SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE		DEPARTMENT OF COMPUTER SCIENCE ENGINEERING	
ProgramName:B. Tech		**Assignment Type: Lab**	**AcademicYear:**2025-2026
CourseCoordinatorName		Venkataramana Veeramsetty	
Instructor(s)Name		Dr. V. Venkataramana (Co-ordinator) Dr. T. Sampath Kumar Dr. Pramoda Patro Dr. Brij Kishor Tiwari Dr.J.Ravichander Dr. Mohammand Ali Shaik Dr. Anirodh Kumar Mr. S.Naresh Kumar Dr. RAJESH VELPULA Mr. Kundhan Kumar Ms. Ch.Rajitha Mr. M Prakash Mr. B.Raju Intern 1 (Dharma teja) Intern 2 (Sai Prasad) Intern 3 (Sowmya) NS_2 (Mounika)	
CourseCode	24CS002PC215	**CourseTitle**	AI Assisted Coding
Year/Sem	II/I	**Regulation**	R24
Date and Day of Assignment	Week3 - Wednesday	**Time(s)**	
Duration	2 Hours	**Applicableto Batches**	
AssignmentNumber:6.3(Present assignment number)/**24**(Total number of assignments)			

Q.No.	Question	*ExpectedTime to complete*
1	Lab 6: AI-Based Code Completion – Classes, Loops, and Conditionals **Lab Objectives:** • To explore AI-powered auto-completion features for core Python constructs. • To analyze how AI suggests logic for class definitions, loops, and conditionals. • To evaluate the completeness and correctness of code generated by AI assistants. **Lab Outcomes (LOs):**	Week3 - Wednesday

After completing this lab, students will be able to:

- Use AI tools to generate and complete class definitions and methods.
- Understand and assess AI-suggested loops for iterative tasks.
- Generate conditional statements through prompt-driven suggestions.
- Critically evaluate AI-assisted code for correctness and clarity.

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

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

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

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

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

Note: Report should be submitted a word document for all tasks in a single document with prompts, comments & code explanation, and output and if required, screenshots

Evaluation Criteria:

Criteria	Max Marks
Class	1.0
Loops	1.0
Conditional Statements	0.5
Total	**2.5 Marks**

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

task-6.1.py > ...

```python
    def calculate_marks(marks):
        else:
            return "fail"
    def student(name,rollno,marks):
        print("----Student Marks----")
        print("Name:", name)
        print("Roll No:", rollno)
        print("Marks:", marks)
        grade = calculate_marks(marks)
        print("Grade:", grade)
        if grade == "fail":
            print("Better luck next time!")
            print("You can do it!")
        else:
            print("Congratulations! You have passed.")
            print("Keep up the good work!")
    name=input("Enter your name: ")
    rollno=input("Enter your roll number: ")
    marks=float(input("Enter your marks: "))
    student(name,rollno,marks)
```

```
Enter your name: swadha reddy
Enter your roll number: 1001
Enter your marks: 97
----Student Marks----
Name: swadha reddy
Roll No: 1001
Marks: 97.0
Grade: A
Congratulations! You have passed.
Keep up the good work!
PS C:\Users\solle\OneDrive\Desktop\AIAC\lab-06> ^C
PS C:\Users\solle\OneDrive\Desktop\AIAC\lab-06>
```

