AI ASSISTED CODING

ASSIGNMENT-6.3

M.KEERTHANA

2403A51259

BATCH-11

CSE-GEN

Task Description#1 (Classes)

- Use AI to complete a Student class with attributes and a method.
- Check output
- Analyze the code generated by AI tool

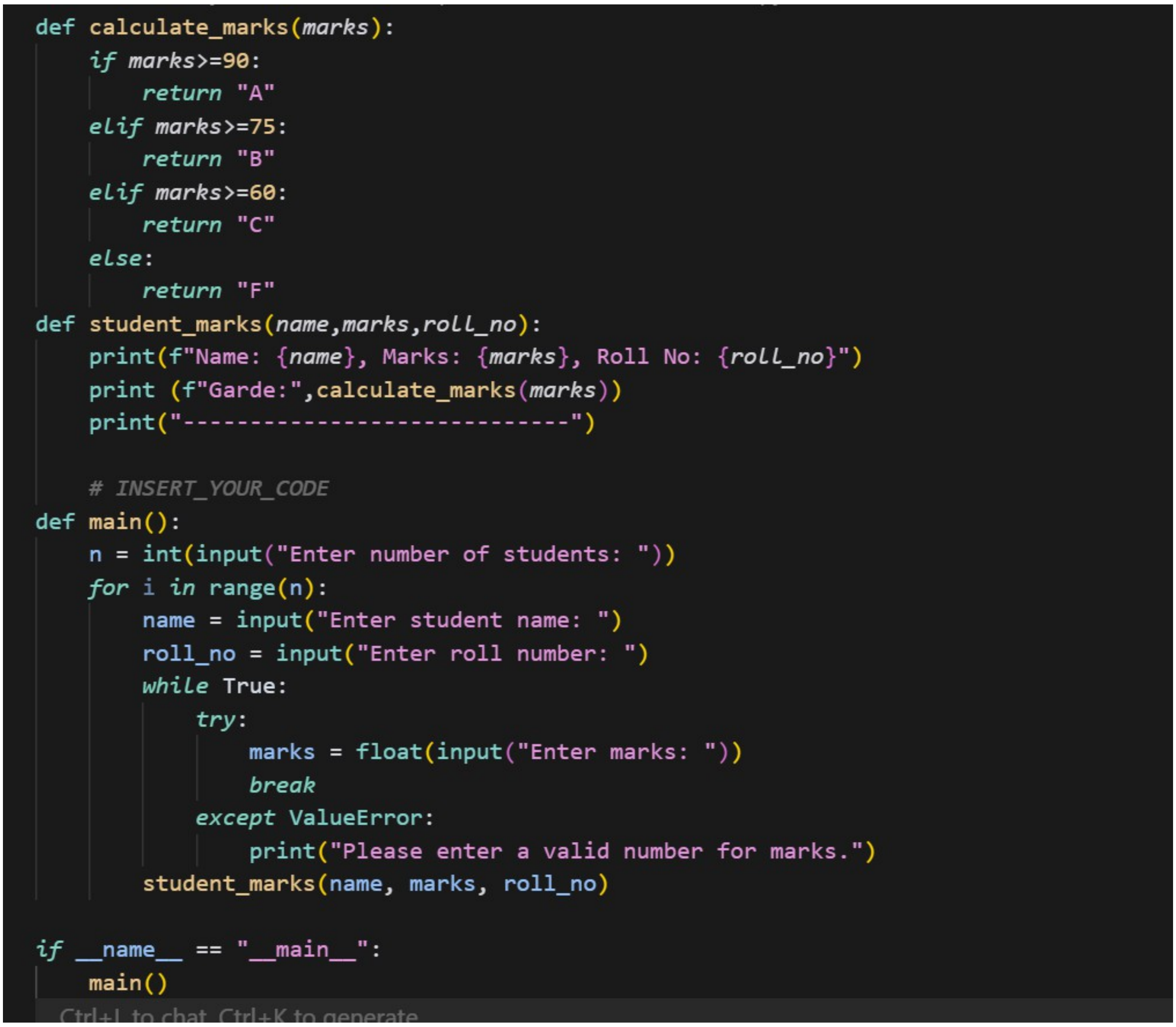
Instructions:

