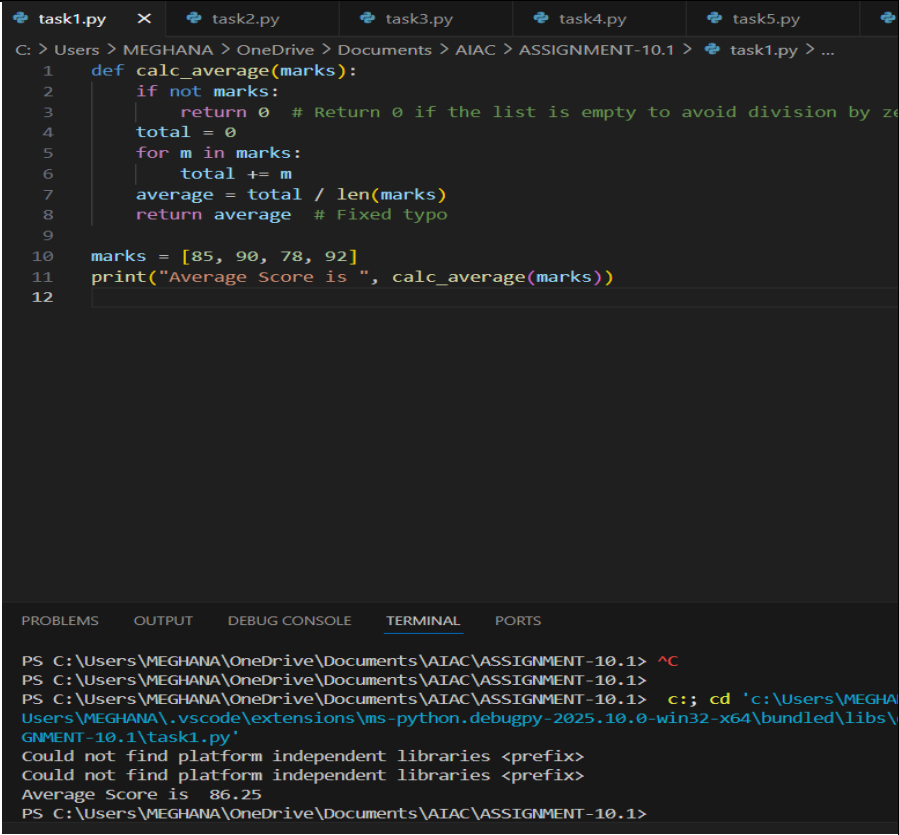
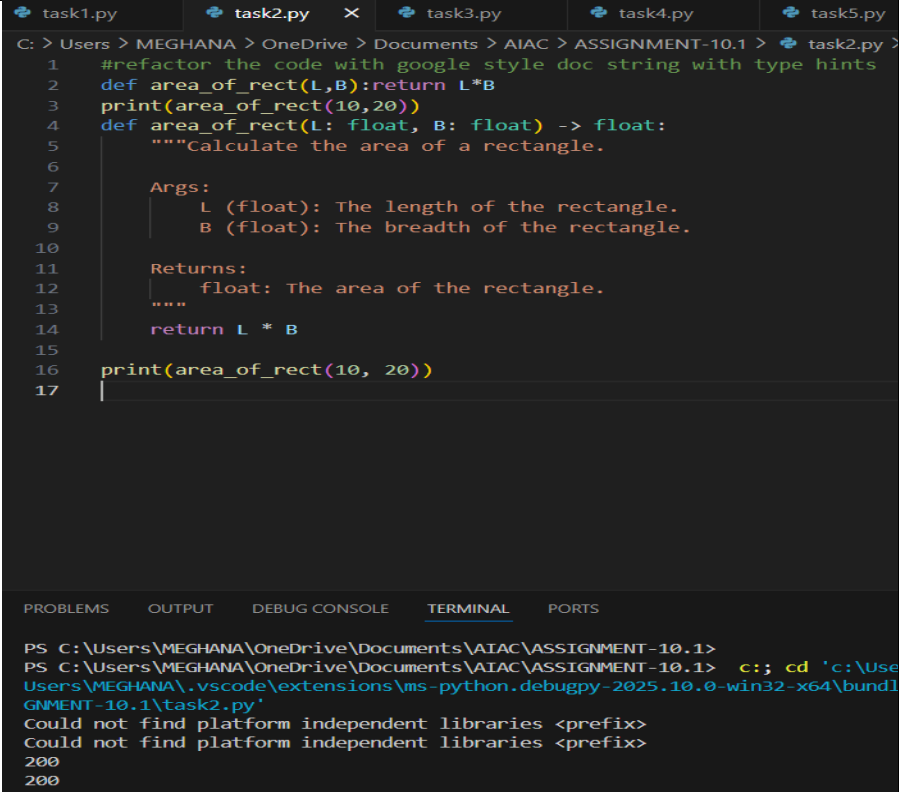
