LAB-3 Assignment 6.4

Roll No: 2403A52085

Batch: 4

Name: G. Ruthvik Reddy

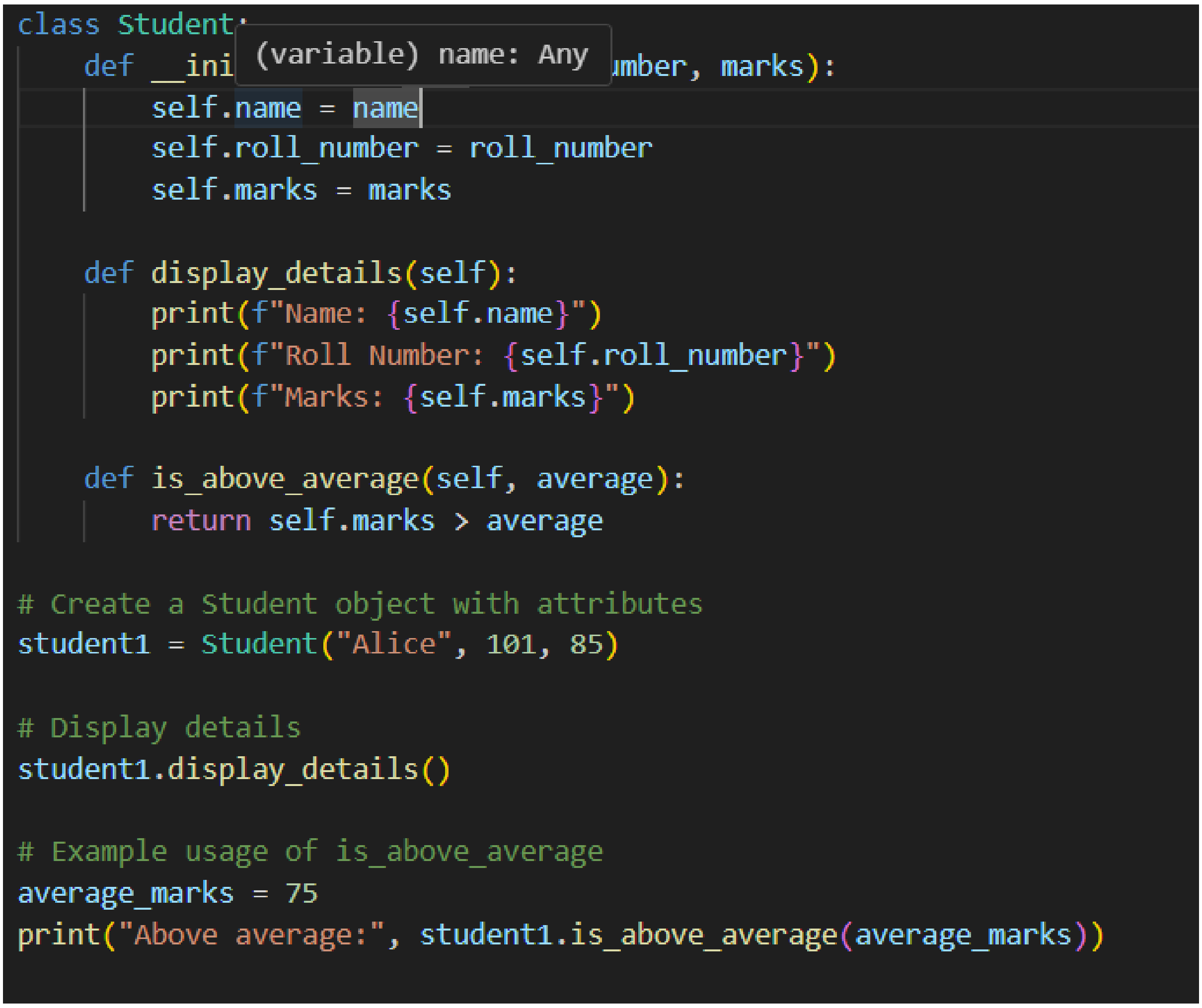
Task 1:

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.

Prompt:

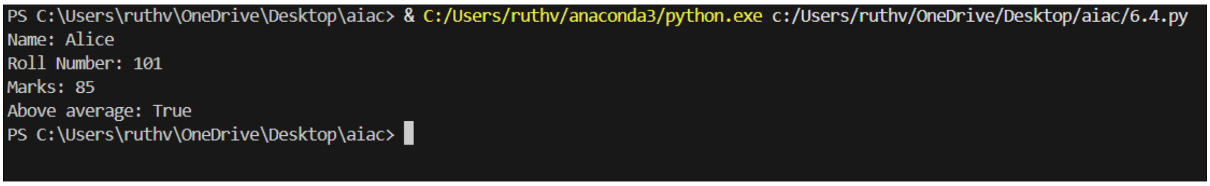
Complete methods for displaying details and checking if marks are above average.

