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

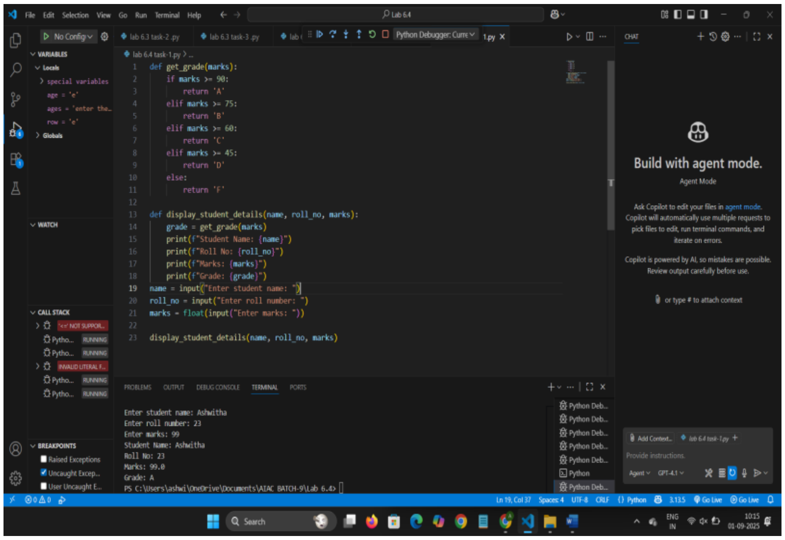
Instructions:

- **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



```python
def get_grade(marks):
    if marks >= 90:
        return 'A'
    elif marks >= 75:
        return 'B'
    elif marks >= 60:
        return 'C'
    elif marks >= 45:
        return 'D'
    else:
        return 'F'

def display_student_details(name, roll_no, marks):
    grade = get_grade(marks)
    print(f"Student Name: {name}")
    print(f"Roll No: {roll_no}")
    print(f"Marks: {marks}")
    print(f"Grade: {grade}")
name = input("Enter student name: ")
roll_no = input("Enter roll number: ")
marks = float(input("Enter marks: "))

display_student_details(name, roll_no, marks)
```

```
Enter student name: Ashwitha
Enter roll number: 23
Enter marks: 99
Student Name: Ashwitha
Roll No: 23
Marks: 99.0
Grade: A
PS C:\Users\ashwi\OneDrive\Documents\AIAC BATCH-9\Lab 6.4>
```

EXPLANATION:
IN THIS CODE WE CREATED THE CLASS FOR ENTERING THE ATTRIBUTES AND DISPLAYED THE ATTRIBUTES .

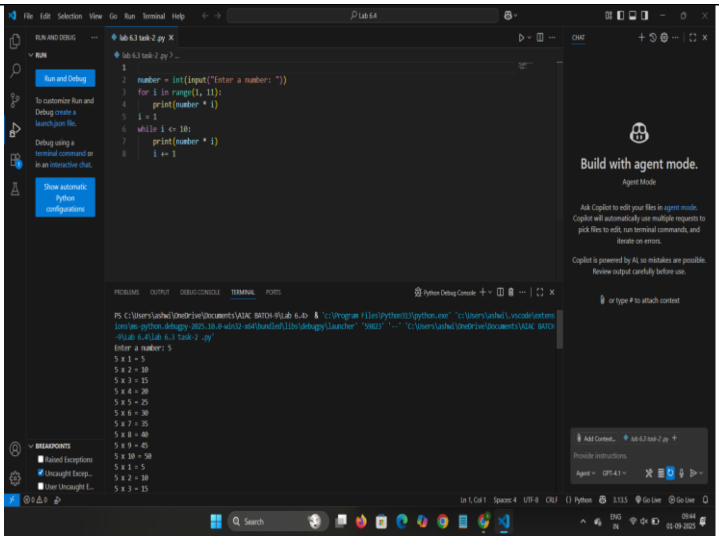
Task Description#2 (Loops)

- 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
number = int(input("Enter a number: "))
for i in range(1, 11):
    print(number * i)
i = 1
while i <= 10:
    print(number * i)
    i += 1
```