- **Initialize class with attributes like name, roll no, marks**
- **Method to display student details**
- **Method to calculate grade based on marks (A:>=90, B: >=75, C: >=60, else Fail)**

Start Writing code and auto complete using any AI tool

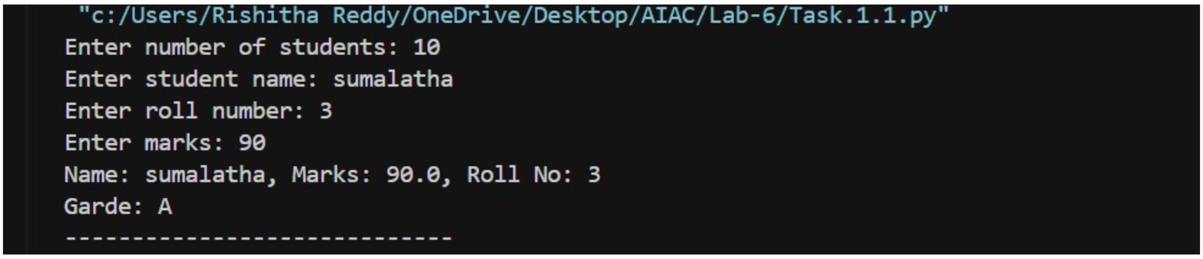
Expected Output#1

- Class with constructor and display_details() method



```python
def calculate_marks(marks):
    if marks>=90:
        return "A"
    elif marks>=75:
        return "B"
    elif marks>=60:
        return "C"
    else:
        return "F"
def student_marks(name,marks,roll_no):
    print(f"Name: {name}, Marks: {marks}, Roll No: {roll_no}")
    print (f"Garde:",calculate_marks(marks))
    print("-----------------------------")

    # INSERT_YOUR_CODE
def main():
    n = int(input("Enter number of students: "))
    for i in range(n):
        name = input("Enter student name: ")
        roll_no = input("Enter roll number: ")
        while True:
            try:
                marks = float(input("Enter marks: "))
                break
            except ValueError:
                print("Please enter a valid number for marks.")
        student_marks(name, marks, roll_no)

if __name__ == "__main__":
    main()
```

OUTPUT:



```
  "c:/Users/Rishitha Reddy/OneDrive/Desktop/AIAC/Lab-6/Task.1.1.py"
Enter number of students: 10
Enter student name: sumalatha
Enter roll number: 3
Enter marks: 90
Name: sumalatha, Marks: 90.0, Roll No: 3
Garde: A
-----------------------------
```

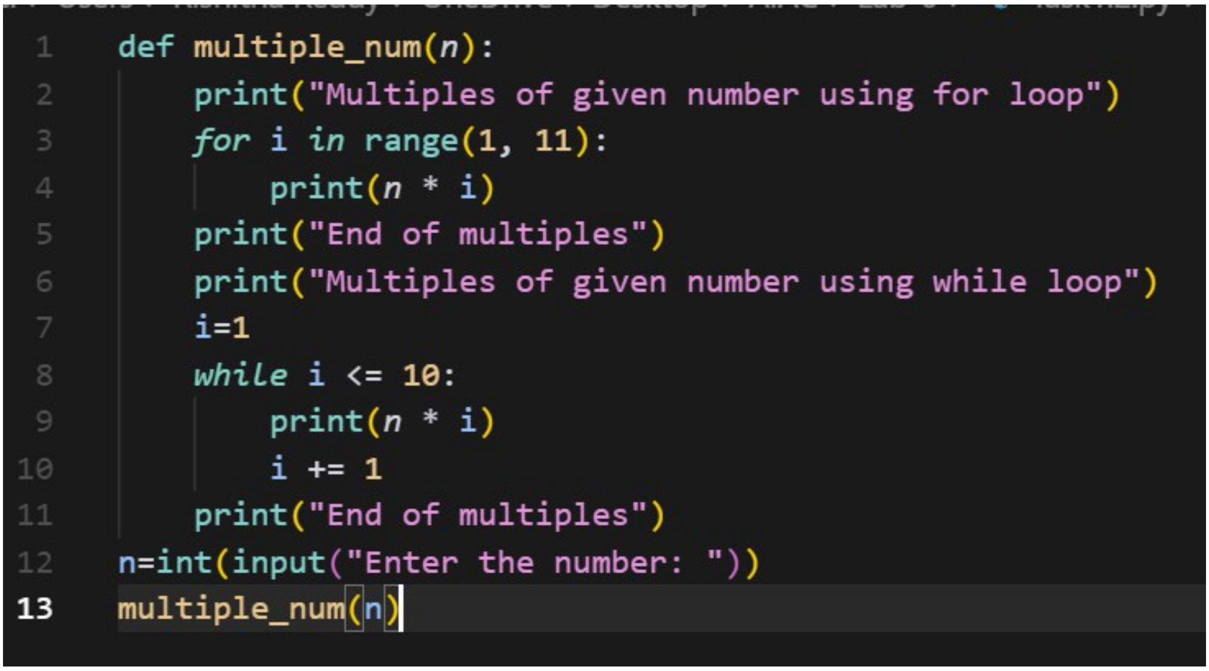
Task Description#2 (Loops)

