**My\_Calculator:**

// Creating a calculator class that handles basic and advanced mathematical operations

CLASS Calculator

// Initialising the calculator with empty input and history

FUNCTION \_\_init\_\_():

SET calculation TO empty string

SET history TO empty list

ENDFUNCTION

// Adding the trigonometric function

FUNCTION add\_symbols(symbol):

IF symbol IN ['cos', 'sin', 'tan', 'asin', 'acos', 'atan'] THEN

APPEND 'math.' + symbol + '(' TO calculation

ELSE

APPEND symbol (as string) TO calculation

ENDIF

RETURN calculation

ENDFUNCTION

//Evaluating current mathematical expressions and storing the results

FUNCTION evaluate\_calculation():

TRY

SET result TO EVALUATE calculation AS mathematical expression

APPEND {"expression": calculation, "result": result} TO history

SET calculation TO result

RETURN result

CATCH Exception

SET calculation TO empty string

RETURN "Error"

ENDTRY

ENDFUNCTION

//Creating a function for squaring the numbers

FUNCTION square():

IF calculation IS NOT EMPTY THEN

APPEND '\*\*2' TO calculation

ENDIF

RETURN calculation

ENDFUNCTION

//Creating a function for powering the numbers

FUNCTION power():

IF calculation IS NOT EMPTY THEN

APPEND '\*\*' TO calculation

ENDIF

RETURN calculation

ENDFUNCTION

//Creating a function that multiplies the input by pi

FUNCTION pi():

TRY

IF calculation IS EMPTY THEN

RETURN "Error"

ENDIF

SET result TO (CONVERT calculation TO FLOAT) \* π

APPEND {"expression": calculation, "result π": result} TO history

SET calculation TO result

RETURN result

CATCH Exception

RETURN "Error"

ENDTRY

ENDFUNCTION

//Creating a function that square roots the numbers and printing error for negative numbers

FUNCTION square\_root():

TRY

IF calculation IS EMPTY THEN

RETURN "Error"

ENDIF

SET num TO FLOAT VALUE OF calculation

IF num < 0 THEN

RETURN "Error"

ENDIF

SET result TO SQUARE ROOT OF num

APPEND {"expression": calculation, "result √": result} TO history

SET calculation TO result

RETURN result

CATCH Exception

RETURN "Error"

ENDFUNCTION

// Returns the list of past calculations

FUNCTION clear():

SET calculation TO empty string

RETURN calculation

ENDFUNCTION

FUNCTION all\_history():

RETURN history

ENDFUNCTION

ENDCLASS

**Calculator\_App:**

// GUI class for handling the calculator interface and user interactions

CLASS CalculatorGUI

//Creating a function that

FUNCTION \_\_init\_\_():

BEGIN

SET calculator TO NEW Calculator() // Create an instance of Calculator

SET root TO NEW Tkinter Window // Initialize the main GUI window

SET root.title TO "Scientific Calculator" // Set the window title

SET root.geometry TO "450x550" // Define window size

SET text\_result TO NEW TextField(height=3, width=15, font="Arial, 30")

PLACE text\_result ON GRID with columnspan=10

// Generating a calculator buttons

CALL create\_buttons()

ENDFUNCTION

// Initialising the GUI components and layout

FUNCTION create\_buttons():

// Creating a list of button labels and their positions

SET buttons TO [

("1", 2, 1), ("2", 2, 2), ("3", 2, 3),

("4", 3, 1), ("5", 3, 2), ("6", 3, 3),

("7", 4, 1), ("8", 4, 2), ("9", 4, 3),

("0", 5, 2), ("+", 2, 4), ("-", 3, 4),

("\*", 4, 4), ("/", 5, 4), ("(", 5, 1),

(")", 5, 3), ("C", 8, 1, 2), ("=", 8, 3, 2),

("^", 1, 1), ("π", 1, 2), ("²", 1, 3), ("√", 1, 4),

("Sin", 6, 1), ("Cos", 6, 2), ("Tan", 6, 3), (".", 6, 4),

("asin", 7, 1), ("ascos", 7, 2), ("astan", 7, 3)

]

FOR EACH (btn\_text, row, col, \*span) IN buttons DO

IF btn\_text EQUALS "C" THEN

SET action TO clear\_field

ELSEIF btn\_text EQUALS "=" THEN

SET action TO evaluate\_calculation

ELSEIF btn\_text EQUALS "^" THEN

SET action TO power

ELSEIF btn\_text EQUALS "π" THEN

SET action TO pi

ELSEIF btn\_text EQUALS "²" THEN

SET action TO square

ELSEIF btn\_text EQUALS "√" THEN

SET action TO square\_root

ELSEIF btn\_text IN ["Sin", "Cos", "Tan", "asin", "ascos", "astan"] THEN

SET action TO FUNCTION() CALL add\_symbols(LOWERCASE(btn\_text))

ELSE

SET action TO FUNCTION() CALL add\_symbols(btn\_text)

ENDIF

CREATE NEW BUTTON WITH text=btn\_text, command=action, width=5, font="Arial, 14"

PLACE button ON GRID AT (row, col) WITH columnspan=span[0] IF span EXISTS ELSE 1

NEXT

//Using the for loop Adjust grid row and column weights for resizing

FOR r FROM 1 TO 8 DO

CONFIGURE grid row r WITH weight=1

NEXT

FOR c FROM 1 TO 4 DO

CONFIGURE grid column c WITH weight=1

NEXT

ENDFUNCTION

// Creating a function that adds a symbol to the current input

FUNCTION add\_symbols(symbol):

SET result TO calculator.add\_symbols(symbol) // Add the symbol to the calculation

CALL update\_display(result) // Update the display

ENDFUNCTION

// Evaluates the current input and shows the result

FUNCTION evaluate\_calculation():

SET result TO calculator.evaluate\_calculation()

CALL update\_display(result) // Display the result

ENDFUNCTION

// Creating a function that squares the input

FUNCTION square():

SET result TO calculator.square()

CALL update\_display(result)

ENDFUNCTION

// Creating a function that powers the input

FUNCTION power():

SET result TO calculator.power() // Adds exponentiation operator

CALL update\_display(result)

ENDFUNCTION

// Creating a function that multiplies the input by π

FUNCTION pi():

SET result TO calculator.pi()

CALL update\_display(result)

ENDFUNCTION

// Creating a function that calculates the square root of the current input

FUNCTION square\_root():

SET result TO calculator.square\_root() // Computes square root

CALL update\_display(result)

ENDFUNCTION

// Creating a function that clears the display and resets the input

FUNCTION clear\_field():

SET result TO calculator.clear() // Clear the current input

CALL update\_display(result)

ENDFUNCTION

//Creating a function that updates the GUI display with new input or result

FUNCTION update\_display(text):

CLEAR text\_result

INSERT text INTO text\_result

SCROLL text\_result TO END

ENDFUNCTION

// Starting the Calculator GUI if the program is run directly

FUNCTION run():

START Tkinter event loop

ENDFUNCTION

// Running the calculator GUI

ENDCLASS

END

IF \_\_name\_\_ EQUALS "\_\_main\_\_" THEN

SET calc TO NEW CalculatorGUI()

CALL calc.run()

ENDIF