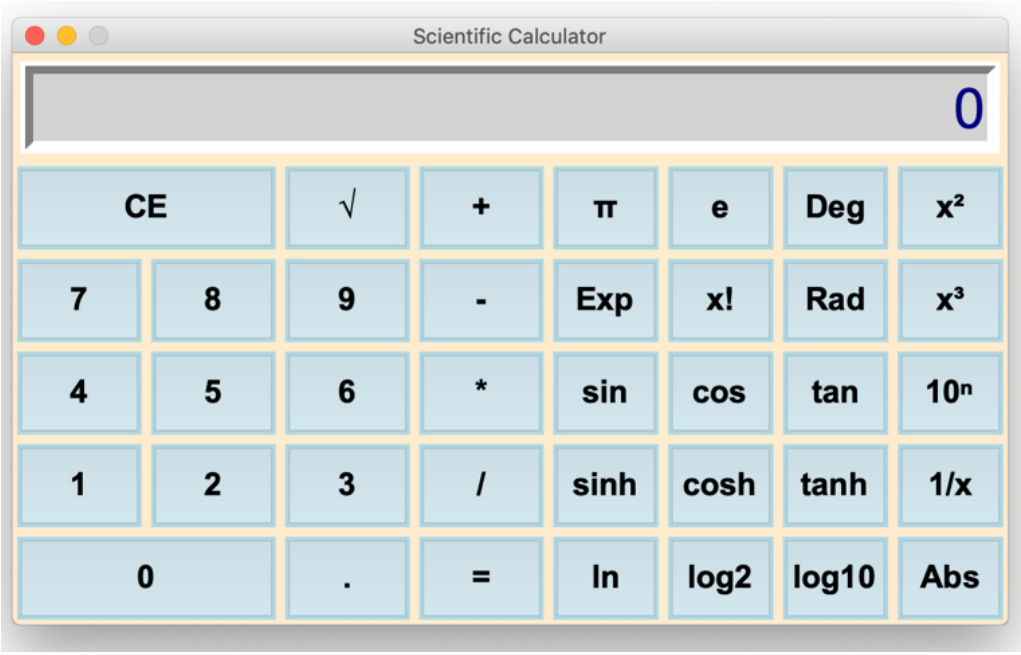
Georgia Marie Vrana
Thursday, May 16, 2024
ICS4UVa-01
Mr. James Wright

CULMINATING TASK PROPOSAL

The final programming project will be a scientific calculator developed using Python's Tkinter library. This calculator will emulate the functionality and appearance of a typical scientific calculator, offering both basic arithmetic operations (addition, subtraction, multiplication, division) and advanced mathematical functions such as trigonometry (sine, cosine, tangent and their inverses'), logarithms, exponentiation, factorials, and more. The user interface will be intuitive and user-friendly, allowing users to input expressions and view results directly within the GUI. The aim is to provide a comprehensive tool suitable for educational and professional use.

