

AI ASSISTED CODING

ASSIGNMENT-6.4

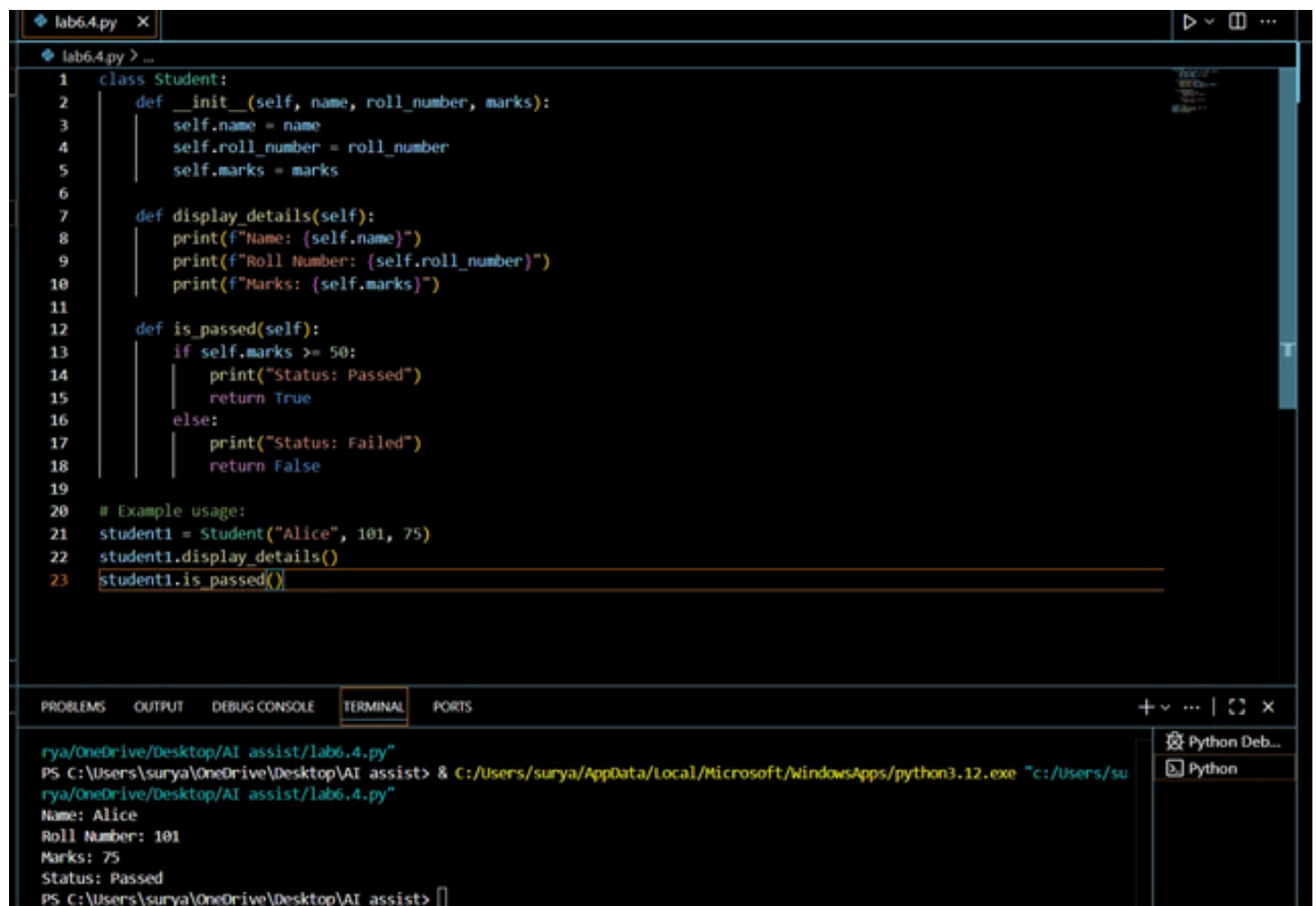
Name : G.OMKAR

HT.no : 2403A52039

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.

Code:



```
lab6.4.py x
lab6.4.py > ...
1 class Student:
2     def __init__(self, name, roll_number, marks):
3         self.name = name
4         self.roll_number = roll_number
5         self.marks = marks
6
7     def display_details(self):
8         print(f"Name: {self.name}")
9         print(f"Roll Number: {self.roll_number}")
10        print(f"Marks: {self.marks}")
11
12    def is_passed(self):
13        if self.marks >= 50:
14            print("Status: Passed")
15            return True
16        else:
17            print("Status: Failed")
18            return False
19
20 # Example usage:
21 student1 = Student("Alice", 101, 75)
22 student1.display_details()
23 student1.is_passed()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
rya/OneDrive/Desktop/AI assist/lab6.4.py"
PS C:\Users\surya\OneDrive\Desktop\AI assist> & C:/Users/surya/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/su
rya/OneDrive/Desktop/AI assist/lab6.4.py"
Name: Alice
Roll Number: 101
Marks: 75
Status: Passed
PS C:\Users\surya\OneDrive\Desktop\AI assist> |
```

Python Deb...
Python

Observation:

Copilot-generated methods like displaying details and checking if marks are above average make the class practical and easy to use. The code is straightforward and demonstrates basic object-oriented programming and conditional checks.