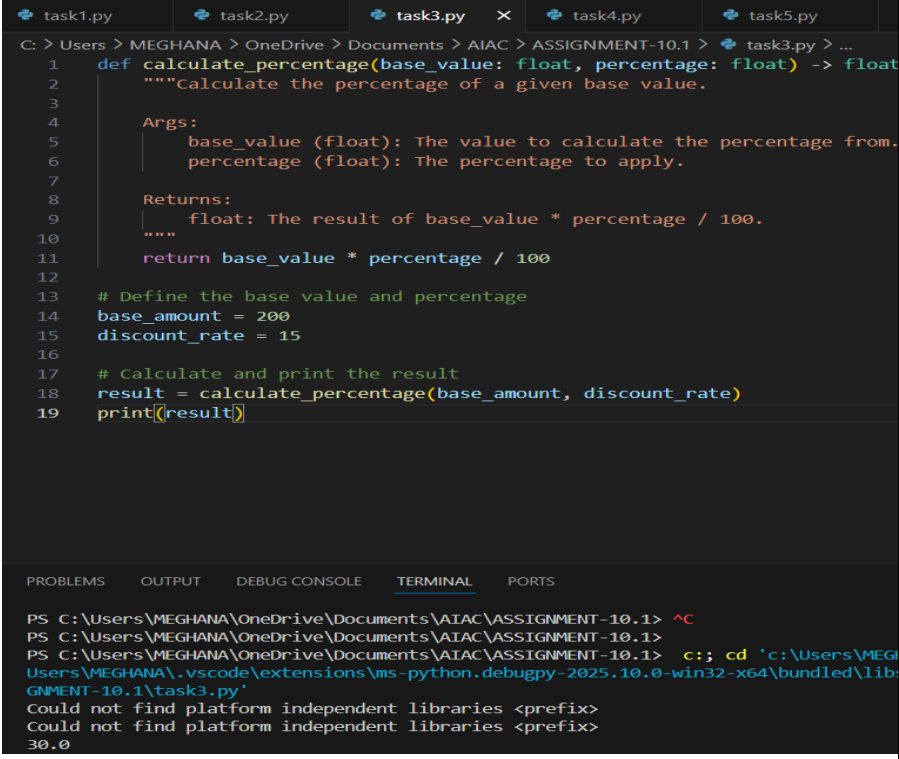


SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE		DEPARTMENT OF COMPUTER SCIENCE ENGINEERING	
Program Name: B. Tech		Assignment Type: Lab	Academic Year:2025-2026
Course Coordinator Name		Venkataramana Veeramsetty	
Instructor(s) Name		Dr. V. Venkataramana (Co-ordinator)	
		Dr. T. Sampath Kumar	
		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	Week5 - Monday	Time(s)	
Duration	2 Hours	Applicable to Batches	
AssignmentNumber:10.1(Present assignment number)/24(Total number of assignments)			
Q.No.	Question		Expected Time to complete
1	<b>Lab 10 – Code Review and Quality: Using AI to Improve Code Quality and Readability</b> <b>Lab Objectives</b> <ul style="list-style-type: none"> <li>Use AI for automated code review and quality enhancement.</li> <li>Identify and fix syntax, logical, performance, and security issues in Python code.</li> <li>Improve readability and maintainability through structured refactoring and comments.</li> </ul>		Week5 - Monday

	<ul style="list-style-type: none"> <li>• Apply prompt engineering for targeted improvements.</li> <li>• Evaluate AI-generated suggestions against PEP 8 standards and software engineering best practices</li> </ul>	
	<p><b>Task Description #1 – Syntax and Logic Errors</b></p> <p>Task: Use AI to identify and fix syntax and logic errors in a faulty Python script.</p> <p>Sample Input Code:</p> <pre># Calculate average score of a student def calc_average(marks):     total = 0     for m in marks:         total += m     average = total / len(marks)     return avrage # Typo here  marks = [85, 90, 78, 92] print("Average Score is ", calc_average(marks))</pre> <p>Expected Output:</p> <ul style="list-style-type: none"> <li>• Corrected and runnable Python code with explanations of the fixes.</li> </ul>	

	 <pre> task1.py x task2.py task3.py task4.py task5.py C: &gt; Users &gt; MEGHANA &gt; OneDrive &gt; Documents &gt; AIAC &gt; ASSIGNMENT-10.1 &gt; task1.py &gt; ... 1 def calc_average(marks): 2     if not marks: 3         return 0 # Return 0 if the list is empty to avoid division by zero 4     total = 0 5     for m in marks: 6         total += m 7     average = total / len(marks) 8     return average # Fixed typo 9 10 marks = [85, 90, 78, 92] 11 print("Average Score is ", calc_average(marks)) 12 </pre> <p>PROBLEMS OUTPUT DEBUG CONSOLE <u>TERMINAL</u> PORTS</p> <pre> PS C:\Users\MEGHANA\OneDrive\Documents\AIAC\ASSIGNMENT-10.1&gt; ^C PS C:\Users\MEGHANA\OneDrive\Documents\AIAC\ASSIGNMENT-10.1&gt; PS C:\Users\MEGHANA\OneDrive\Documents\AIAC\ASSIGNMENT-10.1&gt; c:;; cd 'c:\Users\MEGHANA\OneDrive\Documents\AIAC\ASSIGNMENT-10.1' PS C:\Users\MEGHANA\OneDrive\Documents\AIAC\ASSIGNMENT-10.1&gt; python task1.py Could not find platform independent libraries &lt;prefix&gt; Could not find platform independent libraries &lt;prefix&gt; Average Score is 86.25 PS C:\Users\MEGHANA\OneDrive\Documents\AIAC\ASSIGNMENT-10.1&gt; </pre>	
	<p><b>Task Description #2 – PEP 8 Compliance</b></p> <p>Task: Use AI to refactor Python code to follow PEP 8 style guidelines.</p> <p>Sample Input Code:</p> <pre>def area_of_rect(L,B):return L*B print(area_of_rect(10,20))</pre> <p>Expected Output:</p> <ul style="list-style-type: none"> <li>Well-formatted PEP 8-compliant Python code.</li> </ul>	