- Prompt AI to complete a function that prints the first 10 multiples of a number using a loop.
- Analyze the generated code
- Ask AI to generate code using other controlled looping

Write code using **For** Loop, later complete code using **While** Loop

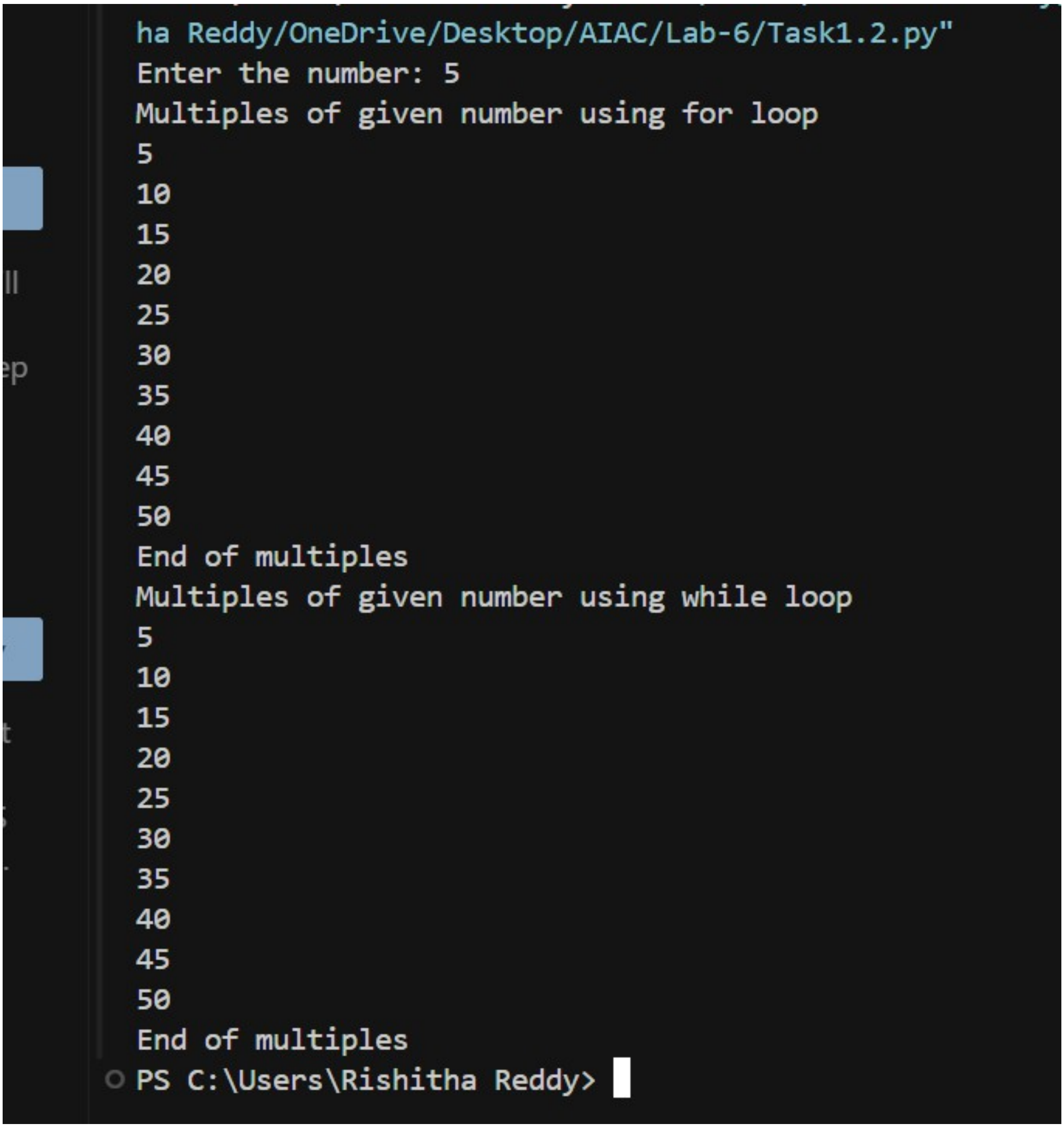
Expected Output#2

- Correct loop-based implementation



```python
def multiple_num(n):
    print("Multiples of given number using for loop")
    for i in range(1, 11):
        print(n * i)
    print("End of multiples")
    print("Multiples of given number using while loop")
    i=1
    while i <= 10:
        print(n * i)
        i += 1
    print("End of multiples")
