|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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 | **Applicableto**  **Batches** | |  | | | |
| **AssignmentNumber:7.3**(Present assignment number)/**24**(Total number of assignments) | | | | | | | | | |
|  | | | | | | | | | |
|  | | | | | | | | | |
|  | **Q.No.** | **Question** | | | | | | ***ExpectedTime***  ***to complete*** |  |
|  | 1 | Lab 7: AI-Error Debugging with AI: Systematic approaches to finding and fixing bugs  **Lab Objectives:**   * To identify and correct syntax, logic, and runtime errors in Python programs using AI tools. * To understand common programming bugs and AI-assisted debugging suggestions. * To evaluate how AI explains, detects, and fixes different types of coding errors. * To build confidence in using AI to perform structured debugging practices.   **Lab Outcomes (LOs):**  After completing this lab, students will be able to:   * Use AI tools to detect and correct syntax, logic, and runtime errors. * Interpret AI-suggested bug fixes and explanations. * Apply systematic debugging strategies supported by AI-generated insights. * Refactor buggy code using responsible and reliable programming patterns.   **Task Description#1**   * Paste a function with a missing colon (add(a, b)), and let AI fix the syntax error.     Prompt: fix the syntax error in python program  Code:    **Explanation**   * **Function definition**: def add(a, b): declares a function named add with two parameters, a and b. * **Return value**: return a + b computes and returns the sum of the two arguments. * **Function call**: print(add(1,2)) calls add with 1 and 2, then prints the result.   **Expected Output#1**   * Corrected function with syntax fix     **Task Description#2 (Loops)**   * Identify and fix a logic error in a loop that causes infinite iteration.     **Prompt:** **the following code runs infinitly.identify and fix the logical error**  **Code:**      **Explanation**   * **Function: count\_down(n) takes a starting integer n.** * **Loop condition: while n >= 0: keeps iterating as long as n is not negative.** * **Action per iteration: print(n) outputs the current value.** * **Progress toward termination: n -= 1 decrements n so the condition will eventually be false.** * **Example call: count\_down(3) prints 3, 2, 1, 0 and then stops.**   **Expected Output#2**   * AI fixes increment/decrement error   Output:    **Task Description#3**   * Debug a runtime error caused by division by zero. Let AI insert try-except.     **Prompt: Fix the runtime error and rewrite the code**  **Code:**    **Explanation**   * Function: divide(a, b) returns the result of a / b, but first checks for a zero denominator. * Zero-division guard: if b == 0: return "Error: division by zero" prevents a runtime ZeroDivisionError and returns a clear message instead. * Normal case: If b is not zero, it returns the floating-point division a / b.   Example calls in the file:   * print(divide(10, 0)) → prints Error: division by zero * print(divide(10, 2)) → prints 5.0   **Expected Output#3**   * Corrected function with safe error handling   Output:    **Task Description#4**   * Provide a faulty class definition (missing self in parameters). Let AI fix it     **Prompt:** fix the missing self parameter in the following python class and rewrite the code with example  **Code:**    **Explanation**   * Class: Rectangle models a rectangle with a length and width. * Constructor: \_\_init\_\_(self, length, width) saves the provided dimensions on the instance (self.length, self.width). * Method: area(self) returns the area by multiplying length \* width. * Example usage: * rect = Rectangle(5, 3) creates a rectangle 5 by 3. * Printing attributes shows its dimensions. * rect.area() computes 15.   **Expected Output#4**   * Correct \_\_init\_\_() method and explanation   Output:    **Task Description#5**   * Access an invalid list index and use AI to resolve the Index Error.     **Prompt:** fix the following python code that raises an index error and suggest checking length or using safe access logic  **Code:**    **Explanation**   * List and index: numbers = [1, 2, 3] and index = 5. Index 5 is out of range for this 3-item list. * Option 1 (length check): * if 0 <= index < len(numbers): ... else: ... * Only accesses numbers[index] when the index is valid; otherwise prints "Index out of range". * Option 2 (safe accessor): * get\_or\_default(seq, idx, default=None) returns seq[idx] if the index is valid; otherwise returns default. * print(get\_or\_default(numbers, index, default="N/A")) prints "N/A" for invalid index.   **Expected Output#5**   * AI suggests checking length or using safe access logic     **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** | | --- | --- | | Identification of bugs | 0.5 | | Application of AI-suggested fixes | 0.5 | | Explanation and understanding of errors | 0.5 | | Corrected code functionality | 0.5 | | Report structure and reflection | 0.5 | | **Total** | **2.5 Marks** | | | | | | | Week4 - Wednesday |  |