|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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.   **Expected Output#1**   * Functional code with output and screenshot   Prompt: Write a Python function to read a CSV file and calculate the mean, minimum, and maximum values for each numeric column.  **Code:**    **Output:**  **WhatsApp Image 2025-10-30 at 18.16.02_6e368ab7**  **Task Description#2**   * Compare Gemini and Copilot outputs for a palindrome check function.   Prompt:  Write a Python function that checks if a word is a palindrome.  Gemini code:    Copilot code:    **Expected Output#2**   * Side-by-side comparison and observations  | **Feature** | **Gemini Output** | **Copilot Output** | | --- | --- | --- | | Readability | ⭐⭐⭐⭐☆ | ⭐⭐⭐ | | Flexibility (parameters) | ⭐⭐☆ | ⭐⭐⭐⭐ | | Documentation | ⭐⭐⭐⭐⭐ | ⭐⭐ | | Educational clarity | ⭐⭐⭐⭐⭐ | ⭐⭐⭐ | | Code efficiency | ⭐⭐⭐ | ⭐⭐⭐⭐ | | Example outputs | Multiple formatted examples | Simple user input |     **Overall Observations:**   * Both programs correctly handle case insensitivity and ignore punctuation. * Differences in results mainly come from using **different test strings** (e.g., “Race Car” vs “race a car”). * Both outputs confirm that the palindrome-checking logic is working properly in each script. * The second version (PowerShell output) shows simpler formatting (no “Is … ?” structure), while the first uses a question–answer style output.   **Task Description#3**   * Ask Gemini to explain a Python function (to calculate area of various shapes) line by   line..  **Prompt:**Write some code in Python of various shapes to find the area of the shapes like square, rectangle, triangle., circle From my user inputs.          **Expected Output#3**   * Detailed explanation with code snippet     **Code snippet:** **Imports and Function Definitions** **import math**  Imports Python’s **math module**, which gives access to mathematical constants and functions (like math.pi).  **def calculate\_square\_area(side)**:  """Calculate area of a square."""  return side \* side  Defines a function named calculate\_square\_area that takes one parameter: side.  The docstring ("""...""") briefly explains what the function does.  Returns the area of a square using the formula side × side.  **def calculate\_rectangle\_area(length, width)**:  """Calculate area of a rectangle."""  return length \* width  Defines another function for a rectangle’s area.  Takes two parameters: length and width.  Returns length × width.  **def calculate\_triangle\_area(base, height)**:  """Calculate area of a triangle."""  return 0.5 \* base \* height  Calculates a triangle’s area using the formula ½ × base × height.  **def calculate\_circle\_area(radius)**:  """Calculate area of a circle."""  return math.pi \* radius \* radius  Calculates a circle’s area using π × r².  Uses math.pi for a precise value of π.  **def main():**  **while True:**  Defines the main function main().  Starts an infinite loop (while True:), meaning it keeps running until break is used.  **print("\nArea Calculator")**  **print("1. Square")**  **print("2. Rectangle")**  **print("3. Triangle")**  **print("4. Circle")**  **print("5. Exit")**  Prints a simple menu to the user.  \n adds a blank line before the menu for readability.  **choice = input("Enter your choice (1-5): ")**  **if choice == '5':**  **print("Goodbye!")**  **break**  If the user enters '5', the program prints “Goodbye!” and breaks the loop.  break stops the infinite loop and exits the program.  **if choice == '1':**  **side = float(input("Enter the side length: "))**  **area = calculate\_square\_area(side)**  **print(f"Area of square = {area:.2f}")**  If the user chooses option 1:  Prompts for the side length and converts it to a **float**.  Calls calculate\_square\_area(side) and stores the result in area.  Prints the area formatted to **2 decimal places** ({area:.2f}).  **elif choice == '2':**  **length = float(input("Enter the length: "))**  **width = float(input("Enter the width: "))**  **area = calculate\_rectangle\_area(length, width)**  **print(f"Area of rectangle = {area:.2f}")**  Similar logic:Takes two inputs: length and width.  Calls the rectangle function.  Prints the result formatted to 2 decimal places.  **elif choice == '3':**  **base = float(input("Enter the base: "))**  **height = float(input("Enter the height: "))**  **area = calculate\_triangle\_area(base, height)**  **print(f"Area of triangle = {area:.2f}")**  Asks for base and height.  Calls the triangle area function.  Displays the area rounded to 2 decimal places.  **elif choice == '4':**  **radius = float(input("Enter the radius: "))**  **area = calculate\_circle\_area(radius)**  **print(f"Area of circle = {area:.2f}")**  Takes the circle’s radius.  Calls the circle area function.  **else:**  **print("Invalid choice! Please select 1-5")**  If the user enters anything not in 1–5, this message appears.  python  **if \_\_name\_\_ == "\_\_main\_\_":**  **main()**  This is a Python convention that ensures main() runs only if the script is executed directly (not imported as a module).  **Task Description#4**   * Install and configure Cursor AI. Use it to generate a Python function (e.g., sum of squares).           **Expected Output#4**   * Screenshots of working environments with few prompts to generate python code   # Sample program:    **Task Description#5**   * Student need to write code to calculate sum of add number and even numbers in the list   Prompt: Write a Python program to calculate the sum of odd numbers and even numbers in a list and print the results separately.  Code:    **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 |  |