n=int(input("Enter the number: "))
multiple_num(n)
```

OUTPUT:



```
ha Reddy/OneDrive/Desktop/AIAC/Lab-6/Task1.2.py"
Enter the number: 5
Multiples of given number using for loop
5
10
15
20
25
30
35
40
45
50
End of multiples
Multiples of given number using while loop
5
10
15
20
25
30
35
40
45
50
End of multiples
PS C:\Users\Rishitha Reddy>
```

Task Description#3 (Conditional Statements)

- Ask AI to write nested if-elif-else conditionals to classify age groups.
- Analyze the generated code
- Ask AI to generate code using other conditional statements

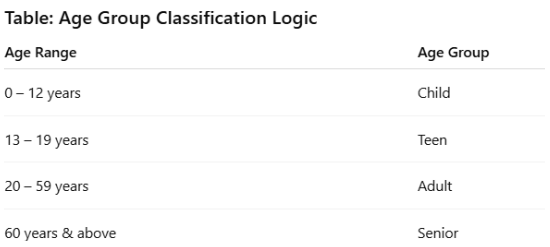
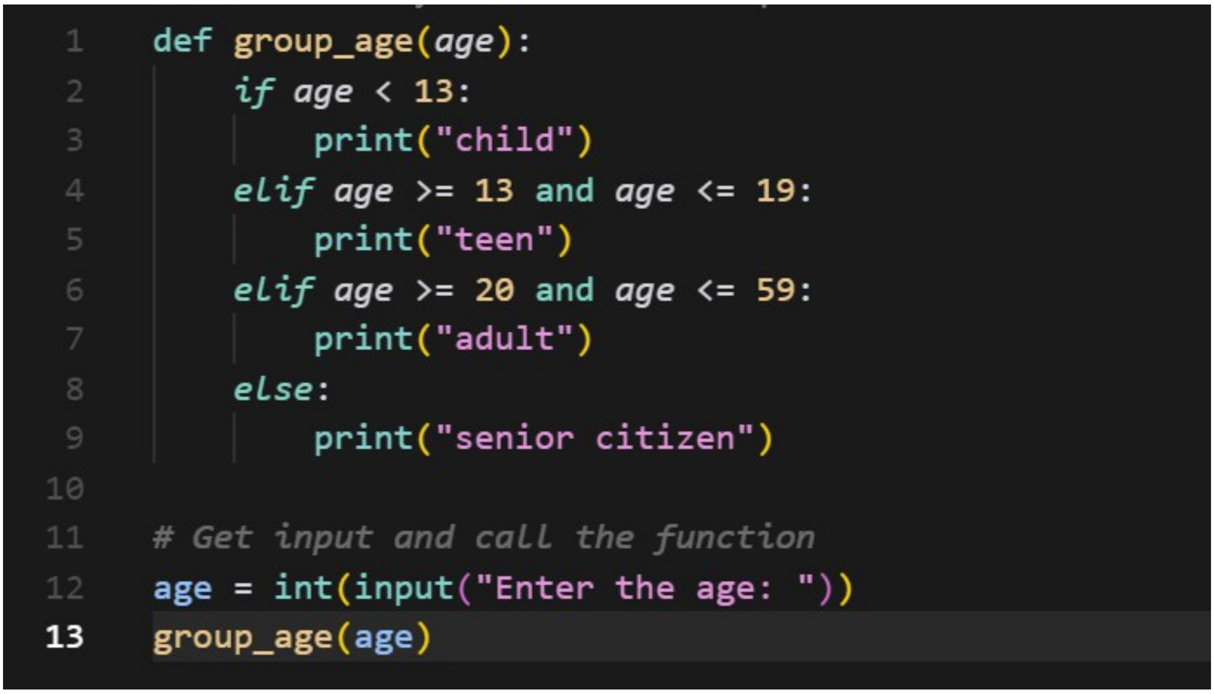