	 <pre> task1.py task2.py X task3.py task4.py task5.py C: &gt; Users &gt; MEGHANA &gt; OneDrive &gt; Documents &gt; AIAC &gt; ASSIGNMENT-10.1 &gt; task2.py 1  #refactor the code with google style doc string with type hints 2  def area_of_rect(L,B):return L*B 3  print(area_of_rect(10,20)) 4  def area_of_rect(L: float, B: float) -&gt; float: 5      """Calculate the area of a rectangle. 6 7      Args: 8          L (float): The length of the rectangle. 9          B (float): The breadth of the rectangle. 10 11     Returns: 12         float: The area of the rectangle. 13     """ 14     return L * B 15 16 print(area_of_rect(10, 20)) 17 PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS PS C:\Users\MEGHANA\OneDrive\Documents\AIAC\ASSIGNMENT-10.1&gt; PS C:\Users\MEGHANA\OneDrive\Documents\AIAC\ASSIGNMENT-10.1&gt; c;; cd 'c:\Use Users\MEGHANA\.vscode\extensions\ms-python.debugpy-2025.10.0-win32-x64\bundl GNMENT-10.1\task2.py' Could not find platform independent libraries &lt;prefix&gt; Could not find platform independent libraries &lt;prefix&gt; 200 200 </pre>	
	<p><b>Task Description #3 – Readability Enhancement</b></p> <p>Task: Use AI to make code more readable without changing its logic.</p> <p>Sample Input Code:</p> <pre>def c(x,y):     return x*y/100 a=200 b=15 print(c(a,b))</pre> <p>Expected Output:</p> <ul style="list-style-type: none"> <li>Python code with descriptive variable names, inline comments, and clear formatting.</li> </ul>	

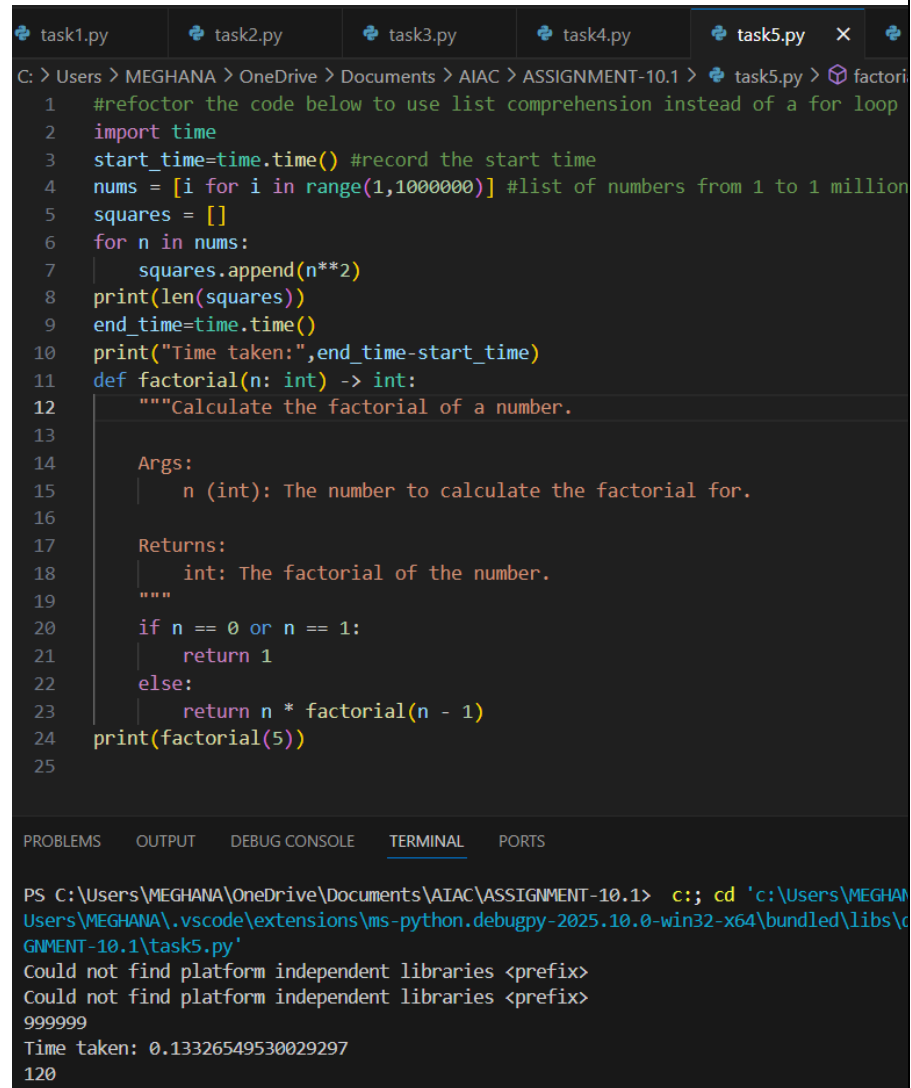
	 <pre> task1.py task2.py task3.py X task4.py task5.py C: &gt; Users &gt; MEGHANA &gt; OneDrive &gt; Documents &gt; AIAC &gt; ASSIGNMENT-10.1 &gt; task3.py &gt; ... 1  def calculate_percentage(base_value: float, percentage: float) -&gt; float: 2      """Calculate the percentage of a given base value. 3 4      Args: 5          base_value (float): The value to calculate the percentage from. 6          percentage (float): The percentage to apply. 7 8      Returns: 9          float: The result of base_value * percentage / 100. 10     """ 11     return base_value * percentage / 100 12 13 # Define the base value and percentage 14 base_amount = 200 15 discount_rate = 15 16 17 # Calculate and print the result 18 result = calculate_percentage(base_amount, discount_rate) 19 print(result)  PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS  PS C:\Users\MEGHANA\OneDrive\Documents\AIAC\ASSIGNMENT-10.1&gt; ^C PS C:\Users\MEGHANA\OneDrive\Documents\AIAC\ASSIGNMENT-10.1&gt; PS C:\Users\MEGHANA\OneDrive\Documents\AIAC\ASSIGNMENT-10.1&gt; c:: cd 'c:\Users\MEGHANA\OneDrive\Documents\AIAC\ASSIGNMENT-10.1' PS C:\Users\MEGHANA\OneDrive\Documents\AIAC\ASSIGNMENT-10.1&gt; python task3.py Users\MEGHANA\.vscode\extensions\ms-python.debugpy-2025.10.0-win32-x64\bundle\lib\ GNMENT-10.1\task3.py' Could not find platform independent libraries &lt;prefix&gt; Could not find platform independent libraries &lt;prefix&gt; 30.0 </pre>	
	<p><b>Task Description #4 – Refactoring for Maintainability</b></p> <p>Task: Use AI to break repetitive or long code into reusable functions.</p> <p>Sample Input Code:</p> <pre>students = ["Alice", "Bob", "Charlie"] print("Welcome", students[0]) print("Welcome", students[1]) print("Welcome", students[2])</pre> <p>Expected Output:</p> <ul style="list-style-type: none"> <li>Modular code with reusable functions.</li> </ul>	

