SCHOOLOFCOMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE		DEPARTMENTOFCOMPUTER SCIENCE ENGINEERING	
ProgramName:B. Tech		**AssignmentType: 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
DateandDay of Assignment	Week7 - WednesDay	**Time(s)**	
Duration	2 Hours	**Applicableto Batches**	
AssignmentNumber:13.3(Presentassignmentnumber)/**24**(Totalnumberofassignments)			

Q.No.	Question	*ExpectedTime to complete*
1	**Lab 13 – Code Refactoring: Improving Legacy Code with AI Suggestions** **Lab Objectives** • To introduce the concept of code refactoring and why it matters (readability, maintainability, performance).	Week5 - Monday

- To practice using AI tools for identifying and suggesting improvements in legacy code.
- To evaluate the before vs. after versions for clarity, performance, and correctness.
- To reinforce responsible AI-assisted coding practices (avoiding over-reliance, validating outputs).

Learning Outcomes

After completing this lab, students will be able to:

1. Use AI to analyze and refactor poorly written Python code.
2. Improve code **readability, efficiency, and error handling**.
3. Document AI-suggested improvements through comments and explanations.
4. Apply refactoring strategies without changing functionality.
5. Critically reflect on AI's refactoring suggestions.

Task Description #1 – Remove Repetition

Task: Provide AI with the following redundant code and ask it to refactor

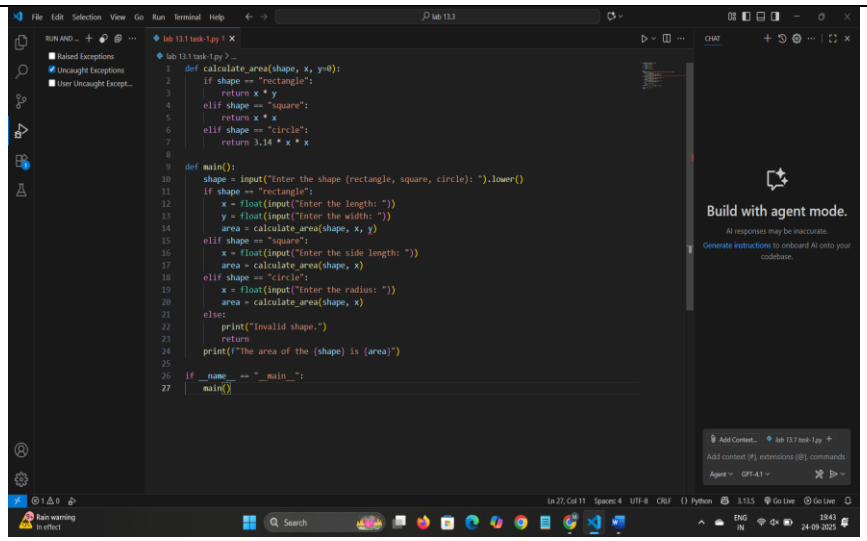
Python Code

```
def calculate_area(shape, x, y=0):
    if shape == "rectangle":
        return x * y
    elif shape == "square":
        return x * x
    elif shape == "circle":
        return 3.14 * x * x
```

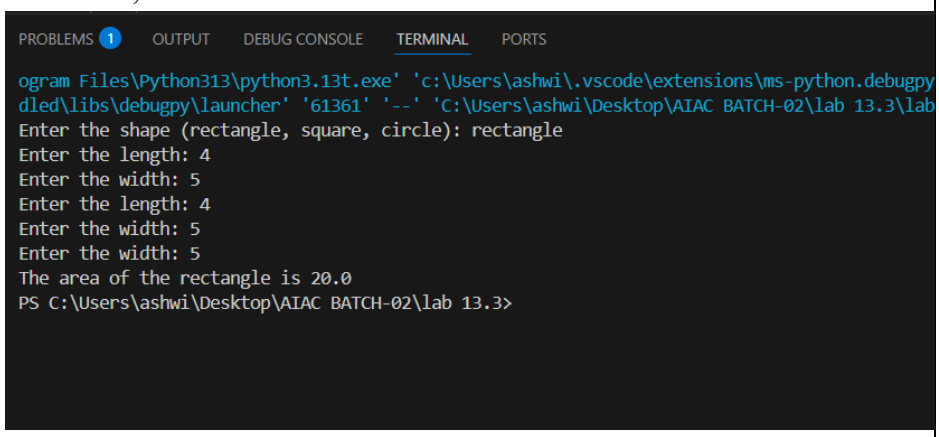
Expected Output

- Refactored version with dictionary-based dispatch or separate functions.
- Cleaner and modular design.