Table: Age Group Classification Logic

Age Range	Age Group
0 – 12 years	Child
13 – 19 years	Teen
20 – 59 years	Adult
60 years & above	Senior

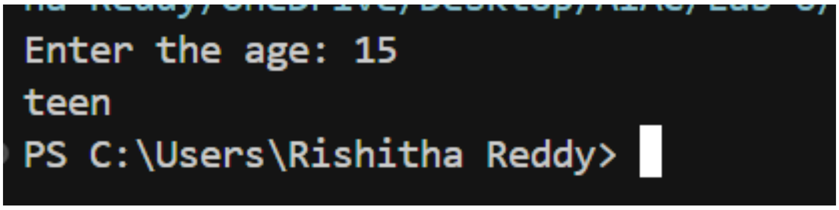
Expected Output#3

- Age classification function with appropriate conditions and with explanation



```python
def group_age(age):
    if age < 13:
        print("child")
    elif age >= 13 and age <= 19:
        print("teen")
    elif age >= 20 and age <= 59:
        print("adult")
    else:
        print("senior citizen")

# Get input and call the function
age = int(input("Enter the age: "))
group_age(age)
```

OUTPUT:



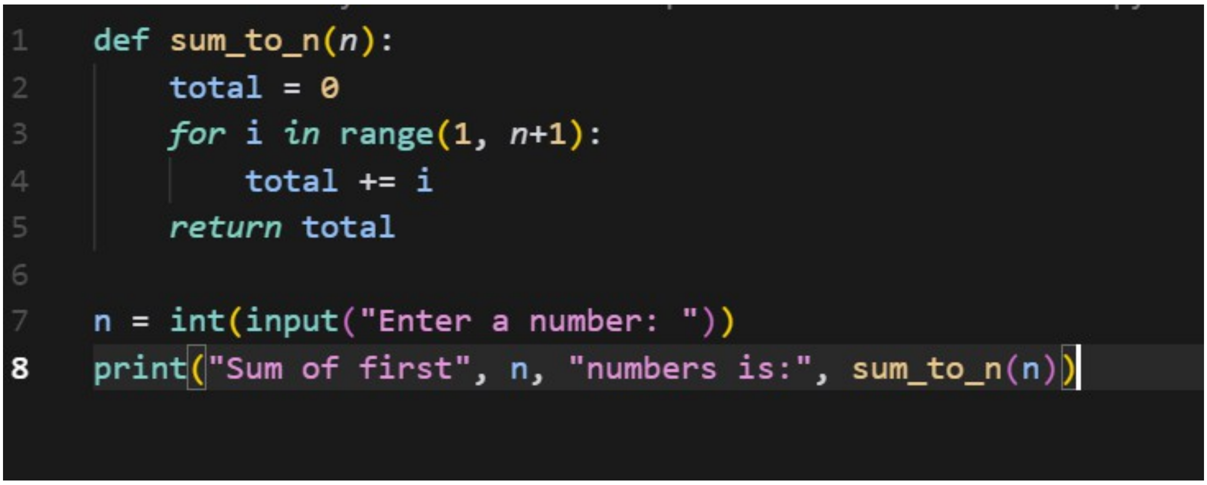
```
Enter the age: 15
teen
PS C:\Users\Rishitha Reddy>
```

