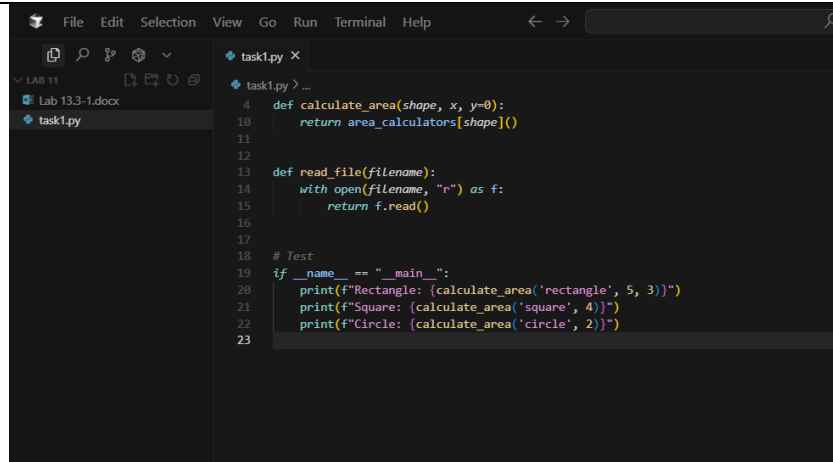


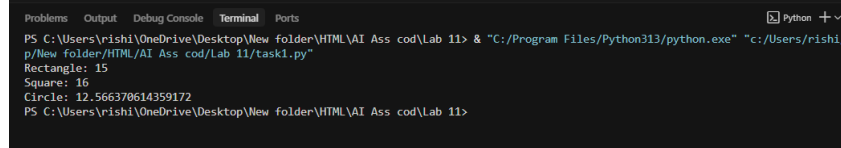
SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE		DEPARTMENT OF COMPUTER SCIENCE ENGINEERING	
Program Name: B. Tech		Assignment Type: Lab	Academic Year: 2025-2026
Course Coordinator Name		Venkataramana Veeramsetty	
Instructor(s) Name		Dr. V. Venkataramana (Co-ordinator)	
		Dr. T. Sampath Kumar	
		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)	
Course Code	24CS002PC215	Course Title	AI Assisted Coding
Year/Sem	II/I	Regulation	R24
Date and Day of Assignment	Week 7 - Wednesday	Time(s)	
Duration	2 Hours	Applicable to Batches	
Assignment Number: 13.3 (Present assignment number) / 24 (Total number of assignments)			
Q.No.	Question	Expected Time to complete	
1	<b>Lab 13 – Code Refactoring: Improving Legacy Code with AI Suggestions</b>  <b>Lab Objectives</b> <ul style="list-style-type: none"> <li>To introduce the concept of code refactoring and why it matters</li> </ul>	Week 5 - Monday	

	<p>(readability, maintainability, performance).</p> <ul style="list-style-type: none"> <li>• To practice using AI tools for identifying and suggesting improvements in legacy code.</li> <li>• To evaluate the before vs. after versions for clarity, performance, and correctness.</li> <li>• To reinforce responsible AI-assisted coding practices (avoiding over-reliance, validating outputs).</li> </ul> <p><b>Learning Outcomes</b></p> <p>After completing this lab, students will be able to:</p> <ol style="list-style-type: none"> <li>1. Use AI to analyze and refactor poorly written Python code.</li> <li>2. Improve code <b>readability, efficiency, and error handling</b>.</li> <li>3. Document AI-suggested improvements through comments and explanations.</li> <li>4. Apply refactoring strategies without changing functionality.</li> <li>5. Critically reflect on AI's refactoring suggestions.</li> </ol> <p><b>Task Description #1 – Remove Repetition</b></p> <p>Task: Provide AI with the following redundant code and ask it to refactor</p> <p><b>Python Code</b></p> <pre>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</pre> <p><b>Expected Output</b></p> <ul style="list-style-type: none"> <li>• Refactored version with dictionary-based dispatch or separate functions.</li> <li>• Cleaner and modular design.</li> </ul> <p><b>Code :</b></p>	
--	--	--



```
task1.py
4 def calculate_area(shape, x, y=0):
10     return area_calculators[shape]()
11
12
13 def read_file(filename):
14     with open(filename, "r") as f:
15         return f.read()
16
17
18 # Test
19 if __name__ == "__main__":
20     print(f"Rectangle: {calculate_area('rectangle', 5, 3)}")
21     print(f"Square: {calculate_area('square', 4)}")
22     print(f"Circle: {calculate_area('circle', 2)}")
23
```

