|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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** | | | Week3 - Wednesday | **Time(s)** | |  | | | |
| **Duration** | | | 2 Hours | **Applicableto**  **Batches** | |  | | | |
| **AssignmentNumber:6.3**(Present assignment number)/**24**(Total number of assignments) | | | | | | | | | |
|  | | | | | | | | | |
|  | **Q.No.** | **Question** | | | | | | ***ExpectedTime***  ***to complete*** |  |
|  | 1 | Lab 6: AI-Based Code Completion – Classes, Loops, and Conditionals  **Lab Objectives:**   * To explore AI-powered auto-completion features for core Python constructs. * To analyze how AI suggests logic for class definitions, loops, and conditionals. * To evaluate the completeness and correctness of code generated by AI assistants.   **Lab Outcomes (LOs):**  After completing this lab, students will be able to:   * Use AI tools to generate and complete class definitions and methods. * Understand and assess AI-suggested loops for iterative tasks. * Generate conditional statements through prompt-driven suggestions. * Critically evaluate AI-assisted code for correctness and clarity.   **Task Description#1 (Classes)**   * Use AI to complete a Student class with attributes and a method. * Check output * Analyze the code generated by AI tool   **Instructions**:   * **Initialize class with attributes like name, roll no, marks** * **Method to display student details** * **Method to calculate grade based on marks (A:>=90, B: >=75, C: >=60, else Fail)**   Start Writing code and auto complete using any AI tool  **Expected Output#1**   * Class with constructor and display\_details() method   CS Code:    **Output:**    **CS Code:**    **Output:**    **Task Description#2 (Loops)**   * Prompt AI to complete a function that prints the first 10 multiples of a number using a loop. * Analyze the generated code * Ask AI to generate code using other controlled looping   Write code using **For** Loop, later complete code using **While** Loop  **Expected Output#2**   * Correct loop-based implementation   VS Code:      **Output:**      **CS Code:**      **Output:**      **Task Description#3 (Conditional Statements)**   * Ask AI to write nested if-elif-else conditionals to classify age groups. * Analyze the generated code * Ask AI to generate code using other conditional statements   **Expected Output#3**   * Age classification function with appropriate conditions and with explanation   Explanation: The function classify\_age(age) is designed to categorize people into different age groups using **if-elif-else conditionals**.   1. **if age < 13:**    * If the input age is **less than 13**, the function returns "Child".    * Example: age = 10 → Child. 2. **elif age >= 13 and age <= 19:**    * If the first condition is false, it checks whether the age is between **13 and 19** (inclusive).    * If true, the function returns "Teenager".    * Example: age = 16 → Teenager. 3. **elif age >= 20 and age <= 59:**    * If neither of the above conditions are true, this block checks whether the age lies between **20 and 59**.    * If true, the function returns "Adult".    * Example: age = 35 → Adult. 4. **else:**    * If none of the above conditions are satisfied, it means the age is **60 or above**.    * The function returns "Senior".    * Example: age = 70 → Senior.   :  VS Code:      Output:      **CS Code:**      **Output:**      **Task Description#4 (For and While loops)**   * Generate a sum\_to\_n() function to calculate sum of first n numbers * Analyze the generated code * Get suggestions from AI with other controlled looping   **Expected Output#4**   * Python code with explanation   **Explanation**   * **For Loop Version (sum\_to\_n\_for)**   + **Uses range(1, n+1) to iterate through numbers from 1 to n.**   + **Each number is added to total.**   + **The final sum is returned.** * **While Loop Version (sum\_to\_n\_while)**   + **Starts with i = 1.**   + **Adds numbers to total one by one until i > n.**   + **Uses i += 1 to move to the next number.**   + **Returns the sum after loop ends.**   **Comparison**   * + **Both methods give the same result.**   + **The for loop is concise and best when we know the exact range.**   + **The while loop is flexible and better when the number of iterations is not predetermined.**       **Output:**      **CS Code:**      **Output:**      **Task Description#5 (Class)**   * Use AI to build a BankAccount class with deposit, withdraw, and balance methods. * Analyze the generated code * Add comments and explain code   **Instructions**   * **Initialize BankAccount class with attributes like name, balance** * **Method to deposit amount** * **Method to withdraw amount** * **Method to check balance**   **Expected Output#5**   * Python code with explanation   **Explanation:**   1. **Class Definition – BankAccount represents a simple bank account.** 2. **Constructor (\_\_init\_\_) – Initializes the account with name and an optional balance (default 0).** 3. **deposit(amount) – Adds funds to the account if the deposit amount is positive.** 4. **withdraw(amount) – Deducts money from the account only if there are enough funds. Prevents overdrawing.** 5. **check\_balance() – Prints the account holder’s name along with the current balance.** 6. **Testing – Demonstrates all operations: creating an account, depositing, withdrawing, and handling insufficient balance.**   VS Code:    **Output:**    **CS Code:**    **Output:**    **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** | | --- | --- | | Class | 1.0 | | Loops | 1.0 | | Conditional Statements | 0.5 | | **Total** | **2.5 Marks** | | | | | | | Week3 - Wednesday |  |