Task Description#4 (For and While loops)

- Generate a sum_to_n() function to calculate sum of first n numbers
- Analyze the generated code
- Get suggestions from AI with other controlled looping

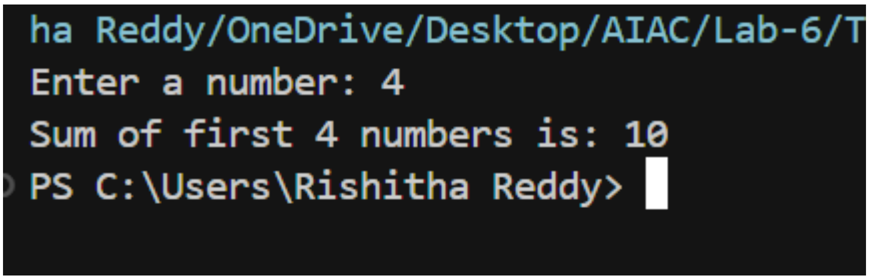
Expected Output#4

- Python code with explanation



```python
def sum_to_n(n):
    total = 0
    for i in range(1, n+1):
        total += i
    return total

n = int(input("Enter a number: "))
print("Sum of first", n, "numbers is:", sum_to_n(n))
```

OUTPUT:



```
ha Reddy/OneDrive/Desktop/AIAC/Lab-6/T
Enter a number: 4
Sum of first 4 numbers is: 10
PS C:\Users\Rishitha Reddy>
```

Task Description#5 (Class)

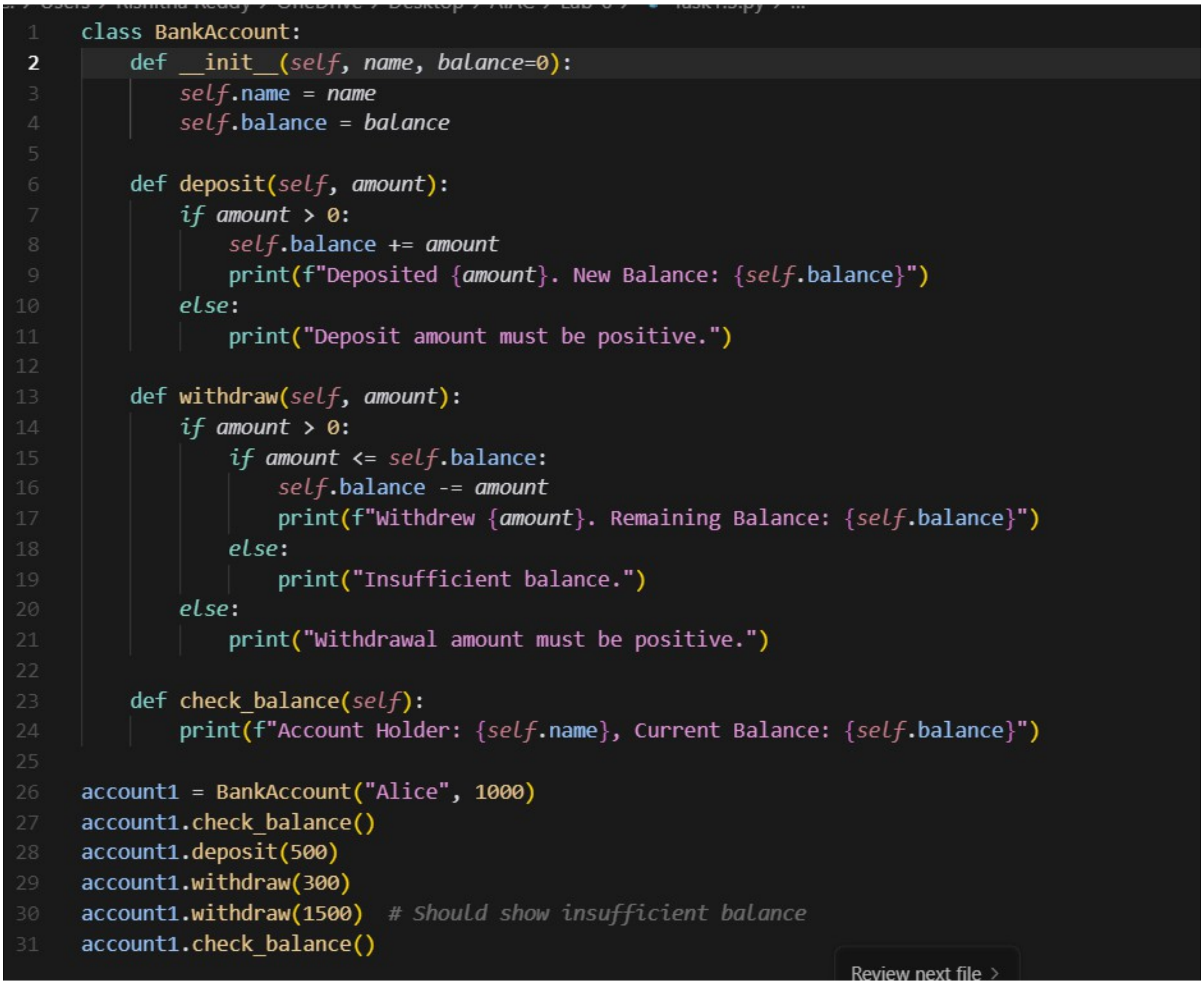
- Use AI to build a BankAccount class with deposit, withdraw, and balance methods.
- Analyze the generated code
- Add comments and explain code