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.



The screenshot shows a Visual Studio Code editor window with a file named `lab6.4.py`. The code in the editor is as follows:

```
1 numbers = [1, 2, 3, 4, 5, 6]
2 for number in numbers:
3     # Copilot: Calculate and print the square of even numbers only
4     if number % 2 == 0:
5         print(number ** 2)
```

Below the editor, the `TERMINAL` tab is active, showing the command prompt output:

```
Marks: 75
Status: Passed
PS C:\Users\surya\OneDrive\Desktop\AI assist> & C:/Users/surya/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/surya/OneDrive/Desktop/AI assist/lab6.4.py"
4
16
36
PS C:\Users\surya\OneDrive\Desktop\AI assist>
```

On the right side of the terminal, there is a sidebar with icons for `Python Debug...` and `Python`.

Observation:

This task demonstrates how to use a for loop to iterate through a list of numbers and apply conditional logic to process only even numbers. By including a comment prompt, Copilot can suggest code to calculate and print the square of even numbers, making the code concise and easy to understand.

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

```
lab6.4.py > ...
1 class BankAccount:
2     def __init__(self, account_holder, balance=0):
3         self.account_holder = account_holder
4         self.balance = balance
5
6     def deposit(self, amount):
7         if amount > 0:
8             self.balance += amount
9             print(f"Deposited: {amount}. New balance: {self.balance}")
10        else:
11            print("Deposit amount must be positive.")
12
13    def withdraw(self, amount):
14        if amount <= 0:
15            print("Withdrawal amount must be positive.")
16        elif amount > self.balance:
17            print("Insufficient balance. Withdrawal denied.")
18        else:
19            self.balance -= amount
20            print(f"Withdrew: {amount}. New balance: {self.balance}")
21
22    # Example usage:
23    acc = BankAccount("John Doe", 100)
24    acc.deposit(50)
25    acc.withdraw(200)
26    acc.withdraw(30)
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

```
PS C:\Users\surya\OneDrive\Desktop\AI assist> & C:/Users/surya/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/su
rya/OneDrive/Desktop/AI assist/lab6.4.py"
PS C:\Users\surya\OneDrive\Desktop\AI assist> & C:/Users/surya/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/su
rya/OneDrive/Desktop/AI assist/lab6.4.py"
Deposited: 50. New balance: 150
Insufficient balance. Withdrawal denied.
Withdrew: 30. New balance: 120
PS C:\Users\surya\OneDrive\Desktop\AI assist>
```

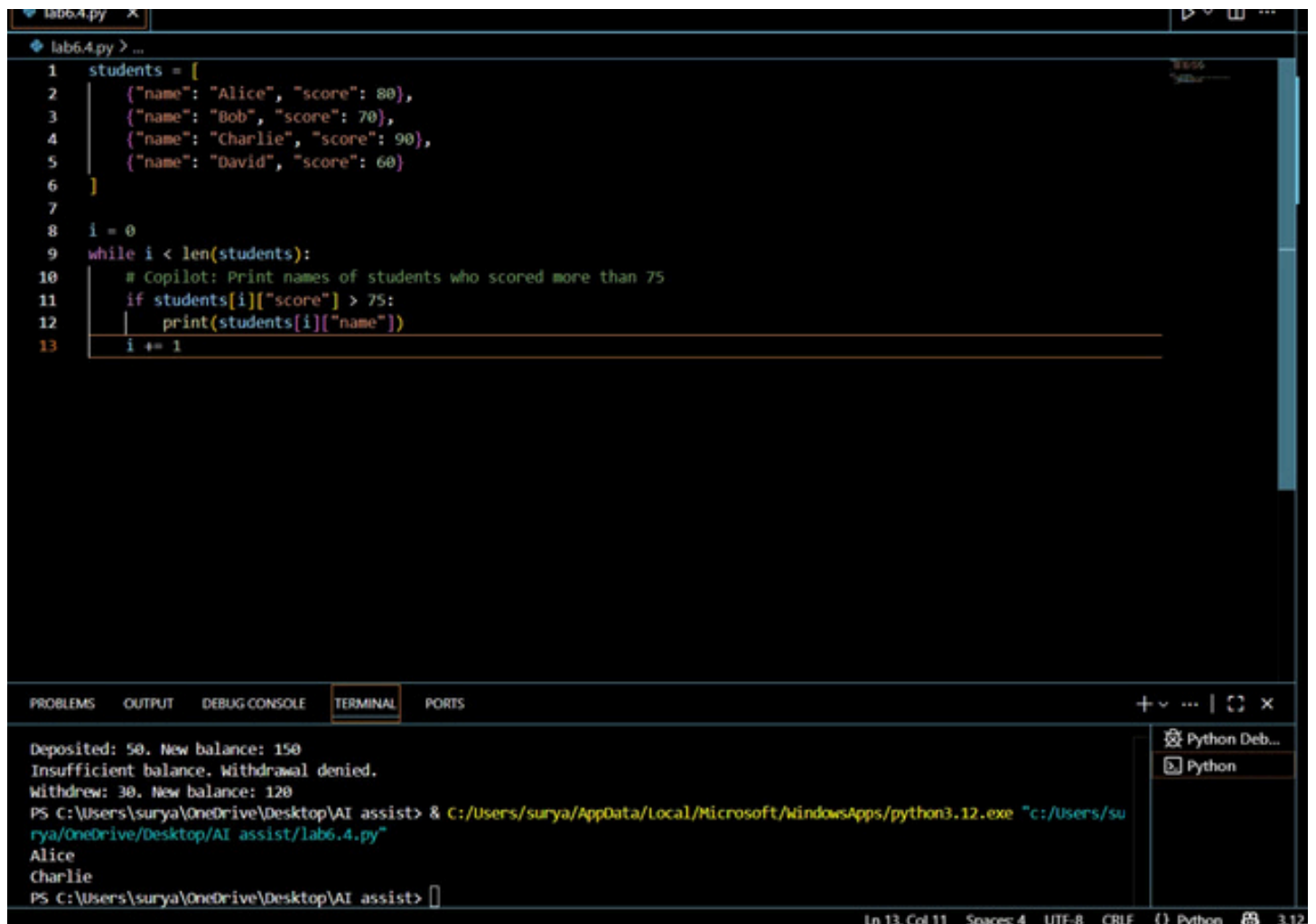
Ln 26, Col 17 Spaces: 4 UTF-8 CRLF Python 3.12.10

Observation:

The BankAccount class lets you deposit and withdraw money, and checks for insufficient balance before withdrawing. The code is simple and easy to understand.

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



The image shows a Visual Studio Code editor window with a Python file named `lab6.4.py`. The code defines a list of student dictionaries and a while loop that prints the names of students with scores greater than 75. Below the editor, the `TERMINAL` tab is active, showing the output of the script. The output indicates that Alice and Charlie are the students who scored more than 75. The status bar at the bottom shows the current line and column as `Ln 13, Col 11`.