	 <pre> task1.py task2.py task3.py task4.py X task5.py C: &gt; Users &gt; MEGHANA &gt; OneDrive &gt; Documents &gt; AIAC &gt; ASSIGNMENT-10.1 &gt; task4.py 1  def welcome_student(name: str) -&gt; None: 2      """Print a welcome message for a student. 3 4      Args: 5          name (str): The name of the student. 6      """ 7      print("Welcome", name) 8 9  students = ["Alice", "Bob", "Charlie"] 10 11  for student in students: 12      welcome_student(student) </pre> <p>PROBLEMS OUTPUT DEBUG CONSOLE <u>TERMINAL</u> PORTS</p> <pre> PS C:\Users\MEGHANA\OneDrive\Documents\AIAC\ASSIGNMENT-10.1&gt; c:; cd 'c:\Us Users\MEGHANA\.vscode\extensions\ms-python.debugpy-2025.10.0-win32-x64\bund GNMENT-10.1\task4.py' Could not find platform independent libraries &lt;prefix&gt; Could not find platform independent libraries &lt;prefix&gt; Welcome Alice Welcome Bob Welcome Charlie </pre>	
	<p><b>Task Description #5 – Performance Optimization</b></p> <p>Task: Use AI to make the code run faster.</p> <p>Sample Input Code:</p> <pre> # Find squares of numbers nums = [i for i in range(1,1000000)] squares = [] for n in nums:     squares.append(n**2) </pre>	

print(len(squares))

Expected Output:

- Optimized code using list comprehensions or vectorized operations.