```
PS C:\Users\ashwi\OneDrive\Documents\AIAC BATCH-9\Lab 6.4> & 'c:\Program Files\Python313\python.exe' 'c:\Users\ashwi\.vscode\extensions\ms-python.debugpy-2025.10.0-win32-x64\bundled\libs\debugpy\launcher' '59823' '--' 'C:\Users\ashwi\OneDrive\Documents\AIAC BATCH-9\Lab 6.4\lab 6.3 task-2 .py'
Enter a number: 5
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
```

EXPLANATION:
IN THIS CODE WE USED FOR LOOP AND WHILE LOOP TO GENERATE THE MULTIPLES OF NUMBER.

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

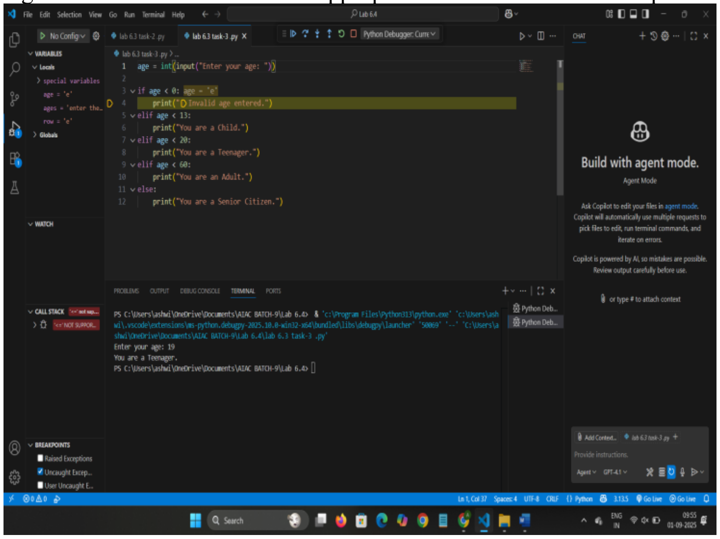
Table: Age Group Classification Logic

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
age = int(input("Enter your age: "))

if age < 0:
    print("Invalid age entered.")
elif age < 13:
    print("You are a Child.")
elif age < 20:
    print("You are a Teenager.")
elif age < 60:
    print("You are an Adult.")
else:
    print("You are a Senior Citizen.")
```

```
PS C:\Users\ashwi\OneDrive\Documents\AIAC BATCH-9\Lab 6.4> & 'c:\Program Files\Python313\python.exe' 'c:\Users\ashwi\.vscode\extensions\ms-python.debugpy-2025.10.0-win32-x64\bundled\libs\debugpy\launcher' '50069' '--' 'C:\Users\ashwi\OneDrive\Documents\AIAC BATCH-9\Lab 6.4\lab 6.3 task-3 .py'
Enter your age: 19
You are a Teenager.
PS C:\Users\ashwi\OneDrive\Documents\AIAC BATCH-9\Lab 6.4> 
```

EXPLANATION:

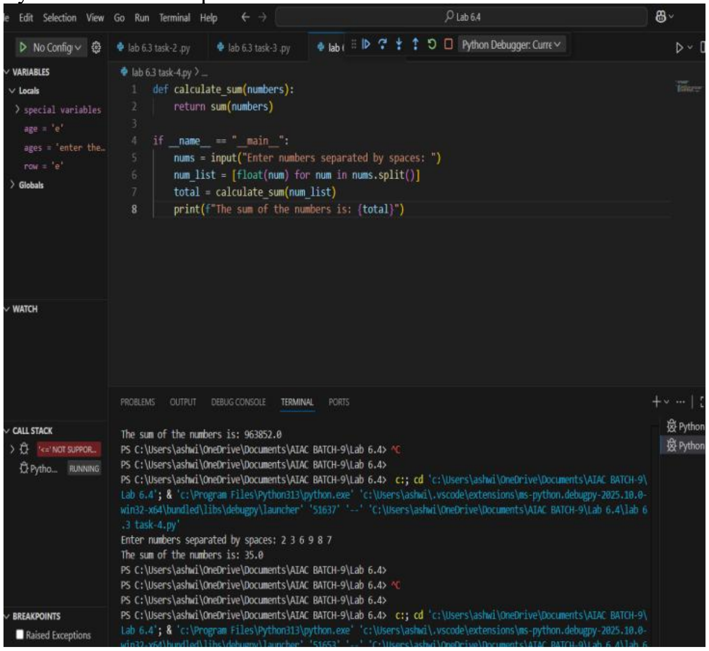
IN THIS CODE USING THE NESTED LOOP TO CLASSIFY THE AGE GROUP GIVEN BY USER.