Code:



```
class Student:
    def __ini                   umber, 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):
        return self.marks > average

# Create a Student object with attributes
student1 = Student("Alice", 101, 85)

# Display details
student1.display_details()

# Example usage of is_above_average
average_marks = 75
print("Above average:", student1.is_above_average(average_marks))
```

Output:



```
PS C:\Users\ruthv\OneDrive\Desktop\aiac> & C:/Users/ruthv/anaconda3/python.exe c:/Users/ruthv/OneDrive/Desktop/aiac/6.4.py
Name: Alice
Roll Number: 101
Marks: 85
Above average: True
PS C:\Users\ruthv\OneDrive\Desktop\aiac> 
```

Observations:

Github Copilet swiftly understood the given prompt and generated the Student class with appropriate methods, I had to give it another prompt to call the function.

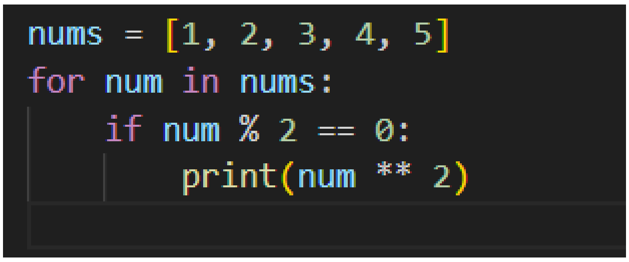
Task 2:

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.

Prompt:

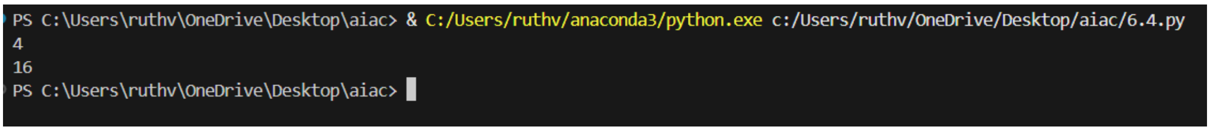
complete the code to print the square of even numbers in the list.

Code:



```python
nums = [1, 2, 3, 4, 5]
for num in nums:
    if num % 2 == 0:
        print(num ** 2)
```

Output:



```
PS C:\Users\ruthv\OneDrive\Desktop\aiac> & C:/Users/ruthv/anaconda3/python.exe c:/Users/ruthv/OneDrive/Desktop/aiac/6.4.py
4
16
PS C:\Users\ruthv\OneDrive\Desktop\aiac> 
```

Observations:

Github Copilet understood the command and finished my code to execute the desired output. It printed the squares of only the even numbers in the list, i.e 2, 4.

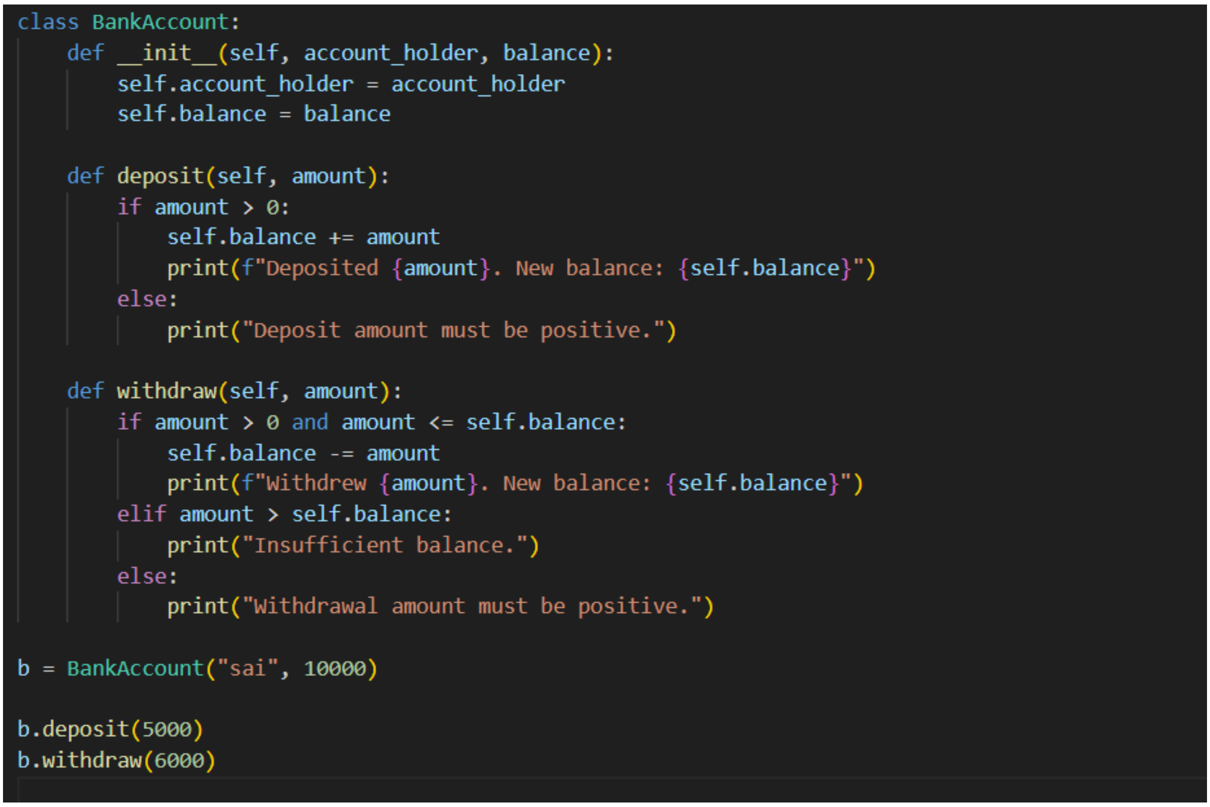
Task #3:

Create a class called BankAccount with attributes account_holder and balance. Use Copilot to complete methods for deposit(), withdraw(), and check for insufficient balance.

Prompt:

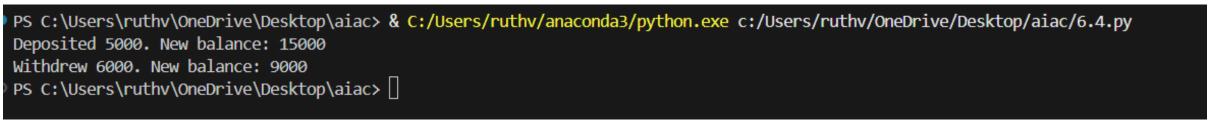
Write deposit() and withdraw() methods for this class.

Code:



```
class BankAccount:
    def __init__(self, account_holder, balance):
        self.account_holder = account_holder
        self.balance = 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 and amount <= self.balance:
            self.balance -= amount
            print(f"Withdrew {amount}. New balance: {self.balance}")
        elif amount > self.balance:
            print("Insufficient balance.")
        else:
            print("Withdrawal amount must be positive.")

b = BankAccount("sai", 10000)

b.deposit(5000)
b.withdraw(6000)
```

Output:



```
PS C:\Users\ruthv\OneDrive\Desktop\aiac> & C:/Users/ruthv/anaconda3/python.exe c:/Users/ruthv/OneDrive/Desktop/aiac/6.4.py
Deposited 5000. New balance: 15000
Withdrew 6000. New balance: 9000
PS C:\Users\ruthv\OneDrive\Desktop\aiac> 
```

Obsevation:

Copliet instantly generated the desired methods deposit() and withdraw() in the BankAccount().

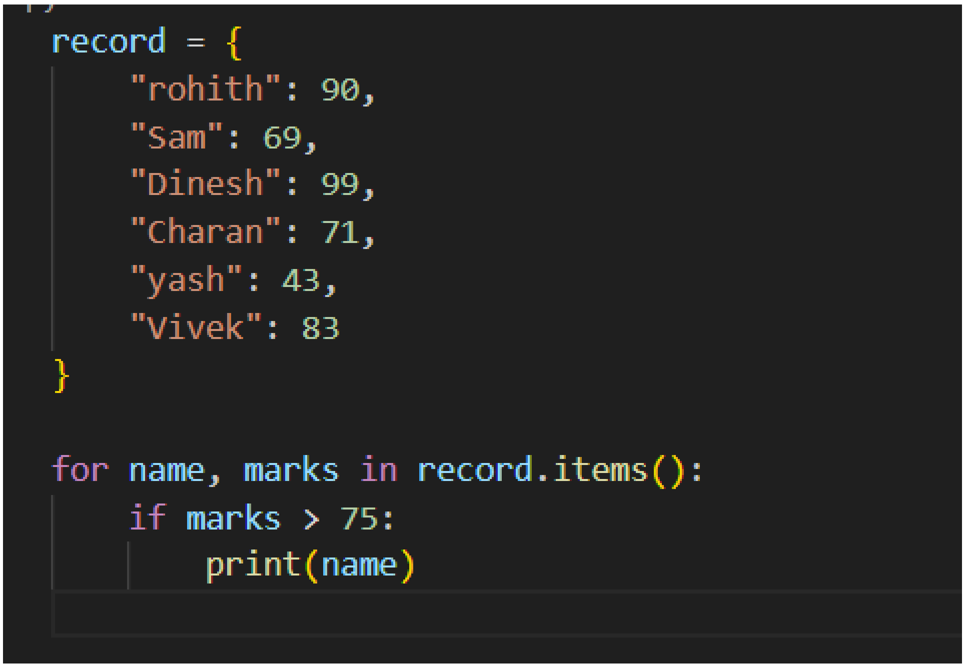
Task #4:

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.

