|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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 - TUESDAY | **Time(s)** | |  | | | |
| **Duration** | | | 2 Hours | **Applicable to**  **Batches** | | M. Tech/MCA | | | |
| **AssignmentNumber:2.3**(Present assignment number)/**24**(Total number of assignments) | | | | | | | | | |
|  | | | | | | | | | |
|  | **Q.No.** | **Question** | | | | | | ***Expected Time***  ***to complete*** |  |
|  | 1 | Lab 2: Exploring Additional AI Coding Tools – Gemini (Colab) and Cursor AI  **Lab Objectives:**   * To explore and evaluate the functionality of Google Gemini for AI-assisted coding within Google Colab. * To understand and use Cursor AI for code generation, explanation, and refactoring. * To compare outputs and usability between Gemini, GitHub Copilot, and Cursor AI. * To perform code optimization and documentation using AI tools.   **Lab Outcomes (LOs):**  After completing this lab, students will be able to:   * Generate Python code using Google Gemini in Google Colab. * Analyze the effectiveness of code explanations and suggestions by Gemini. * Set up and use Cursor AI for AI-powered coding assistance. * Evaluate and refactor code using Cursor AI features. * Compare AI tool behavior and code quality across different platforms.   **Task Description#1**   * Use Google Gemini in Colab to write a function that reads a CSV file and calculates mean, min, max   **Input :**    **Expected Output#1**   * Functional code with output and screenshot   **OUTPUT :**    **Task Description#2**   * Compare Gemini and Copilot outputs for a palindrome check function   **CODE FROM GEMINI**    **CODE FROM COPLOT:**    **Expected Output#2**    side - by-side comparison and observations   * Copilot is Microsoft AI assistant . it works really well with Microsoft like Word , excel * Gemini is google ‘s AI assistant . * Gemini works best with google tools like Doc, Gmail   **Task Description#3**   * Ask Gemini to explain a Python function (to calculate area of various shapes) line by line..   **INPUT:**  import math  # Function to calculate the area of a rectangle  def area\_rectangle(length, width):      """      Calculates the area of a rectangle.      Args:          length (float): The length of the rectangle.          width (float): The width of the rectangle.      Returns:          float: The calculated area.      """      if length <= 0 or width <= 0:          return "Error: Length and width must be positive."      return length \* width  # Function to calculate the area of a circle  def area\_ circle(radius):      """      Calculates the area of a circle.      Args: Sum of even numbers: 30  Sum of odd numbers: 25          radius (float): The radius of the circle.      Returns:          float: The calculated area.      """      if radius <= 0:          return "Error: Radius must be positive."      # Area = pi \* r^2      return math .pi \* (radius \*\* 2)  # Function to calculate the area of a triangle (using base and height)  def area \_triangle (base, height):      """      Calculates the area of a triangle using base and height.      Args :          base (float): The base length of the triangle.          height (fl# Sample list of numbers  numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]  # Initialize sums  Sum\_even = 0  sum\_odd = 0  # Loop through each number in the list  for num in numbers:      if num % 2 == 0:          sum\_even += num      else:          sum\_odd += num  # Print the results  print("Sum of even numbers:", sum\_even)  print("Sum of odd numbers:", sum\_odd)  voat): The height of the triangle.      Returns:          float: The calculated area.      """      if base <= 0 or height <= 0:          return "Error: Base and height must be positive."      # Area = 0.5 \* base \* height      return 0.5 \* base \* height  # --- Main execution block to demonstrate usage ---  def main():      print("--- Area Calculations ---")      # 1. Rectangle Example      rect\_length = 10.5      rect\_width = 5.0      rect\_area = area\_rectangle(rect\_length, rect\_width)      print(f"\nRectangle (Length: {rect\_length}, Width: {rect\_width})")      print(f"Area: {rect\_area}")      # 2. Circle Example      circle\_radius = 7.0      circle\_area = area\_circle(circle\_radius)      # Using ':.2f' to format output for better readability      print(f"\nCircle (Radius: {circle\_radius})")      print(f"Area: {circle\_area:.2f}")      # 3. Triangle Example      tri\_base = 8.0      tri\_height = 4.0      tri\_area = area\_triangle(tri\_base, tri\_height)      print(f"\nTriangle (Base: {tri\_base}, Height: {tri\_height})")      print(f"Area: {tri\_area}")      # Example of error handling      print("\nError Example:")      error\_area = area\_circle(-5)      print(f"Circle (Radius: -5): {error\_area}")  # Ensures the main function runs when the script is executed directly  if \_\_name\_\_ == "\_\_main\_\_":      main()  **OUTPUT:**  -- Area Calculations ---  Rectangle (Length: 10.5, Width: 5.0)  Area: 52.5  Circle (Radius: 7.0)  Area: 153.94  Triangle (Base: 8.0, Height: 4.0)  Area: 16.0  Error Example:  Circle (Radius: -5): Error: Radius must be positive.  === Code execution complete ===  --- Area Calculations ---  Rectangle (Length: 10.5, Width: 5.0)  Area: 52.5  Circle (Radius: 7.0)  Area: 153.94  Triangle (Base: 8.0, Height: 4.0)  Area: 16.0  Error Example:  Circle (Radius: -5): Error: Radius must be positive.  === Code execution complete ===  **Expected Output#3**   * Detailed explanation with code snippet  1. **. The main goal is to find the area of shapes like circle and rectangle** 2. **It uses separate function for each shape , which keeps the code clean and easy to read** 3. **Each function uses the correct geometric formula and checks that is valid or not** 4. **It imports the math library specifically o get the accurate value of Pi for the circle** 5. **Finally, program has runed and showing the results**     **Task Description#4**   * Install and configure Cursor AI. Use it to generate a Python function (e.g., sum of squares     .I**NPUT COD**E :    **Expected Output#4**   * Screenshots of working environments with few prompts to generate python code   **OUTPUT:**    **Task Description#5**   * Student need to write code to calculate sum of add number and even numbers in the list     **Expected Output#5**  Refactored code written by student with improved logic  **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 Use of Gemini in Colab (Task#1 & #2) | 2.5 | | Code Explanation Accuracy (Gemini) (Task#3) | 2.5 | | Cursor AI Setup and Usage (Task#4) | 2.5 | | Refactoring and Improvement Analysis (Task#5) | 2.5 | | **Total** | **10 Marks** | | | | | | | Week1 - TuesDay |  |