•  
**Output:**



```
Problems Output Debug Console Terminal Ports
PS C:\Users\rishi\OneDrive\Desktop\New folder\HTML\AI Ass cod\Lab 11> & "C:/Program Files/Python313/python.exe" "c:/Users/rishi/p/New folder/HTML/AI Ass cod/Lab 11/task1.py"
Rectangle: 15
Square: 16
Circle: 12.566370614359172
PS C:\Users\rishi\OneDrive\Desktop\New folder\HTML\AI Ass cod\Lab 11>
```

## Task Description #2 – Error Handling in Legacy Code

Task: Legacy function without proper error handling

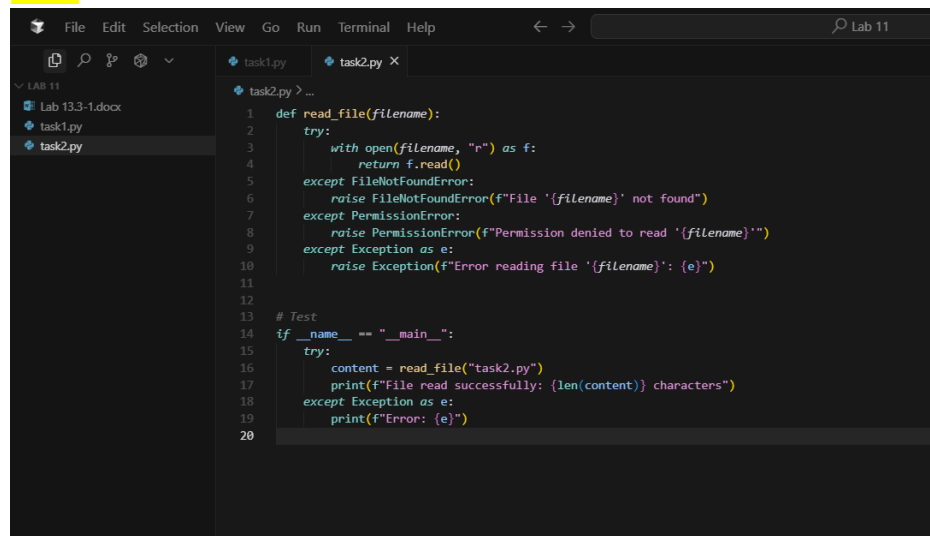
### Python Code

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

### Expected Output:

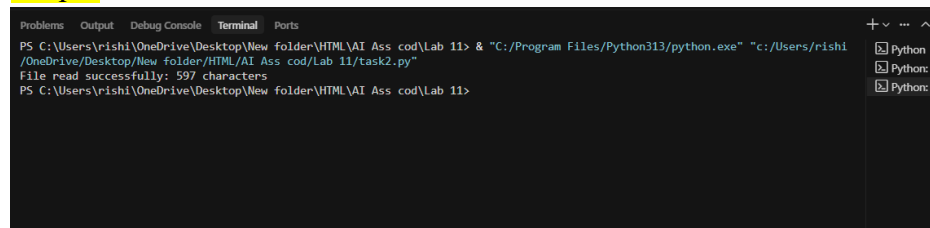
AI refactors with with open() and try-except:

## Code:



```
File Edit Selection View Go Run Terminal Help
task1.py task2.py x
task2.py > ...
1 def read_file(filename):
2     try:
3         with open(filename, "r") as f:
4             return f.read()
5     except FileNotFoundError:
6         raise FileNotFoundError(f"File '{filename}' not found")
7     except PermissionError:
8         raise PermissionError(f"Permission denied to read '{filename}'")
9     except Exception as e:
10        raise Exception(f"Error reading file '{filename}': {e}")
11
12
13 # Test
14 if __name__ == "__main__":
15     try:
16         content = read_file("task2.py")
17         print(f"File read successfully: {len(content)} characters")
18     except Exception as e:
19         print(f"Error: {e}")
20
```

## Output:



```
Problems Output Debug Console Terminal Ports
PS C:\Users\rishi\OneDrive\Desktop\New folder\HTML\AI Ass cod\Lab 11> & "C:/Program Files/Python313/python.exe" "c:/Users/rishi/OneDrive/Desktop/New folder/HTML/AI Ass cod/Lab 11/task2.py"
File read successfully: 597 characters
PS C:\Users\rishi\OneDrive\Desktop\New folder\HTML\AI Ass cod\Lab 11>
```

## Task Description #3 – Complex Refactoring

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

### Python Code

class Student:

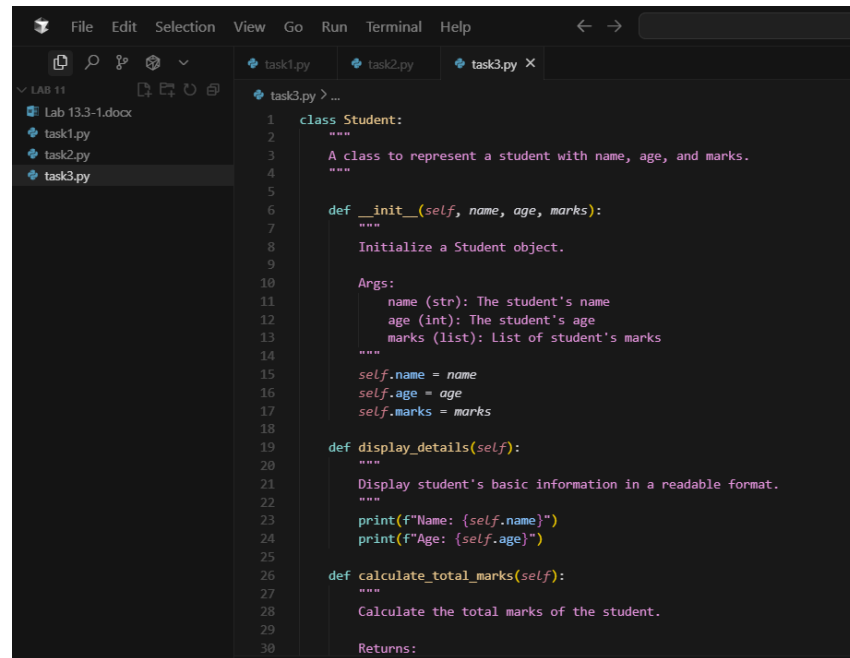
```
def __init__(self, n, a, m1, m2, m3):
    self.n = n
    self.a = a
    self.m1 = m1
    self.m2 = m2
    self.m3 = m3
def details(self):
    print("Name:", self.n, "Age:", self.a)
def total(self):
```

```
return self.m1+self.m2+self.m3
```