Prompt:

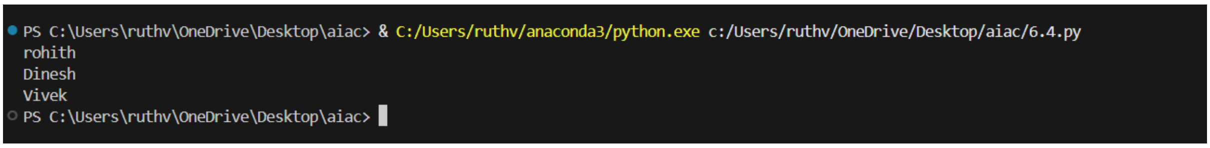
generate a loop to print the names of the students who scored more than 75 marks from the above dictionary.

Code:



```
record = {
    "rohith": 90,
    "Sam": 69,
    "Dinesh": 99,
    "Charan": 71,
    "yash": 43,
    "Vivek": 83
}

for name, marks in record.items():
    if marks > 75:
        print(name)
```

Output:



```
PS C:\Users\ruthv\OneDrive\Desktop\aiac> & C:/Users/ruthv/anaconda3/python.exe c:/Users/ruthv/OneDrive/Desktop/aiac/6.4.py
rohith
Dinesh
Vivek
PS C:\Users\ruthv\OneDrive\Desktop\aiac> 
```

Observation:

Copliet understood the assignment and created a loop that considers both keys and values of the dictionary and prints the names of the students who scored more than 75.

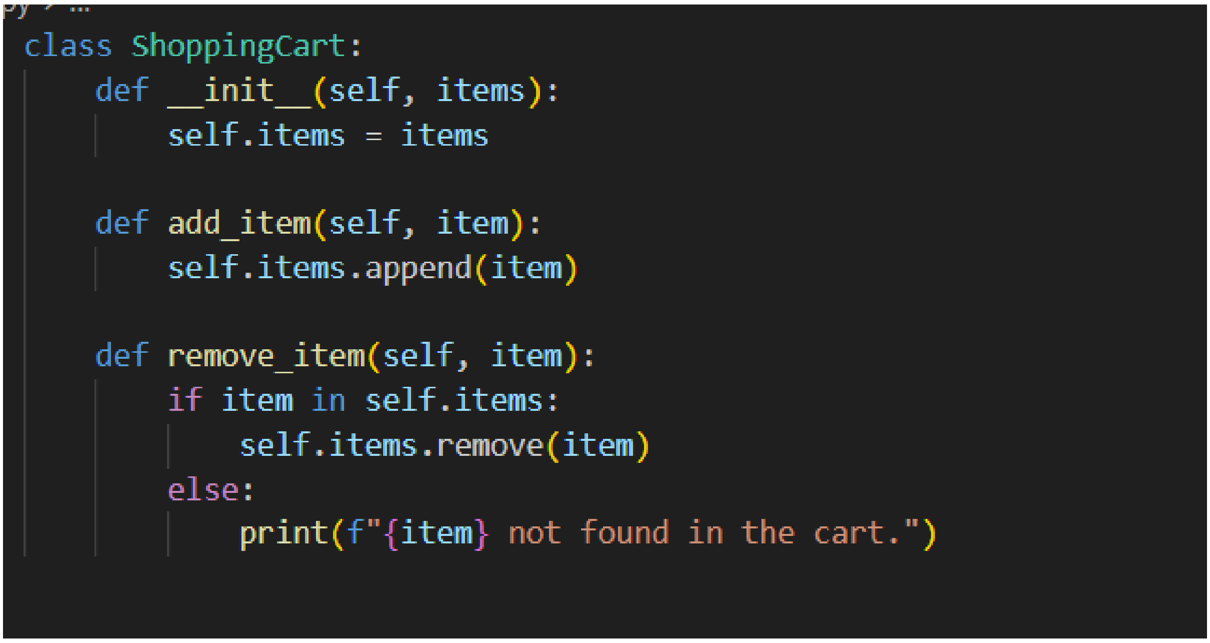
Task #5:

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.

Prompt:

Add methods to the class, add_item and remove_item.

Code:



```
py > ...
class ShoppingCart:
    def __init__(self, items):
        self.items = items

    def add_item(self, item):
        self.items.append(item)

    def remove_item(self, item):
        if item in self.items:
            self.items.remove(item)
        else:
            print(f"{item} not found in the cart.")
```

Observation:

Copilet successfully added the required methods.