

SR UNIVERSITY

AI ASSIST CODING

LAB 6.4: AI-Based Code Completion – Classes, Loops, and Conditionals

NAME

:SHAIK FAHEEM

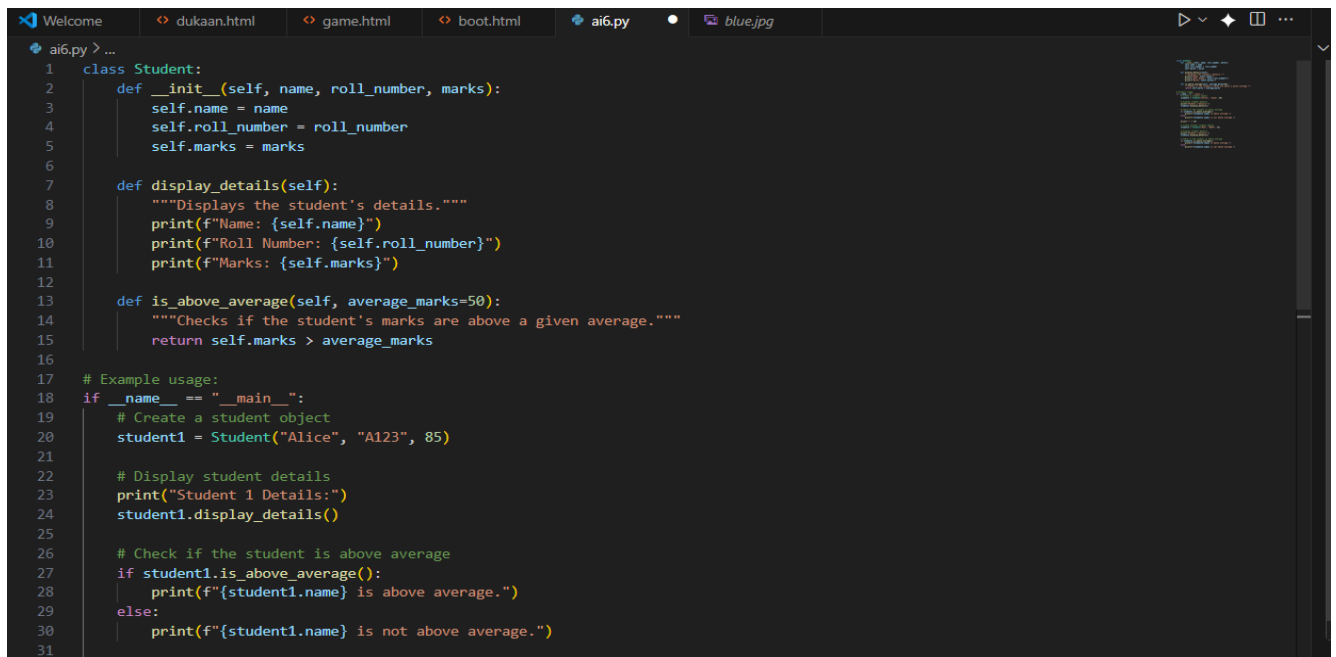
HTNO:2503A51L39

TASK #1:

Prompt:

- 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 Generated:

A screenshot of a code editor window with a dark theme. The editor shows a Python file named 'ai6.py' with the following code:

```
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         """Displays the student's details."""
9         print(f"Name: {self.name}")
10        print(f"Roll Number: {self.roll_number}")
11        print(f"Marks: {self.marks}")
12
13    def is_above_average(self, average_marks=50):
14        """Checks if the student's marks are above a given average."""
15        return self.marks > average_marks
16
17 # Example usage:
18 if __name__ == "__main__":
19     # Create a student object
20     student1 = Student("Alice", "A123", 85)
21
22     # Display student details
23     print("Student 1 Details:")
24     student1.display_details()
25
26     # Check if the student is above average
27     if student1.is_above_average():
28         print(f"{student1.name} is above average.")
29     else:
30         print(f"{student1.name} is not above average.")
31
```

The editor has tabs for 'Welcome', 'dukaan.html', 'game.html', 'boot.html', 'ai6.py', and 'blue.jpg'. The 'ai6.py' tab is active, showing the code. The code defines a 'Student' class with an 'init' method and two other methods: 'display_details' and 'is_above_average'. It also includes a main block that creates a 'Student' object and calls these methods.