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

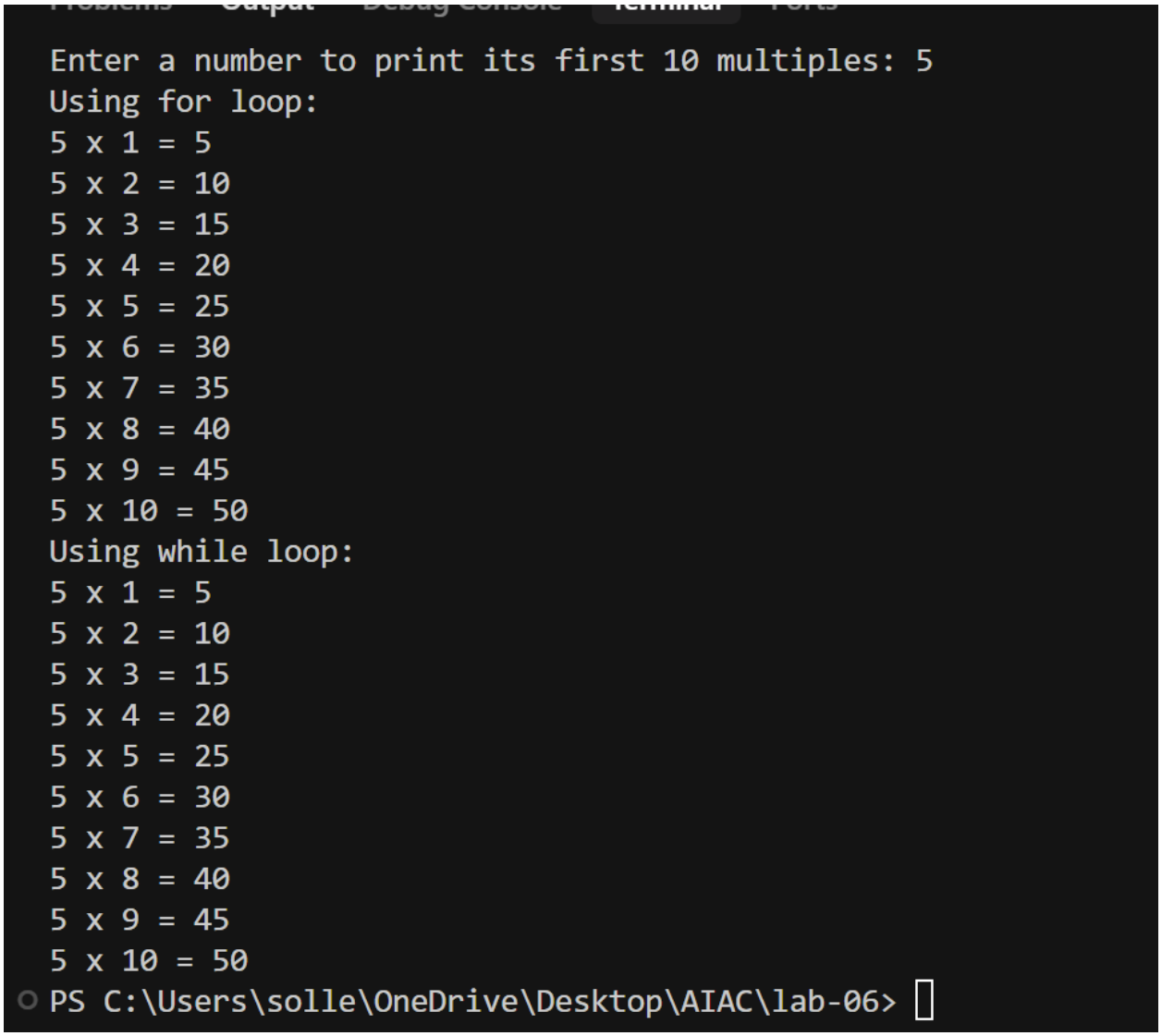
Expected Output#2

- Correct loop-based implementation

task-6.2.py > ...

```python
def print_multiples():
    num = int(input("Enter a number to print its first 10 multiples: "))
    print("Using for loop:")
    for i in range(1, 11):
        print(f"{num} x {i} = {num * i}")
    print("Using while loop:")
    i = 1
    while i <= 10:
        print(f"{num} x {i} = {num * i}")
        i += 1

print_multiples()
```



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

task-6.3.py > ...

```python
def classify_age(age):
    if age < 0:
        print("Invalid age")
    elif age <= 12:
        print("Child")
    elif age <= 19:
        print("Teen")
    elif age <= 59:
        print("Adult")
    else:
        print("Senior")

# Get age input from the user
try:
    user_age = int(input("Enter your age: "))
    classify_age(user_age)
except ValueError:
    print("Please enter a valid number.")
```

```
313/python.exe c:/Users/solle/OneDrive/Desktop/AIA
Enter your age: 18
Teen
PS C:\Users\solle\OneDrive\Desktop\AIAC\lab-06>
```

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

task-6.4.py > ...

```python
def sum_to_n(num):
        total += i
    return total
def print_first_n_numbers(n):
    print("Using for loop:")
    for i in range(1, n + 1):
        print(i, end=' ')
    print("\nUsing while loop:")
    i = 1
    while i <= n:
        print(i, end=' ')
        i += 1
        print()

    # Example usage:
n = int(input("Enter n: "))
print_first_n_numbers(n)
print(f"Sum of first {n} numbers is: {sum_to_n(n)}")

```

```
PS C:\Users\solle\OneDrive\Desktop\AIAC\lab-06> & C:/Users/solle/AppData/Local/Programs/Pyt
313/python.exe c:/Users/solle/OneDrive/Desktop/AIAC/lab-06/task-6.4.py
Enter n: 10
Using for loop:
1 2 3 4 5 6 7 8 9 10
Using while loop:
1
2
3
4
5
6
7
8
9
10
Sum of first 10 numbers is: 55
PS C:\Users\solle\OneDrive\Desktop\AIAC\lab-06> ^C
PS C:\Users\solle\OneDrive\Desktop\AIAC\lab-06>
```

Task Description#5 (Class)

