

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	Week4 - Wednesday	Time(s)	
Duration	2 Hours	Applicable to Batches	
AssignmentNumber:9.3(Present assignment number)/24(Total number of assignments)			
Q.No.	Question		Expected Time to complete
1	Lab 9: Documentation Generation: Automatic documentation and code comments Lab Objectives: <ul style="list-style-type: none"> To understand the importance of documentation and code comments in software development. To explore how AI-assisted coding tools can generate meaningful documentation and 		Week4 - Wednesday

inline comments.

- To practice generating function-level and module-level docstrings automatically.
- To evaluate the quality, accuracy, and limitations of AI-generated documentation.
- To develop a small automated tool for documentation generation in Python..

Lab Outcomes (LOs):

After completing this lab, students will be able to:

- Apply AI-assisted coding tools to generate docstrings and inline comments for Python code.
- Critically analyze AI-generated documentation for correctness, completeness, and readability.
- Create structured documentation (function-level, module-level) following standard formats.
- Design and implement a mini documentation generator tool to automate code commenting and docstring creation.

Task Description#1 Basic Docstring Generation

- Write python function to return sum of even and odd numbers in the given list.
- Incorporate manual **docstring** in code with Google Style
- Use an AI-assisted tool (e.g., Gemini, Copilot, Cursor AI) to generate a docstring describing the function.
- Compare the AI-generated docstring with your manually written one.

Prompt: "Write a Python function to return the sum of even and odd numbers in a list and generate a Google-style docstring for it."

```
10 def sum_even_odd(numbers): """Sum of even and odd numbers in a list.
11     Parameters:
12     - numbers (list): A list of integers.
13     Returns:
14     - tuple: A tuple containing the sum of even numbers and the sum of odd numbers.
15     """
16     even_sum = 0
17     odd_sum = 0
18     for num in numbers:
19         if num % 2 == 0:
20             even_sum += num
21         else:
22             odd_sum += num
23     return even_sum, odd_sum
24
25 # Example usage
26 numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
27 even_sum, odd_sum = sum_even_odd(numbers)
28 print(f"Sum of even numbers: {even_sum}, Sum of odd numbers: {odd_sum}")
```

Expected Outcome#1: Students understand how AI can produce function-level documentation

```
> & C:/Users/HARSHINI/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/HARSHINI/Downloads/aiaa/B_2aiaa.py
Sum of even numbers: 30
Sum of odd numbers: 25
PS C:/Users/HARSHINI/Downloads/aiaa> []
```

Task Description#2 Automatic Inline Comments

- Write python program for **sru_student** class with attributes like name, roll no., hostel_status and **fee_update** method and **display_details** method.
- Write comments manually for each line/code block
- Ask an AI tool to add inline comments explaining each line/step.
- Compare the AI-generated comments with your manually written one.

Prompt: "Write a Python program for a sru_student class with attributes and methods, and generate inline comments explaining each line of code."

```

# 3. Summary
# 4. Summary
# 5. Summary
# 6. Summary
# 7. Summary
# 8. Summary
# 9. Summary
# 10. Summary
# 11. Summary
# 12. Summary
# 13. Summary
# 14. Summary
# 15. Summary
# 16. Summary
# 17. Summary
# 18. Summary
# 19. Summary
# 20. Summary
# 21. Summary
# 22. Summary
# 23. Summary
# 24. Summary
# 25. Summary
# 26. Summary
# 27. Summary
# 28. Summary
# 29. Summary
# 30. Summary
# 31. Summary
# 32. Summary
# 33. Summary
# 34. Summary
# 35. Summary
# 36. Summary
# 37. Summary
# 38. Summary
# 39. Summary
# 40. Summary
# 41. Summary
# 42. Summary
# 43. Summary
# 44. Summary
# 45. Summary
# 46. Summary
# 47. Summary
# 48. Summary
# 49. Summary
# 50. Summary
# 51. Summary
# 52. Summary
# 53. Summary
# 54. Summary
# 55. Summary
# 56. Summary
# 57. Summary
# 58. Summary
# 59. Summary
# 60. Summary
# 61. Summary
# 62. Summary
# 63. Summary
# 64. Summary
# 65. Summary
# 66. Summary
# 67. Summary
# 68. Summary
# 69. Summary
# 70. Summary
# 71. Summary
# 72. Summary
# 73. Summary
# 74. Summary
# 75. Summary
# 76. Summary
# 77. Summary
# 78. Summary
# 79. Summary
# 80. Summary
# 81. Summary
# 82. Summary
# 83. Summary
# 84. Summary
# 85. Summary
# 86. Summary
# 87. Summary
# 88. Summary
# 89. Summary
# 90. Summary
# 91. Summary
# 92. Summary
# 93. Summary
# 94. Summary
# 95. Summary
# 96. Summary
# 97. Summary
# 98. Summary
# 99. Summary
# 100. Summary

