George Maidhof (gpm58) & Parker Schless (pis7) & Edward Liu (ejl248)

CS 3110

16 May 2024

# CS 3110 Final Project

# ArduinBros Calculator Testing Plan

## Testing Approach:

Our testing approach for the calculator was primarily separated into back-end logic and the user interface. Due to the reliance on programmed math functions like in trigonometry, arithmetic, probability, etc., we split each group of math functions into separately unit testable sections. For back-end logic, we created a set of test cases for each function based on basic property testing using a glass box testing approach. Additionally, we could test how the parser was evaluating queries by creating test cases for each implemented math function and checking that the string output matched our expectations. For plotting and the REPL, we primarily relied on manual testing. For functions that produced floating point numbers, we would usually use a function to check that the answer was in a tolerance of 10^-8 of the expected answer.

## OUnit Testing:

The following modules were tested with OUnit:

- Arithmetic
    - Basic arithmetic operations like addition, multiplication, division, subtraction, exponentiation, logarithms, etc. are programmed here. These functions typically show up on the main screen of the TI-84.
    - We utilized glass box testing to ensure various properties of the functions held correctly
- Math
    - These functions show up under the Math tab of the MATH button of the home screen
    - We used glass box testing to ensure each function produced expected output
- Num
    - These functions are under the Num tab of the MATH button of the home screen
    - We used glass box testing to ensure each function produced expected output
- Probability
    - These functions are under the Prb tab of the MATH button of the home screen

    - We used glass box testing to ensure each function produced expected output
- Query
    - We tested invocations of the supported math functions by writing them out as sample queries and asserted the correctness of the result.
    - We used black box testing based on the expected output of the other tested math functions
- Trigonometry
    - These functions show up on the Main screen of the TI-84 but are separated into a separate module
    - We used glass box testing to ensure each function produced expected output.

**Physical:**

- Plot
    - Some testing in our code would be hard to perform in a testbench file, so the approach we went about making sure our code worked for plot was through trial and error and making sure it worked. The first section takes in inputs of strings to label the graph (i.e. plot title, x-axis, and y-axis). The second part of the plot function you input bounds and the function that you want to input as floats, ints and some of the eval types (i.e. sin, cos, tan). If you enter a wrong type for the bounds, you are prompted to re-input an approved value. When inputting an invalid function, you will unfortunately have to redo the plot function. After you finishing these functions, you are then prompted to choose a file output (just type 8 for pdf), and for file name, input your name followed by .pdf (i.e. <name>.pdf) and the file should be output to your home directory of the code with your name and function type.

**Completeness of Testing:**

We believe the testing to be complete as we have individually tested each math function with a variety of edge cases using glass box testing, ensured that the parser properly solved queries with the math functions based on their black box testing output, and visually confirmed that the REPL/plotting/help features produced expected output.