ASSIGNMENT-2.3

---------------------------------------

**NAME: PRADEEP GUPTHA**

**HALL-TICKET: 2403A510C7**

**BATCH NO: 05**

Task 1:   
● Description:  
 Use Google Gemini in Colab to write a function that reads a CSV file and calculates  
mean, min, max.

Prompt:

Write a function that reads a CSV file and calculates mean, min, max.

CODE SCREENSHOT:

A screen shot of a computer code

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

OUTPUT:

A black background with white text

AI-generated content may be incorrect.

EXPLANATION:

The function **reads a CSV file** using the pandas library.

It calculates **mean**, **minimum**, and **maximum** for each **numeric column**.

It displays the original CSV data and the results.

--------------------------------------------------------------------------------------------------------------

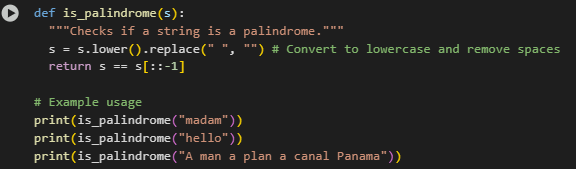
Task 2:

● Compare Gemini and Copilot outputs for a palindrome check function.

Prompt: Give a python function to check whether the given function is palindrome or not.

CODE SCREENSHOT:

BY GEMINI:



Output:

True

False

True

BY COPILOT:

A computer screen with text

AI-generated content may be incorrect.

Output:

True

False

COMPARISION:

| **Aspect** | **Gemini** | **Copilot** |
| --- | --- | --- |
| **Function Name** | is\_palindrome(s) | is\_palindrome(s) |
| **Docstring / Comments** | Has a short docstring: """Checks if a string is a palindrome.""" | No docstring, but has inline comments explaining steps |
| **Preprocessing Order** | Converts to lowercase **first**, then removes spaces | Removes spaces **first**, then converts to lowercase |
| **Method Used** | s.lower().replace(" ", "") | s.replace(" ", "").lower() |
| **Comparison Logic** | return s == s[::-1] | return s == s[::-1] |
| **Examples Given** | - "madam" → True - "hello" → False - "A man a plan a canal Panama" → True | - "Madam" → True - "Hello World" → False |
| **Number of Examples** | 3 examples | 2 examples |
| **Handling Case & Spaces** | Yes (by converting to lowercase and removing spaces) | Yes (by converting to lowercase and removing spaces) |
| **Clarity** | More compact, fewer comments | More descriptive comments, better for beginners |

----------------------------------------------------------------------------------------------------------

Task 3:

● Ask Gemini to explain a Python function (to calculate area of various shapes) line by  
line.

PROMPT:

Generate a Python program that calculates the area of various shapes .

CODE:

**import math**

**def calculate\_square\_area(side):**

**"""Calculates the area of a square."""**

**return side \* side**

**def calculate\_rectangle\_area(length, width):**

**"""Calculates the area of a rectangle."""**

**return length \* width**

**def calculate\_circle\_area(radius):**

**"""Calculates the area of a circle."""**

**return math.pi \* radius\*\*2**

**def calculate\_triangle\_area(base, height):**

**"""Calculates the area of a triangle."""**

**return 0.5 \* base \* height**

**print("Welcome to the Area Calculator!")**

**print("Please choose a shape from the options below:")**

**print("1. Square")**

**print("2. Rectangle")**

**print("3. Circle")**

**print("4. Triangle")**

**shape\_choice = input("Enter the number corresponding to your chosen shape: ")**

**dimensions = {}**

**calculated\_area = None**

**if shape\_choice == '1':**

**try:**

**side = float(input("Enter the side length of the square: "))**

**if side >= 0:**

**dimensions['side'] = side**

**calculated\_area = calculate\_square\_area(dimensions['side'])**

**print(f"The area of the square is: {calculated\_area}")**

**else:**

**print("Error: Side length cannot be negative.")**

**except ValueError:**

**print("Error: Invalid input. Please enter a numeric value for the side length.")**

**elif shape\_choice == '2':**

**try:**

**length = float(input("Enter the length of the rectangle: "))**

**width = float(input("Enter the width of the rectangle: "))**

**if length >= 0 and width >= 0:**

**dimensions['length'] = length**

**dimensions['width'] = width**

**calculated\_area = calculate\_rectangle\_area(dimensions['length'], dimensions['width'])**

**print(f"The area of the rectangle is: {calculated\_area}")**

**else:**

**print("Error: Length and width cannot be negative.")**

**except ValueError:**

**print("Error: Invalid input. Please enter numeric values for length and width.")**

**elif shape\_choice == '3':**

**try:**

**radius = float(input("Enter the radius of the circle: "))**

**if radius >= 0:**

**dimensions['radius'] = radius**

**calculated\_area = calculate\_circle\_area(dimensions['radius'])**

**print(f"The area of the circle is: {calculated\_area}")**

**else:**

**print("Error: Radius cannot be negative.")**

**except ValueError:**

**print("Error: Invalid input. Please enter a numeric value for the radius.")**

**elif shape\_choice == '4':**

**try:**

**base = float(input("Enter the base length of the triangle: "))**

**height = float(input("Enter the height of the triangle: "))**

**if base >= 0 and height >= 0:**

**dimensions['base'] = base**

**dimensions['height'] = height**

**calculated\_area = calculate\_triangle\_area(dimensions['base'], dimensions['height'])**

**print(f"The area of the triangle is: {calculated\_area}")**

**else:**

**print("Error: Base and height cannot be negative.")**

**except ValueError:**

**print("Error: Invalid input. Please enter numeric values for base and height.")**

**else:**

**print("Invalid shape choice.")**

**print("No valid shape was chosen to calculate the area.")**

**OUTPUT:**

**Welcome to the Area Calculator!**

**Please choose a shape from the options below:**

**1. Square**

**2. Rectangle**

**3. Circle**

**4. Triangle**

**Enter the number corresponding to your chosen shape: 1**

**Enter the side length of the square: 5**

**The area of the square is: 25.0**

EXPLANATION:

This code does the following:

1. **Imports the math module**: This is needed to use math.pi for circle area calculations.
2. **Defines area calculation functions**: It creates four functions:
   * calculate\_square\_area(side): Calculates the area of a square.
   * calculate\_rectangle\_area(length, width): Calculates the area of a rectangle.
   * calculate\_circle\_area(radius): Calculates the area of a circle.
   * calculate\_triangle\_area(base, height): Calculates the area of a triangle.
3. **Provides a user interface**:
   * It welcomes the user and lists the available shapes with corresponding numbers.
   * It prompts the user to enter the number for their chosen shape.
4. **Collects dimensions and calculates area with error handling**:
   * Based on the user's shape\_choice, it prompts for the necessary dimensions (side, length and width, radius, or base and height).
   * It uses a try-except block to catch ValueError if the user enters non-numeric input.
   * It checks if the entered dimensions are non-negative.
   * If the input is valid, it calls the appropriate area calculation function and prints the result.
   * If the input is invalid (non-numeric or negative), it prints an informative error message.
5. **Handles invalid shape choice**: If the user enters a number that doesn't correspond to any of the shapes, it prints an "Invalid shape choice" message.

In essence, the code guides the user to select a shape, get the required measurements, and then calculates and displays the area while handling potential input errors.

------------------------------------------------------------------------------------------------------------

TASK 4:

● Install and configure Cursor AI. Use it to generate a Python function (e.g., sum of  
squares)

PROMPT:

Generate a python code to calculate factorial of a given number .

CODES:

A screenshot of a computer

AI-generated content may be incorrect.

OUTPUT:

A black background with white letters and numbers

AI-generated content may be incorrect.

EXPLANATION:

**What the code does:**

**Function Definition (factorial):**

* Takes a number n as input
* Calculates the factorial of that number (n!)
* Factorial means: n × (n-1) × (n-2) × ... × 1
* For example: 5! = 5 × 4 × 3 × 2 × 1 = 120

**How it works:**

1. **Input validation**: Checks if the input is negative (factorials aren't defined for negative numbers)
2. **Calculation**: Uses a loop to multiply all numbers from 1 to n together
3. **Returns**: The factorial result or an error message for negative numbers

**Example usage:**

* The program asks the user to input a number
* It then calculates and displays the factorial of that number

**Example outputs:**

* Input: 5 → Output: "The factorial of 5 is 120"
* Input: 3 → Output: "The factorial of 3 is 6"
* Input: -1 → Output: "The factorial of -1 is Factorial is not defined for negative numbers"

This is a basic mathematical utility program that demonstrates loops, conditional statements, and function definitions in Python.

--------------------------------------------------------------------------------------------------------------

TASK 5:

● Student need to write code to calculate sum of odd number and even numbers in the list

CODE :

A computer screen shot of numbers and equations

AI-generated content may be incorrect.

Output:



Explanation:

The program defines a function sum\_odd\_even(numbers) that loops through each element of a list. It checks whether each number is even or odd using the modulo operator %. Odd numbers are added to sum\_odds and even numbers to sum\_evens. Finally, it returns both sums. Example: For [1,2,3,4,5], odd sum = 9, even sum = 6.