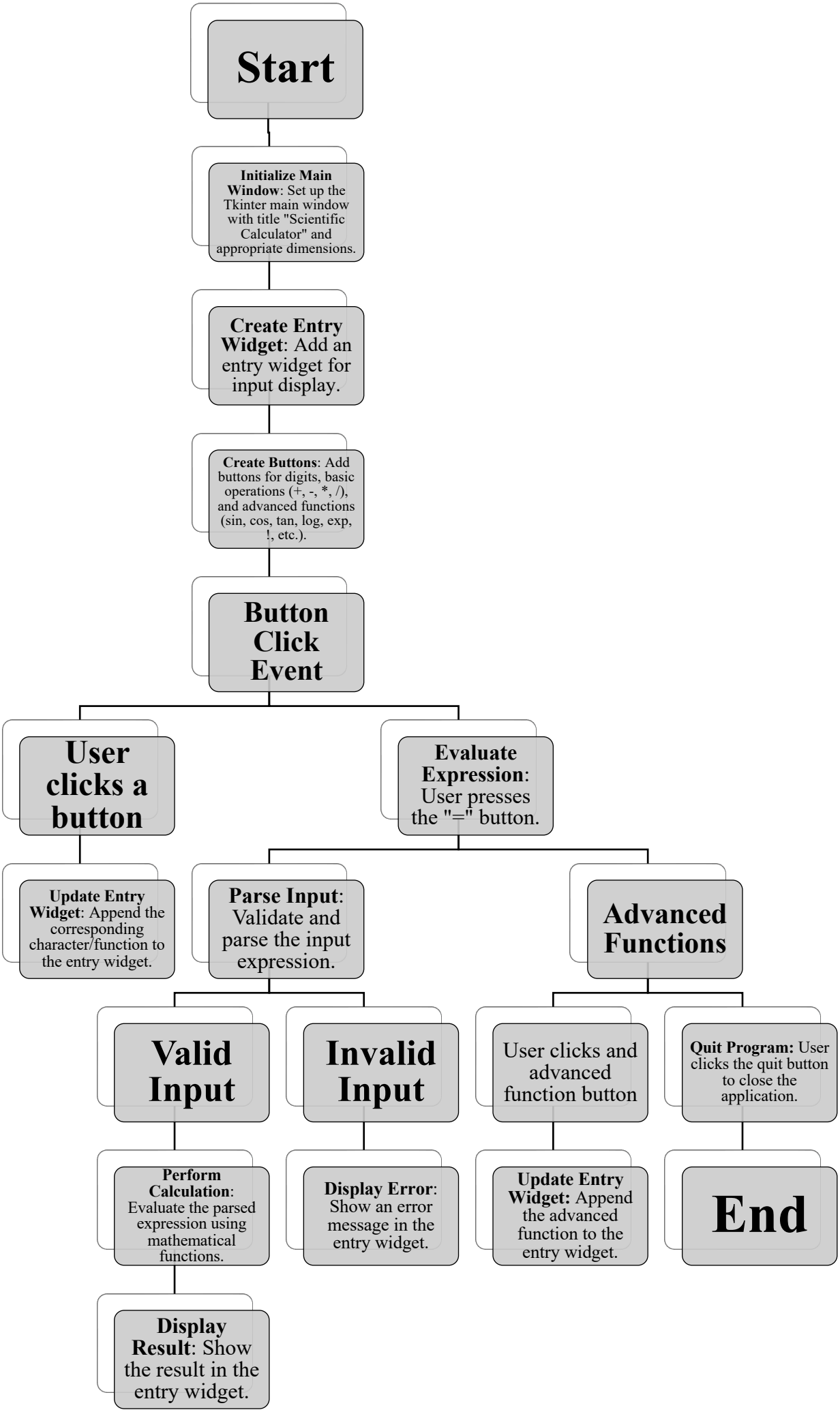
Diagram 1–Calculator Rough Draft Diagram (created by me using Freeform)



Developing a scientific calculator presents several challenges. Firstly, implementing complex mathematical functions accurately and ensuring they perform correctly for a wide range of inputs can be difficult. Error handling is another critical issue; the program must gracefully handle invalid inputs, such as division by zero or malformed expressions, without crashing. Designing an intuitive and visually appealing user interface requires careful layout planning, especially given the need to accommodate numerous buttons and functions. Additionally, the calculator must parse and evaluate mathematical expressions efficiently, which involves handling operator precedence and parentheses correctly. Ensuring the application is responsive and performs calculations quickly is also essential to provide a seamless user experience.

The primary target audience for this scientific calculator includes students and educators who require a reliable tool for mathematical calculations and learning. Professionals in fields such as engineering, physics, and computer science who need a quick and efficient way to perform scientific calculations will also benefit from this tool. Additionally, general users who need a versatile calculator for everyday mathematical tasks are part of the intended audience. The program aims to be user-friendly enough for novices while offering advanced functionalities required by professionals.

Figure 2– Flowchart Diagram of the Program (next page)


Start
Initialize Main Window: Set up the Tkinter main window with title "Scientific Calculator" and appropriate dimensions.
Create Entry Widget: Add an entry widget for input display.
Create Buttons: Add buttons for digits, basic operations (+, -, *, /), and advanced functions (sin, cos, tan, log, exp, !, etc.).
Button Click Event
User clicks a button
Update Entry Widget: Append the corresponding character/function to the entry widget.
Evaluate Expression: User presses the "=" button.
Parse Input: Validate and parse the input expression.
Valid Input
Perform Calculation: Evaluate the parsed expression using mathematical functions.
Display Result: Show the result in the entry widget.
Invalid Input
Display Error: Show an error message in the entry widget.
Advanced Functions
User clicks and advanced function button
Update Entry Widget: Append the advanced function to the entry widget.
Quit Program: User clicks the quit button to close the application.
End