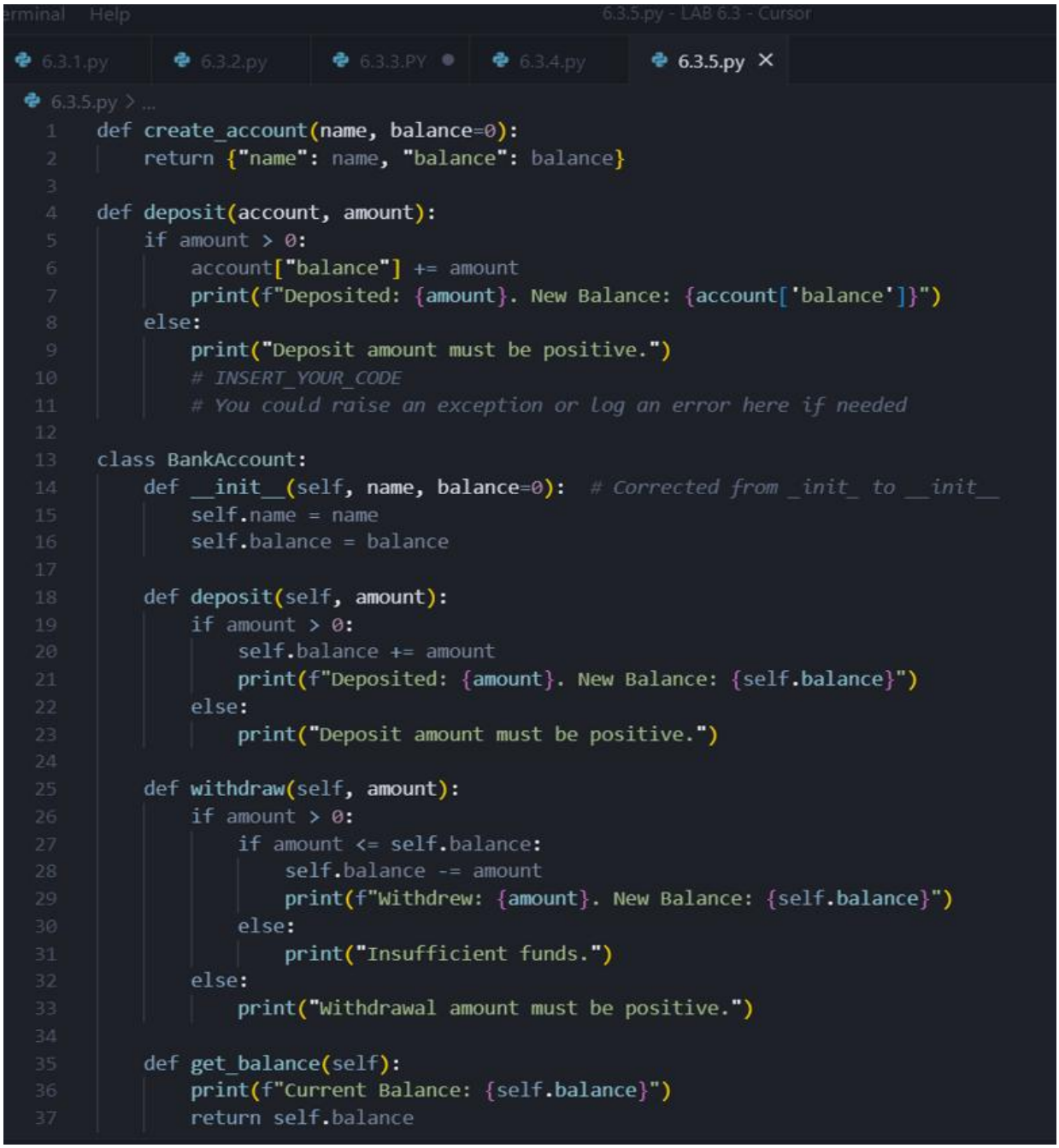
- Use AI to build a BankAccount class with deposit, withdraw, and balance methods.
- Analyze the generated code
- Add comments and explain code

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 explanat
-



Terminal Help 6.3.5.py - LAB 6.3 - Cursor

6.3.1.py 6.3.2.py 6.3.3.PY 6.3.4.py 6.3.5.py

6.3.5.py > ...

```
50      # Example usage:
51      while True:
52          print("\nOptions: 1) Deposit 2) Withdraw 3) Balance 4) Exit")
53          choice = input("Choose an option: ")
54          if choice == "1":
55              try:
56                  amt = float(input("Enter amount to deposit: "))
57                  account.deposit(amt)
58              except ValueError:
59                  print("Invalid amount.")
60          elif choice == "2":
61              try:
62                  amt = float(input("Enter amount to withdraw: "))
63                  account.withdraw(amt)
64              except ValueError:
65                  print("Invalid amount.")
66          elif choice == "3":
67              account.get_balance()
68          elif choice == "4":
69              print("Exiting.")
70              break
71          else:
72              print("Invalid option.")
```

Problems Output Debug Console Terminal Ports Python

```
PS C:\Users\ABHI\OneDrive\Documents\Desktop\AIAC\LAB 6.3> & C:/Users/ABHI/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/ABHI/OneDrive/Documents/Desktop/AIAC/LAB 6.3/6.3.5.py"
Enter account holder's name: Abhigna
Enter initial balance: 100000
Account created for Abhigna with balance 100000.0

Options: 1) Deposit 2) Withdraw 3) Balance 4) Exit
Choose an option: 4
Exiting.
PS C:\Users\ABHI\OneDrive\Documents\Desktop\AIAC\LAB 6.3>
```

Ctrl+K to generate a command