```

Expected Output#2: Students critically analyze AI-generated code comments

```

# 1. Summary
# 2. Summary
# 3. Summary
# 4. Summary
# 5. Summary
# 6. Summary
# 7. Summary
# 8. Summary
# 9. Summary
# 10. Summary
# 11. Summary
# 12. Summary
# 13. Summary
# 14. Summary
# 15. Summary
# 16. Summary
# 17. Summary
# 18. Summary
# 19. Summary
# 20. Summary
# 21. Summary
# 22. Summary
# 23. Summary
# 24. Summary
# 25. Summary
# 26. Summary
# 27. Summary
# 28. Summary
# 29. Summary
# 30. Summary
# 31. Summary
# 32. Summary
# 33. Summary
# 34. Summary
# 35. Summary
# 36. Summary
# 37. Summary
# 38. Summary
# 39. Summary
# 40. Summary
# 41. Summary
# 42. Summary
# 43. Summary
# 44. Summary
# 45. Summary
# 46. Summary
# 47. Summary
# 48. Summary
# 49. Summary
# 50. Summary
# 51. Summary
# 52. Summary
# 53. Summary
# 54. Summary
# 55. Summary
# 56. Summary
# 57. Summary
# 58. Summary
# 59. Summary
# 60. Summary
# 61. Summary
# 62. Summary
# 63. Summary
# 64. Summary
# 65. Summary
# 66. Summary
# 67. Summary
# 68. Summary
# 69. Summary
# 70. Summary
# 71. Summary
# 72. Summary
# 73. Summary
# 74. Summary
# 75. Summary
# 76. Summary
# 77. Summary
# 78. Summary
# 79. Summary
# 80. Summary
# 81. Summary
# 82. Summary
# 83. Summary
# 84. Summary
# 85. Summary
# 86. Summary
# 87. Summary
# 88. Summary
# 89. Summary
# 90. Summary
# 91. Summary
# 92. Summary
# 93. Summary
# 94. Summary
# 95. Summary
# 96. Summary
# 97. Summary
# 98. Summary
# 99. Summary
# 100. Summary

```

Task Description#3

Prompt: Write a Python calculator script with add, subtract, multiply, and divide functions, and create NumPy-style docstrings for the module and each function

```

# 1. Summary
# 2. Summary
# 3. Summary
# 4. Summary
# 5. Summary
# 6. Summary
# 7. Summary
# 8. Summary
# 9. Summary
# 10. Summary
# 11. Summary
# 12. Summary
# 13. Summary
# 14. Summary
# 15. Summary
# 16. Summary
# 17. Summary
# 18. Summary
# 19. Summary
# 20. Summary
# 21. Summary
# 22. Summary
# 23. Summary
# 24. Summary
# 25. Summary
# 26. Summary
# 27. Summary
# 28. Summary
# 29. Summary
# 30. Summary
# 31. Summary
# 32. Summary
# 33. Summary
# 34. Summary
# 35. Summary
# 36. Summary
# 37. Summary
# 38. Summary
# 39. Summary
# 40. Summary
# 41. Summary
# 42. Summary
# 43. Summary
# 44. Summary
# 45. Summary
# 46. Summary
# 47. Summary
# 48. Summary
# 49. Summary
# 50. Summary
# 51. Summary
# 52. Summary
# 53. Summary
# 54. Summary
# 55. Summary
# 56. Summary
# 57. Summary
# 58. Summary
# 59. Summary
# 60. Summary
# 61. Summary
# 62. Summary
# 63. Summary
# 64. Summary
# 65. Summary
# 66. Summary
# 67. Summary
# 68. Summary
# 69. Summary
# 70. Summary
# 71. Summary
# 72. Summary
# 73. Summary
# 74. Summary
# 75. Summary
# 76. Summary
# 77. Summary
# 78. Summary
# 79. Summary
# 80. Summary
# 81. Summary
# 82. Summary
# 83. Summary
# 84. Summary
# 85. Summary
# 86. Summary
# 87. Summary
# 88. Summary
# 89. Summary
# 90. Summary
# 91. Summary
# 92. Summary
# 93. Summary
# 94. Summary
# 95. Summary
# 96. Summary
# 97. Summary
# 98. Summary
# 99. Summary
# 100. Summary

```

	<p>Expected Output#3: Students learn structured documentation for multi-function scripts</p> <pre>> & C:/Users/HARSHINI/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/HARSHINI/Downloads/aiac/8_2aiac.py 10 + 5 = 15 10 - 5 = 5 10 * 5 = 50 10 / 5 = 2.0 10 / 0 = Error: Cannot divide by zero PS C:\Users\HARSHINI\Downloads\aiac> </pre> <p>Push documentation whole workspace as .md file in GitHub Repository</p> <p>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</p>	
--	--	--