PROMPT: REFACTOR THE GIVEN CODE SEPARATE THE FUNCTIONS AND USER SHOULD ENTER THE INPUT PRINT THE OUTPUT WITHOUT ANY ERRORS



OUTPUT;



Task Description #2 – Error Handling in Legacy Code

Task: Legacy function without proper error handling

Python Code

```
def read_file(filename):
    f = open(filename, "r")
    data = f.read()
    f.close()
    return data
```

Expected Output:

AI refactors with with open() and try-except:

PROMPT: REFACTOR THE CODE BY USING OPEN AND TRYEXCEPT PRINT THE OUTPUT AND THE USER ENTER THE INPUT

lab 13.3task-2.py ✕ ~~task3(1).py~~

C: > Users > MEGHANA > OneDrive > Documents > AIAC > ASSIGNMENT-13.3 > lab 13.3ta

```python
def read_file(filename):
    try:
        with open(filename, "r") as f:
            data = f.read()
        return data
    except FileNotFoundError:
        print(f"Error: The file '{filename}' was not found.")
        return None
    except Exception as e:
        print(f"An error occurred: {e}")
        return None

if __name__ == "__main__":
    filename = input("Enter the filename to read: ")
    content = read_file(filename)
    if content is not None:
        print("File content:")
        print(content)
```

```
Enter the filename to read: python
Error: The file 'python' was not found.
PS C:\Users\MEGHANA\OneDrive\Documents\AIAC\ASSIGNMENT-13.3> ^C
```

Problems Output Debug Console **Terminal** Ports

```
Enter the filename to read: lab 13.3task-2.py
File content:
def read_file(filename):
    try:
        with open(filename, "r") as f:
            data = f.read()
        return data
    except FileNotFoundError:
        print(f"Error: The file '{filename}' was not found.")
        return None
    except Exception as e:
File content:
def read_file(filename):
    try:
        with open(filename, "r") as f:
            data = f.read()
        return data
    except FileNotFoundError:
        print(f"Error: The file '{filename}' was not found.")
        return None
    except Exception as e:
    try:
        with open(filename, "r") as f:
            data = f.read()
        return data
    except FileNotFoundError:
        print(f"Error: The file '{filename}' was not found.")
        return None
    except Exception as e:
            data = f.read()
        return data
    except FileNotFoundError:
        print(f"Error: The file '{filename}' was not found.")
        return None
```

Task Description #3 – Complex Refactoring

Task: Provide this legacy class to AI for readability and modularity improvements:

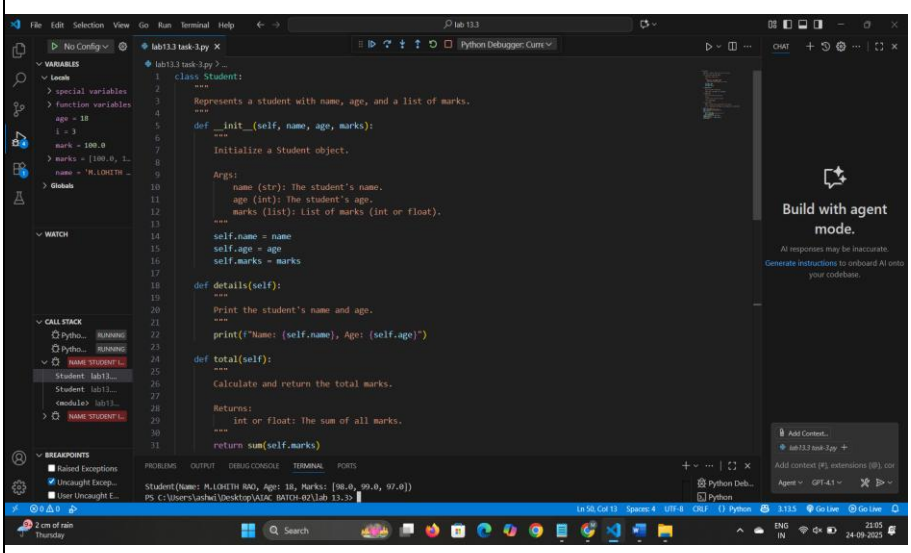
Python Code

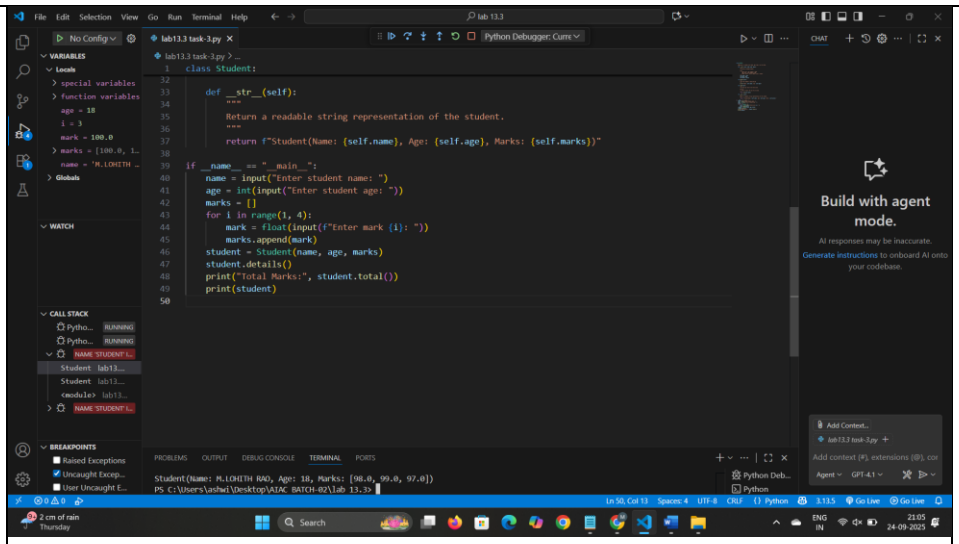
```
class Student:
    def __init__(self, n, a, m1, m2, m3):
        self.n = n
        self.a = a
        self.m1 = m1
        self.m2 = m2
        self.m3 = m3
    def details(self):
        print("Name:", self.n, "Age:", self.a)
    def total(self):
        return self.m1+self.m2+self.m3
```

Expected Output:

- AI improves naming (`name`, `age`, `marks`).
- Adds docstrings.
- Improves print readability.
- Possibly uses `sum(self.marks)` if marks stored in a list.

PROMPT: REFACTOR THE CODE BY USING THE DOCSTRINGS PRINT THE OUTPUT WHEN USER ENTERS THE INPUT






```
Enter student name: megha
Enter student age: 19
Enter mark 1: 98
Enter mark 2: 100
Enter mark 3: 97
Name: megha, Age: 19
Total Marks: 295.0
Student(Name: megha, Age: 19, Marks: [98.0, 100.0, 97.0])
PS C:\Users\MEGHANA\OneDrive\Documents\AIAC\ASSIGNMENT-13.3> ^C
```

Task Description #4 – Inefficient Loop Refactoring

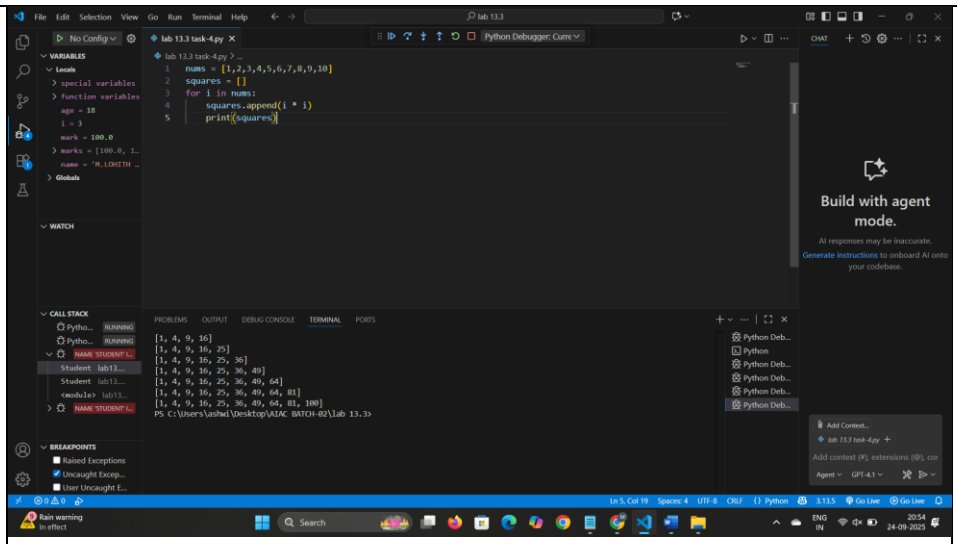
Task: Refactor this inefficient loop with AI help

Python Code

```
nums = [1,2,3,4,5,6,7,8,9,10]
squares = []
for i in nums:
    squares.append(i * i)
```

Expected Output: AI suggested a **list comprehension**

PROMPT: REFACTOR THE CODE BY COMPLETING THE LOOP AND PRINT OUTPUT


lab 13.3 task-4.py
nums = [1,2,3,4,5,6,7,8,9,10]
squares = []
for i in nums:
squares.append(i * i)
print(squares)
[1, 4, 9, 16]
[1, 4, 9, 16, 25]
[1, 4, 9, 16, 25, 36]
[1, 4, 9, 16, 25, 36, 49]
[1, 4, 9, 16, 25, 36, 49, 64]
[1, 4, 9, 16, 25, 36, 49, 64, 81]
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
PS C:\Users\ashwi\Desktop\UIAC BATCH-02\lab 13.3>
Build with agent mode.