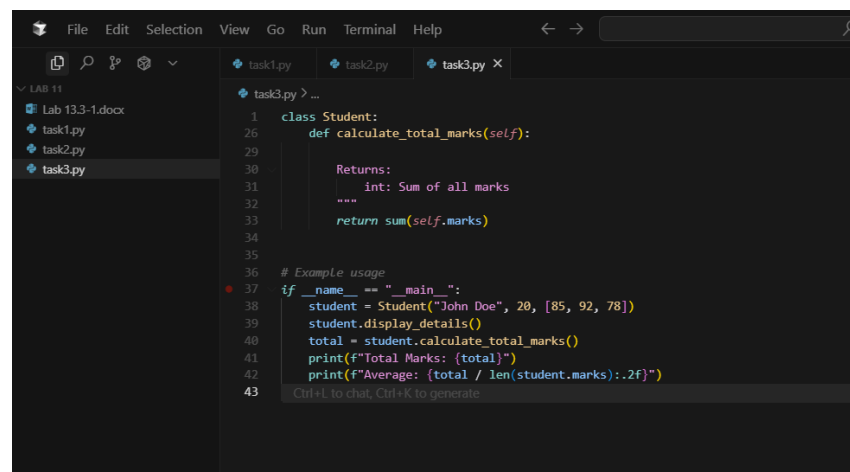
### Expected Output:

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

### CODE:



```
1 class Student:
2     """
3     A class to represent a student with name, age, and marks.
4     """
5
6     def __init__(self, name, age, marks):
7         """
8         Initialize a Student object.
9
10        Args:
11            name (str): The student's name
12            age (int): The student's age
13            marks (list): List of student's marks
14        """
15        self.name = name
16        self.age = age
17        self.marks = marks
18
19    def display_details(self):
20        """
21        Display student's basic information in a readable format.
22        """
23        print(f"Name: {self.name}")
24        print(f"Age: {self.age}")
25
26    def calculate_total_marks(self):
27        """
28        Calculate the total marks of the student.
29
30        Returns:
```



```
26    def calculate_total_marks(self):
29
30        Returns:
31            int: Sum of all marks
32        """
33        return sum(self.marks)
34
35
36 # Example usage
37 if __name__ == "__main__":
38     student = Student("John Doe", 20, [85, 92, 78])
39     student.display_details()
40     total = student.calculate_total_marks()
41     print(f"Total Marks: {total}")
42     print(f"Average: {total / len(student.marks):.2f}")
43
```

## OUTPUT:

```
Problems Output Debug Console Terminal Ports
PS C:\Users\rishi\OneDrive\Desktop\New folder\HTML\AI Ass cod\Lab 11> & "C:/Program Files/Python313/python.exe" "c:/Users/rishi/OneDrive/Desktop/New folder/HTML/AI Ass cod/Lab 11/task2.py"
File read successfully: 597 characters
PS C:\Users\rishi\OneDrive\Desktop\New folder\HTML\AI Ass cod\Lab 11>
```

## 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

## CODE:

```
File Edit Selection View Go Run Terminal Help
task1.py task2.py task3.py task4.py x
Lab 133-1.docx
task1.py
task2.py
task3.py
task4.py
task4.py > ...
1  nums = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
2
3  # List comprehension way - cleaner and more Pythonic
4  squares = [i * i for i in nums]
5
6  # Additional examples of list comprehensions
7  even_squares = [i * i for i in nums if i % 2 == 0]
8  cubes = [i ** 3 for i in nums]
9  strings = [str(i) for i in nums]
10
11 print(f"Numbers: {nums}")
12 print(f"Squares: {squares}")
13 print(f"Even squares: {even_squares}")
14 print(f"Cubes: {cubes}")
15 print(f"String numbers: {strings}")
16
```

## OUTPUT:

```
Problems Output Debug Console Terminal Ports
PS C:\Users\rishi\OneDrive\Desktop\New folder\HTML\AI Ass cod\Lab 11> & "C:/Program Files/Python313/python.exe" "c:/Users/rishi/OneDrive/Desktop/New folder/HTML/AI Ass cod/Lab 11/task4.py"
Numbers: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
Squares: [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
Even squares: [4, 16, 36, 64, 100]
Cubes: [1, 8, 27, 64, 125, 216, 343, 512, 729, 1000]
String numbers: ['1', '2', '3', '4', '5', '6', '7', '8', '9', '10']
PS C:\Users\rishi\OneDrive\Desktop\New folder\HTML\AI Ass cod\Lab 11>
```