Instructions

- Initialize BankAccount class with attributes like name, balance
- Method to deposit amount
- Method to withdraw amount
- Method to check balance

Expected Output#5

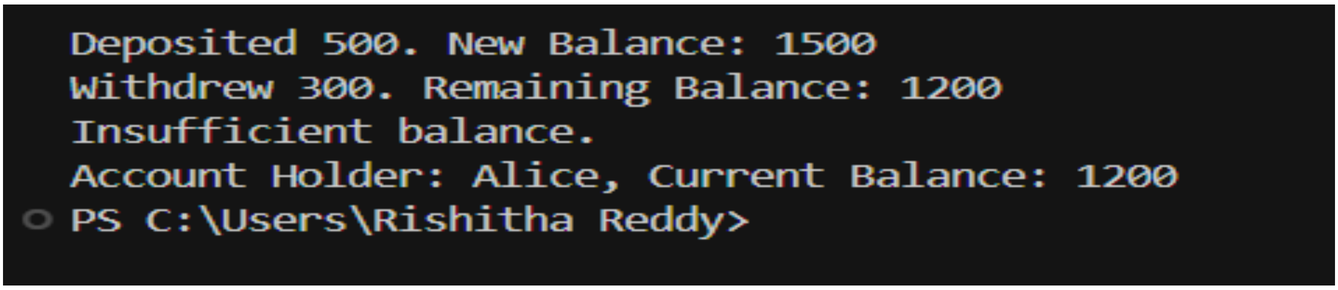
- Python code with explanation



```python
class BankAccount:
    def __init__(self, name, balance=0):
        self.name = name
        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:
            if amount <= self.balance:
                self.balance -= amount
                print(f"Withdrew {amount}. Remaining Balance: {self.balance}")
            else:
                print("Insufficient balance.")
        else:
            print("Withdrawal amount must be positive.")

    def check_balance(self):
        print(f"Account Holder: {self.name}, Current Balance: {self.balance}")

account1 = BankAccount("Alice", 1000)
account1.check_balance()
account1.deposit(500)
account1.withdraw(300)
account1.withdraw(1500)  # Should show insufficient balance
account1.check_balance()
```

-

OUTPUT:



```
Deposited 500. New Balance: 1500
Withdrew 300. Remaining Balance: 1200
Insufficient balance.
Account Holder: Alice, Current Balance: 1200
PS C:\Users\Rishitha Reddy>
```