The screenshot shows a VS Code editor with a file explorer at the top displaying tabs for task1.py, task2.py, task3.py, task4.py, and task5.py. The main editor window shows a Python script in task5.py. The script includes a comment about refactoring a for loop into a list comprehension, imports the time module, records start and end times, creates a list of numbers from 1 to 1,000,000, and calculates their squares. It also defines a factorial function with a docstring and prints the factorial of 5. The terminal at the bottom shows the command to run the script, followed by output indicating that platform-independent libraries could not be found, a series of 9s, the execution time, and the result 120.

```
1 #refactor the code below to use list comprehension instead of a for loop
2 import time
3 start_time=time.time() #record the start time
4 nums = [i for i in range(1,1000000)] #list of numbers from 1 to 1 million
5 squares = []
6 for n in nums:
7     squares.append(n**2)
8 print(len(squares))
9 end_time=time.time()
10 print("time taken:",end_time-start_time)
11 def factorial(n: int) -> int:
12     """Calculate the factorial of a number.
13
14     Args:
15         n (int): The number to calculate the factorial for.
16
17     Returns:
18         int: The factorial of the number.
19     """
20     if n == 0 or n == 1:
21         return 1
22     else:
23         return n * factorial(n - 1)
24 print(factorial(5))
25
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\MEGHANA\OneDrive\Documents\AIAC\ASSIGNMENT-10.1> c;; cd 'c:\Users\MEGHANA\OneDrive\Documents\AIAC\ASSIGNMENT-10.1' & python 'c:\Users\MEGHANA\OneDrive\Documents\AIAC\ASSIGNMENT-10.1\task5.py'

Could not find platform independent libraries <prefix>  
Could not find platform independent libraries <prefix>  
999999  
Time taken: 0.13326549530029297  
120

### Task Description #6 – Complexity Reduction

Task: Use AI to simplify overly complex logic.

Sample Input Code:

```
def grade(score):
    if score >= 90:
        return "A"
    else:
        if score >= 80:
```

```
        return "B"
    else:
        if score >= 70:
            return "C"
        else:
            if score >= 60:
                return "D"
            else:
                return "F"
```

Expected Output:

- Cleaner logic using elif or dictionary mapping.

```
task1.py task2.py task3.py task4.py task5.py task6.py
C: > Users > MEGHANA > OneDrive > Documents > AIAC > ASSIGNMENT-10.1 > task6.py > ...
1  def grade(score):
2      if score >= 90:
3          return "A"
4      else:
5          if score >= 80:
6              return "B"
7          else:
8              if score >= 70:
9                  return "C"
10             else:
11                 if score >= 60:
12                     return "D"
13                 else:
14                     return "F"
15             # Cleaner logic using elif
16             # Option 1: Using elif
17             # Remove all nested else blocks and use elif for clarity
18
19             # Option 1 implementation:
20             # (Place this as the entire function body)
21
22             if score >= 90:
23                 return "A"
24             elif score >= 80:
25                 return "B"
26             elif score >= 70:
27                 return "C"
28             elif score >= 60:
29                 return "D"
30             else:
31                 return "F"
32 print(grade(55)) # Output: "F"
33 print(grade(95)) # Output: "A"
34 print(grade(85)) # Output: "B"
35 print(grade(75)) # Output: "C"
36 print(grade(65)) # Output: "D"
37 print(grade(55)) # Output: "F"
```

```
37 print(grade(55)) # Output: "F"
38
39 # Option 2: Using dictionary mapping with ranges
40 # (Alternative approach)
41
42 # grades = {range(90, 101): "A", range(80, 90): "B", range(70, 80): "C", range(60, 70): "D"}
43 # for r, grade_letter in grades.items():
44 #     if score in r:
45 #         return grade_letter
46 # return "F"
```

OUTPUT:



F  
A  
B  
C  
D  
F

PS C:\Users\MEGHANA\OneDrive\Documents\AIAC\ASSIGNMENT-10.1>