



Arithmetic Library (Non-Coders)

Introductory Document

The objective of this arithmetic library is to provide the user with an easy way of performing the four basic arithmetic operations: addition, subtraction, multiplication, and division.

To demonstrate the adaptability of this library, some possible uses are:

1. Middle school or high school coding class – A teacher can use this library to teach students how to use the library to perform arithmetic operations to teach the concept of reusability or importing libraries to their codes.
2. Data analysis – A data analyst can use this library to perform basic arithmetic operations for large numbers or to clean up their codes and reduce the amount of LOC used.
3. Personal budget – A user can use this library to create their own application for budget planning.

Even though this is intended to be a simple arithmetic library with only four main operations, the users could face some challenges that can influence the appropriate use of the library or the efficiency level to which it is used. One of these challenges might be how to integrate the modules in the library if there are compatibility issues. Another challenge could be the lack of examples on how to use or implement the library's modules. However, the benefits are bigger than the challenges as the library can help clean the codes and minimize the LOC amount. This leads to coding efficiency as well as readability. This library can also serve as a basis for future educational values as it can be reused and extended to meet any user needs due to the modularity of the library. These extensibilities could be using the outputs to calculate standard deviations, value comparisons, or any other mathematical needs.

This is how the library works:

After a user had identified their requirements, they can look for the arithmetic operation they need to perform. The library is split in 5 modules: Addition, Subtraction, Multiplication, Division, and Input Definition. All 4 core modules use the module of input and output definition which validates the user input by assuring that all inputs are of the same data type. If the inputs are not of the same data type, then it will throw an error handling message. After this step has been done, the modules then proceed to perform the assigned arithmetic operation. The outputs are given in the same data type of the inputs. Each core module is created to that it can be used individually, which creates an opportunity for more readiness and better usability.