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 calculate_sum(numbers):
    return sum(numbers)

if __name__ == "__main__":
    nums = input("Enter numbers separated by spaces: ")
    num_list = [float(num) for num in nums.split()]
    total = calculate_sum(num_list)
    print(f"The sum of the numbers is: {total}")
```

```
The sum of the numbers is: 963852.0
PS C:\Users\ashwi\OneDrive\Documents\AIAC BATCH-9\Lab 6.4> ^C
PS C:\Users\ashwi\OneDrive\Documents\AIAC BATCH-9\Lab 6.4>
PS C:\Users\ashwi\OneDrive\Documents\AIAC BATCH-9\Lab 6.4>  c:; cd 'c:\Users\ashwi\OneDrive\Documents\AIAC BATCH-9\
Lab 6.4'; & 'c:\Program Files\Python313\python.exe' 'c:\Users\ashwi\.vscode\extensions\ms-python.debugpy-2025.10.0-
win32-x64\bundled\libs\debugpy\launcher' '51637' '--' 'C:\Users\ashwi\OneDrive\Documents\AIAC BATCH-9\Lab 6.4\lab 6
.3 task-4.py'
Enter numbers separated by spaces: 2 3 6 9 8 7
The sum of the numbers is: 35.0
PS C:\Users\ashwi\OneDrive\Documents\AIAC BATCH-9\Lab 6.4>
PS C:\Users\ashwi\OneDrive\Documents\AIAC BATCH-9\Lab 6.4> ^C
PS C:\Users\ashwi\OneDrive\Documents\AIAC BATCH-9\Lab 6.4>
PS C:\Users\ashwi\OneDrive\Documents\AIAC BATCH-9\Lab 6.4>  c:; cd 'c:\Users\ashwi\OneDrive\Documents\AIAC BATCH-9\
