         {"name": "Diana", "score": 92},
         {"name": "Ethan", "score": 88}
     ]

     i = 0
     while i < len(students):
         if students[i]["score"] > 75:
             print(students[i]["name"])
         i += 1
```

Prompt:

Generate a python program by using while loop to print the names of students who scored more than 75.

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.



```
[ ]  class ShoppingCart:
         def __init__(self):
             self.items = []  # start with an empty list of items

         def add_item(self, item_name, price, quantity=1):
             """Adds an item to the shopping cart."""
             self.items.append({"name": item_name, "price": price, "quantity": quantity})
             print(f"Added {quantity} x {item_name} to the cart.")

         def remove_item(self, item_name):
             """Removes an item from the shopping cart."""
             initial_item_count = len(self.items)
             self.items = [item for item in self.items if item["name"] != item_name]
             if len(self.items) < initial_item_count:
                 print(f"Removed {item_name} from the cart.")
             else:
                 print(f"{item_name} not found in the cart.")

         def calculate_total(self):
             """Calculates the total bill with conditional discounts."""
             total_price = 0
             for item in self.items:
                 item_total = item["price"] * item["quantity"]
                 # Apply a discount if the item price is above a certain threshold (example)
                 if item["price"] > 50:
                     discount = item_total * 0.10  # 10% discount
                     item_total -= discount
                     print(f"Applied 10% discount on {item['name']}")
                 else: # No discount
                     pass # Explicitly stating no discount for clarity with if-else
                 total_price += item_total
             return total_price

     # Example Usage:
     cart = ShoppingCart()
     cart.add_item("Laptop", 1000, 1)
     cart.add_item("Mouse", 25, 2)
     cart.add_item("Keyboard", 75, 1)
     cart.remove_item("Mouse")

     total = cart.calculate_total()
     print(f"\nTotal bill: ${total:.2f}")
```

Prompt:

Generate a python program for methods to add_item, remove_item, and use a loop to calculate the total bill using conditional discounts by using if else stmts.