|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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** | | | | | | | ***ExpectedTi me***  ***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):** | | | | | | | Week3 - Wednesday |

|  |  |  |
| --- | --- | --- |
|  | 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 CODE I WROTE:   **CODE COMPLETION USING AI:**  **GITHUB COPILOT:**  **PROMPT: Complete this code and Initialize class with attributes like name, roll no, marks and a method to display student details and also method to calculate grade based on marks (A:>=90, B: >=75, C: >=60, else Fail) and finally add a display details method.**  **CODE:** |  |

|  |  |  |
| --- | --- | --- |
|  | **OUTPUT:**    **CURSOR AI:**  **PROMPT: complete this python program to manage multiple students using a class, where each student has a name, roll number, and marks. The program displays their details and calculates grades (A, B, C, or Fail) based on marks**  **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   **MY CODE:**    **CODE COMPLETION USING AI:**  **GITHUB COPILOT:**  **PROMPT: complete this code using while loop also take the number as a user input.** |  |

|  |  |  |
| --- | --- | --- |
|  | **CODE USING AI:**    **OUTPUT:**    **CURSOR:**  **Prompt:** complete the code **to generate a multiples for any number up to a user- defined limit that is 10**  **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 My code:   **CODE USING AI:**  **GITHUB COPILOT:**  **prompt : complete this code and add two more groups called adult ranging from 20 to 59 and senior for age 60 or above take the age as a user input**  **Code :**    **OUTPUT:** |  |

|  |  |  |
| --- | --- | --- |
|  | **CURSOR:**  **PROMPT: complete the code to classify a person’s age into categories (Child, Teenager, Adult, Senior) with validation for invalid inputs."**  **CODE:**    **OUTPUT:**    **EXPLANATION: This Python code defines a function age(n) that classifies a person based on their age:** |  |

|  |  |  |
| --- | --- | --- |
|  | * **If the input age is less than or equal to 0 or greater than 120, it prints "Invalid age".** * **If the age is 0–13, it prints "Child".** * **If the age is 14–19, it prints "Teenager".** * **If the age is 20–59, it prints "Adult".** * **If the age is 60 or above, it prints "Senior".**   **Finally, it asks the user to enter their age (input()), converts it into an integer, and passes it to the age() function for classification.**  **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 MY CODE:   **CODE USING AI:**  **GITHUB COPILOT:**  **Prompt: complete this code where the function is returning the sum of first n natural numbers using controlled looping and take the n as user input**  **CODE:** |  |

|  |  |  |
| --- | --- | --- |
|  | **OUTPUT:**    **CURSOR:**  **PROMPT: Complete the code to calculate the sum of the first n natural numbers using the formula n(n+1)/2 with input validation.**  **CODE AND OUTPUT:** |  |

|  |  |  |
| --- | --- | --- |
|  | **Explanation:**   1. **Function sum\_to\_n(n)**    * **Starts with total = 0.**    * **Uses a for loop from 1 to n (inclusive).**    * **Adds each number i to total.**    * **Returns the final sum.** 2. **User Input**    * **The program asks the user to enter a positive integer (num).** 3. **Function Call**    * **Calls sum\_to\_n(num) and stores the result in result.** 4. **Output**    * **Prints:**    * **Sum of first {num} natural numbers is {result}**   **Example: If you enter 5, it calculates 1+2+3+4+5 = 15 and prints: Sum of first 5 natural numbers is 15.**  **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   MY CODE: |  |

|  |  |  |
| --- | --- | --- |
|  | CODE USING AI:  GITHUB COPILOT:  PROMPT: complete this code **with attributes like name, balance and Method to deposit amount, Method to withdraw amount, Method to check balance and take user input.**  **CODE:** |  |

|  |  |  |
| --- | --- | --- |
|  | OUTPUT:    CURSOR:  PROMPT: Complete the program to simulate a simple bank account system with deposit, withdrawal, and balance check options using OOP and a menu-driven interface.  CODE: |  |

|  |  |  |
| --- | --- | --- |
|  | OUTPUT: |  |

|  |  |  |
| --- | --- | --- |
|  | **Explanation:**   1. **Class Definition (BankAccount)**    * init : Initializes account holder’s name and balance (default is 0).    * deposit(amount): Adds money to the balance if the amount is positive.    * withdraw(amount): Deducts money if the amount is positive and there are enough funds.    * check\_balance(): Displays the account holder’s name and current balance. 2. **Account Creation** |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | * The user enters their name and initial balance, which creates a BankAccount object.  1. **Menu-Driven Loop**    * The program runs continuously with 4 options:      1. Deposit money      2. Withdraw money      3. Check balance      4. Exit    * Based on the user’s choice, the appropriate method is called   **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** |