Lab 6.4'; & 'c:\Program Files\Python313\python.exe' 'c:\Users\ashwi\.vscode\extensions\ms-python.debugpy-2025.10.0-
win32-x64\bundled\libs\debugpy\launcher' '51653' '--' 'C:\Users\ashwi\OneDrive\Documents\AIAC BATCH-9\Lab 6.4\lab 6
```

EXPLANATION :
IN THIS CODE USED A CLASS CALLED SUM TO CALUCALATE THE SUM OF NUMBERS

Task Description#5 (Class)

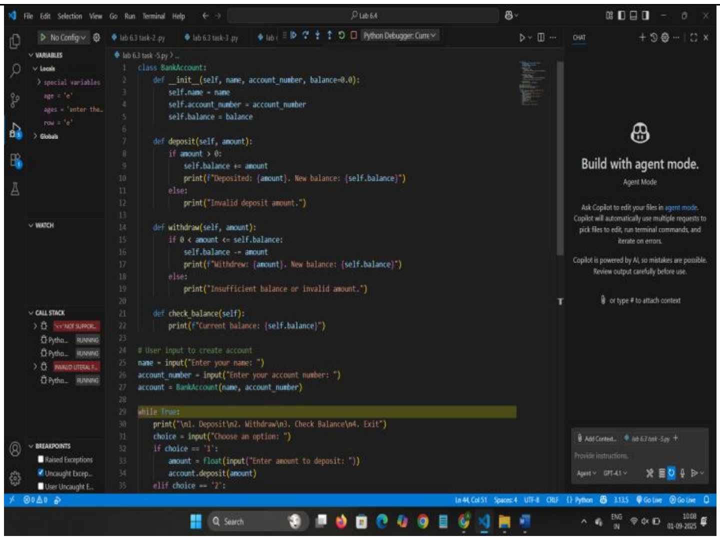
- Use AI to build a BankAccount class with deposit, withdraw, and balance methods.
- Analyze the generated code
- Add comments and explain code

Instructions

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

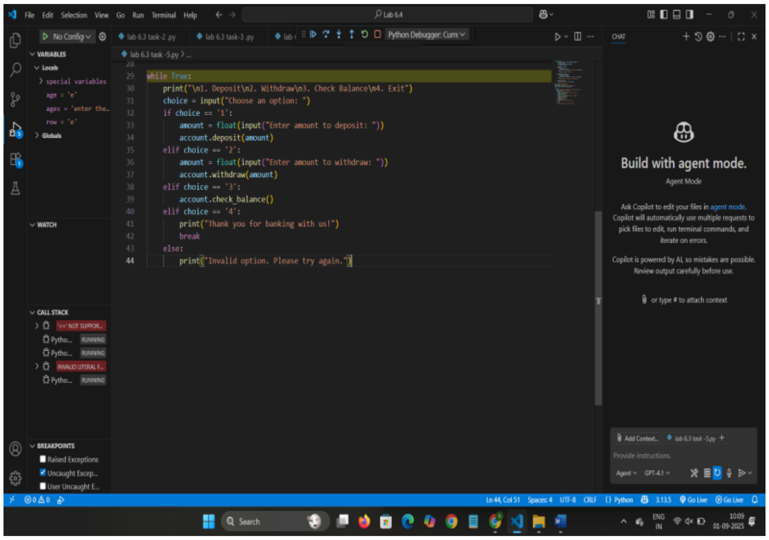
Expected Output#5

- Python code with explanation

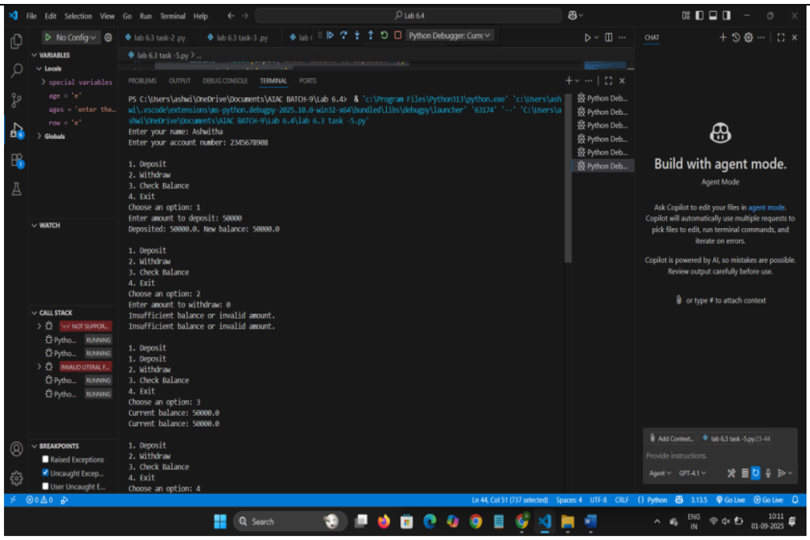


lab 6.3 task-2 .py
lab 6.3 task-3 .py
Python Debugger: Curre
lab 6.3 task -5.py
Lab 6.4
VARIABLES
WATCH
CALL STACK
BREAKPOINTS
Raised Exceptions
Uncaught Excep...
User Uncaught E...
CHAT
Build with agent mode.
Agent Mode
Ask Copilot to edit your files in agent mode. Copilot will automatically use multiple requests to pick files to edit, run terminal commands, and iterate on errors.
Copilot is powered by AI, so mistakes are possible. Review output carefully before use.
Ln 44, Col 51 Spaces: 4 UTF-8 CRLF Python