Output After executing Code:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
em/OneDrive/Desktop/intezam/ai6.py"
• Student 1 Details:
Name: Alice
Roll Number: A123
Marks: 85
Alice is above average.
-----
Student 2 Details:
Name: Bob
Roll Number: B456
Marks: 45
Bob is not above average.
PS C:\Users\Shaik Faheem\OneDrive\Desktop\intezam>
```

Observations:

- The **Employee class** stores an employee's name, id, and monthly salary.
- It has a method to calculate the **yearly salary** (monthly \times 12).

TASK #2:

Prompt:

- 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.

Code Generated:

```
ai6.py > ...
1 numbers = [1, 2, 3, 4, 5, 6] # list of numbers
2 for num in numbers:
3     # TODO: calculate and print the square of even numbers only
4     if num % 2 == 0:
5         print(num ** 2)
6
```

Output After executing Code:

```
em/OneDrive/Desktop/intezam/ai6.py"
4
16
36
PS C:\Users\Shaik Faheem\OneDrive\Desktop\intezam>
```

Observations:

- The function iterates through numbers.
- We have to give the Condition if $\text{num} \% 2 == 0$ checks even numbers.
- It results in Prints their square using $\text{num} ** 2$.

TASK#3:

PROMPT:

- Create a class called Bank Account with attributes accountholder and balance .Use Copilot to complete methods for deposit() ,withdraw() ,and check for insufficient balance.

Code Generated:

```
ai6.py > BankAccount
1 class BankAccount:
2     def __init__(self, accountholder, balance=0):
3         self.accountholder = accountholder
4         self.balance = balance
5
6     def deposit(self, amount):
7         self.balance += amount
8         print(f"Deposited {amount}. New balance is {self.balance}.")
9
10    def withdraw(self, amount):
11        if self.balance >= amount:
12            self.balance -= amount
13            print(f"Withdrew {amount}. New balance is {self.balance}.")
14        else:
15            print("Insufficient balance. Withdrawal denied.")
16
17    def get_balance(self):
18        print(f"Current balance: {self.balance}")
19        return self.balance
20
21
22 # Example usage
23 account = BankAccount("Alice", 1000)
24 account.deposit(500)
25 account.withdraw(200)
26 account.withdraw(2000) # This will trigger insufficient balance
27 account.get_balance()
28
```

Output After executing Code:

```
em/OneDrive/Desktop/intezam/ai6.py"
Deposited 500. New balance is 1500.
Withdrew 200. New balance is 1300.
Insufficient balance. Withdrawal denied.
Current balance: 1300
PS C:\Users\Shaik_Faheem\OneDrive\Desktop\intezam>
```

Observations:

- We used function deposit(): increases balance.
- we can able to use the function withdraw(): prevents overdrawing using if conditions . its results in check_balance(): shows current balance.

TASK#4:

PROMPT:

- 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.

Code Generated:

```
ai6.py > ...
1  students = [
2      {"name": "Alice", "score": 82},
3      {"name": "Bob", "score": 67},
4      {"name": "Charlie", "score": 91},
5      {"name": "David", "score": 72}
6  ]
7
8  i = 0
9  while i < len(students):
10     if students[i]["score"] > 75:
11         print(students[i]["name"])
12     i += 1
13
```

Output After executing Code:

```
PS C:\Users\Shaik Faheem\OneDrive\Desktop\intezam> & "C:\Users\Shaik Faheem\OneDrive\Desktop\intezam\ai6.py"
Alice
Charlie
```

Observations:

- We Uses while loop with counter i.
- The loop Checks if score > 75.
- It will Prints qualifying students.

TASK#5:

PROMPT:

- Begin writing a class Shopping Cart 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

.Code Generated:

```
Welcome | dukaan.html | game.html | boot.html | ai6.py | X
ai6.py > ...
1 class ShoppingCart:
2     def __init__(self):
3         self.items = [] # Each item will be a dictionary: {"name": "Apple", "price": 50, "quantity": 2}
4
5     def add_item(self, name, price, quantity=1):
6         self.items.append({"name": name, "price": price, "quantity": quantity})
7         print(f"Added {quantity} x {name} to the cart.")
8
9     def remove_item(self, name):
10        for item in self.items:
11            if item["name"] == name:
12                self.items.remove(item)
13                print(f"Removed {name} from the cart.")
14            return
15        print(f"Item {name} not found in the cart.")
16
17    def calculate_total(self):
18        total = 0
19        for item in self.items:
20            total += item["price"] * item["quantity"]
21
22        # Apply conditional discounts
23        if total > 1000:
24            discount = 0.1 * total # 10% discount
25            total -= discount
26            print(f"10% discount applied: -{discount}")
27        elif total > 500:
28            discount = 0.05 * total # 5% discount
29            total -= discount
30            print(f"5% discount applied: -{discount}")
```

Output After executing Code:

```
em/OneDrive/Desktop/intezam/ai6.py"
Added 1 x Laptop to the cart.
Added 2 x Mouse to the cart.
Removed Mouse from the cart.
Added 1 x Headphones to the cart.
10% discount applied: -110.0
Total bill: 990.0
```

Observations:

- If we want to add item use function-add_item(): adds item to cart.
- If we want to remove item use function remove_item(): removes by name
-