|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE** | | | | | **DEPARTMENT OF COMPUTER SCIENCE ENGINEERING** | | | | |
| **Program Name:** M. Tech/MCA | | | | **Assignment Type: Lab** | | | **AcademicYear:**2025-2026 | | |
| **Course Coordinator Name** | | | | Venkataramana Veeramsetty | | | | | |
| **Course Code** | | |  | **Course Title** | | AI Assisted Problem Solving Using Python | | | |
| **Year/Sem** | | | I/I | **Regulation** | | R24 | | | |
| **Date and Day**  **of Assignment** | | | Week1 - Monday | **Time(s)** | |  | | | |
| **Duration** | | | 2 Hours | **Applicable to**  **Batches** | | M. Tech/MCA | | | |
| **AssignmentNumber:1.3**(Present assignment number)/**24**(Total number of assignments) | | | | | | | | | |
|  | | | | | | | | | |
|  | **Q.No.** | **Question** | | | | | | ***Expected Time***  ***to complete*** |  |
|  | 1 | Lab 1: Environment Setup – GitHub Copilot and VS Code Integration  **Lab Objectives:**   * To install and configure GitHub Copilot in Visual Studio Code. * To explore AI-assisted code generation using GitHub Copilot. * To analyze the accuracy and effectiveness of Copilot's code suggestions. * To understand prompt-based programming using comments and code context   **Lab Outcomes (LOs):**  After completing this lab, students will be able to:   * Set up GitHub Copilot in VS Code successfully. * Use inline comments and context to generate code with Copilot. * Evaluate AI-generated code for correctness and readability. * Compare code suggestions based on different prompts and programming styles.   **Task Description#1**   * Install and configure GitHub Copilot in VS Code. Take screenshots of each step.   **Installed and Configured GitHub Copilot**  **Screenshot of Copilot signed in (Command Palette):**  **Expected Output#1**      **Steps Performed:**  **1. Opened Extensions panel → Searched "GitHub Copilot" → Installed the official extension.**  **2. Used Command Palette → Typed "Copilot" → Selected "GitHub Copilot: Sign in" → Authorized via GitHub.**  **3. Copilot is now active (confirmed by options like "Change Completions Model", "Open Completions Panel", etc.).**  **Proof: Screenshot of Command Palette showing multiple Copilot commands available → Confirms successful installation and authentication.**   * Install and configure GitHub Copilot in VS Code. Take screenshots of each step.   **Task Description#2**   * Use Copilot to generate a is\_prime() Python function**.**   **Prompt :** generate a is\_prime() Python function**.**  **Expected Output#2**      **Task Description#3**   * Write a comment like # Function to reverse a string and use Copilot to generate the function. * Promt : Function to reverse a string and use Copilot to generate the function.   **Expected Output#3**    This function:   * Takes a string input parameter named text * Uses Python's string slicing with a step of -1 ([::-1]) to reverse the string * Returns the reversed string * Auto-completed reverse function     **Output:**  **Task Description#4**   * Generate both recursive and iterative versions of a factorial function using comments.. * Promt : Generate both recursive and iterative versions of a factorial function using comments   **Expected Output#4**  **# Iterative factorial using loop**   * *Iterative factorial implementations*   **This implementation: Takes an integer input n Checks if the number is negative (factorial is not defined for negative numbers)**  **Uses a loop to multiply numbers from 1 to n Returns the factorial result**  **The function will: Return 1 for n = 0 (0! = 1 by definition) Calculate factorial for positive integers Raise an error for negative numbers**  **# Recursive factorial**  **This implementation:**  **Takes an integer input n**  **Has base cases for n = 0 and n = 1 (both return 1)**  **Uses recursion for other cases by calling itself with n-1**  **Includes error checking for negative numbers**  **The recursive solution works by breaking down the factorial calculation into smaller subproblems:**  **5! = 5 \* 4!**  **4! = 4 \* 3!**  **3! = 3 \* 2!**  **2! = 2 \* 1!**  **1! = 1 (base case)**          **Task Description#5**   * Use Copilot to find the largest number in a list. Assess code quality and efficiency.   **Expected Output#5**  **Promt :** find the largest number in a list. Assess code quality and efficiency   * A valid function with your review     **Find Largest Number (logic)**   1. **Check** if list is empty → **Error** if yes 2. **Start** with first number as "largest" 3. **Loop** through every number 4. **If** current > largest → **Update** largest 5. **Return** largest at end       **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** | | --- | --- | | Successful Setup of Copilot (Task #1) | 2 | | is\_prime() Python function (Task #2) | 2 | | Reverse a string function (Task #3) | 2 | | Factorial Function (Task #4) | 2 | | Find the largest number (Task #5) | 2 | | **Total** | **10 Marks** | | | | | | | Week1 - Wednesday |  |