```python
class BankAccount:
    def __init__(self, name, account_number, balance=0.0):
        self.name = name
        self.account_number = account_number
        self.balance = balance

    def deposit(self, amount):
        if amount > 0:
            self.balance += amount
            print(f"Deposited: {amount}. New balance: {self.balance}")
        else:
            print("Invalid deposit amount.")

    def withdraw(self, amount):
        if 0 < amount <= self.balance:
            self.balance -= amount
            print(f"Withdrew: {amount}. New balance: {self.balance}")
        else:
            print("Insufficient balance or invalid amount.")

    def check_balance(self):
        print(f"Current balance: {self.balance}")

# User input to create account
name = input("Enter your name: ")
account_number = input("Enter your account number: ")
account = BankAccount(name, account_number)

while True:
    print("\n1. Deposit\n2. Withdraw\n3. Check Balance\n4. Exit")
    choice = input("Choose an option: ")
    if choice == '1':
        amount = float(input("Enter amount to deposit: "))
        account.deposit(amount)
    elif choice == '2':
```

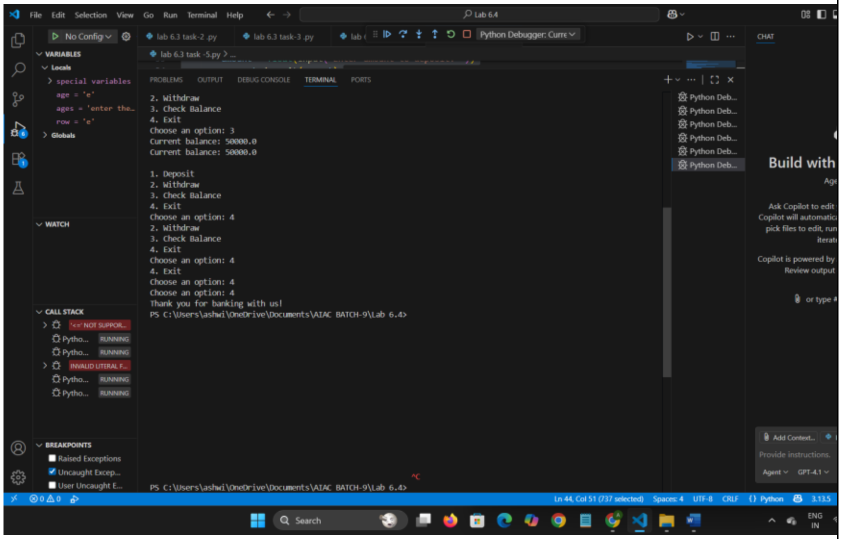


```python
while True:
    print("\n1. Deposit\n2. Withdraw\n3. Check Balance\n4. Exit")
    choice = input("Choose an option: ")
    if choice == '1':
        amount = float(input("Enter amount to deposit: "))
        account.deposit(amount)
    elif choice == '2':
        amount = float(input("Enter amount to withdraw: "))
        account.withdraw(amount)
    elif choice == '3':
        account.check_balance()
    elif choice == '4':
        print("Thank you for banking with us!")
        break
    else:
        print("Invalid option. Please try again.")
```



```
PS C:\Users\ashwi\OneDrive\Documents\AIAC BATCH-9\Lab 6.4>  & 'c:\Program Files\Python313\python.exe' 'c:\Users\ashwi\.vscode\extensions\ms-python.debugpy-2025.10.0-win32-x64\bundled\libs\debugpy\launcher' '63174' '--' 'C:\Users\ashwi\OneDrive\Documents\AIAC BATCH-9\Lab 6.4\lab 6.3 task -5.py'
Enter your name: Ashwitha
Enter your account number: 2345678908

1. Deposit
2. Withdraw
3. Check Balance
4. Exit
Choose an option: 1
Enter amount to deposit: 50000
Deposited: 50000.0. New balance: 50000.0

1. Deposit
2. Withdraw
3. Check Balance
4. Exit
Choose an option: 2
Enter amount to withdraw: 0
Insufficient balance or invalid amount.
Insufficient balance or invalid amount.

1. Deposit
1. Deposit
2. Withdraw
3. Check Balance
4. Exit
Choose an option: 3
Current balance: 50000.0
Current balance: 50000.0

1. Deposit
2. Withdraw
3. Check Balance
4. Exit
Choose an option: 4
```

Build with agent mode.
Agent Mode
Ask Copilot to edit your files in agent mode. Copilot will automatically use multiple requests to pick files to edit, run terminal commands, and iterate on errors.
Copilot is powered by AI, so mistakes are possible. Review output carefully before use.



```
2. Withdraw
3. Check Balance
4. Exit
Choose an option: 3
Current balance: 50000.0
Current balance: 50000.0

1. Deposit
2. Withdraw
3. Check Balance
4. Exit
Choose an option: 4
2. Withdraw
3. Check Balance
4. Exit
Choose an option: 4
4. Exit
Choose an option: 4
Choose an option: 4
Thank you for banking with us!
PS C:\Users\ashwi\OneDrive\Documents\AIAC BATCH-9\Lab 6.4>

                                                             ^C
PS C:\Users\ashwi\OneDrive\Documents\AIAC BATCH-9\Lab 6.4>
```

EXPLANATION:
IN THIS CODE WE USED A CLASS BANK TO USE ATTRIBUTES TO DEPOSIT WITHDRAW CHECKBALANCE OF A AMOUNT ENTERED BY THE USER

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**