```
1 students = [  
2     {"name": "Alice", "score": 80},  
3     {"name": "Bob", "score": 70},  
4     {"name": "Charlie", "score": 90},  
5     {"name": "David", "score": 60}  
6 ]  
7  
8 i = 0  
9 while i < len(students):  
10     # Copilot: Print names of students who scored more than 75  
11     if students[i]["score"] > 75:  
12         print(students[i]["name"])  
13     i += 1
```

Deposited: 50. New balance: 150
Insufficient balance. Withdrawal denied.
Withdrew: 30. New balance: 120
PS C:\Users\surya\OneDrive\Desktop\AI assist> & C:/Users/surya/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/surya/OneDrive/Desktop/AI assist/lab6.4.py"
Alice
Charlie
PS C:\Users\surya\OneDrive\Desktop\AI assist>

Ln 13, Col 11 Spaces: 4 UTF-8 CRLF Python 3.12

Observation:

A list of student dictionaries stores each student's name and score. The while loop checks each student and prints the names of those who scored more than 75. The code is simple and uses basic list and loop concepts.

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.

```
lab6.4.py > ...
1 class ShoppingCart:
2     def __init__(self):
3         self.items = []
4
5     def add_item(self, name, price):
6         self.items.append((name, price))
7
8     def remove_item(self, name):
9         self.items = [item for item in self.items if item[0] != name]
10
11    def total_bill(self):
12        total = 0
13        for name, price in self.items:
14            if price > 100:
15                price *= 0.9
16            total += price
17        print(total)
18
19    # Example usage:
20    cart = ShoppingCart()
21    cart.add_item("Shoes", 120)
22    cart.add_item("T-shirt", 80)
23    cart.add_item("Bag", 150)
24    cart.remove_item("T-shirt")
25    cart.total_bill()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Charlie
PS C:\Users\surya\OneDrive\Desktop\AI assist> & C:/Users/surya/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Use
rs/surya/OneDrive/Desktop/AI assist/lab6.4.py"
Total bill: 243.0
PS C:\Users\surya\OneDrive\Desktop\AI assist> & C:/Users/surya/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Use
rs/surya/OneDrive/Desktop/AI assist/lab6.4.py"
243.0
PS C:\Users\surya\OneDrive\Desktop\AI assist>

Python Deb...
Python

Ln 25, Col 18: Spaces 4 LITE-8 CRLF Python

Observation:

The ShoppingCart class starts with an empty items list. It has methods to add and remove items, and uses a loop to calculate the total bill. If an item costs more than 100, a discount is applied. The code is simple and shows basic class and loop usage.