# Design Document

## Algorithms

### Add

Describe the algorithm used.

Prior adding the numbers, it will add zero at the shortest number so they will be in length equal for easy addition. When start adding the numbers the method start addition from the right most number and continue until it finishes. While adding both numbers, and the result is greater than 10 then it will have a carry value that will be added to the next number. Once it finish adding them the result will be return to the called.

### Subtract

Describe the algorithm used.

The program follows regular subtraction that any human used to do in elementary school which is to subtract numbers from the right most until all number are completed.

While doing the subtraction from the right most, if there is any number that is smaller number that the one subtracted, then it start borrowing number from the next number from the left side. This is the way human do substation and try to emulate this way of doing so we can do it even for larger numbers beyond the capacity of the memory available in the older machine.

### Multiply

Describe the algorithm used. This method also follows how human do multiplication by starting from last digit of second number and multiply it with first number. Then it multiply next digit of second number with first number, and so on. At the end , we will be adding all the number . When we add we shit the numbers so the addition become correct as human do the addition.

### Divide

Describe the algorithm used.

The division does the opposite of what I have explained above in multiplication. It follows the same procedure as how humans do for a division. As the dividend and the result that we get can be very large we store them in a temporary String value. The first thing we do is to take numbers that are divisible by devider in our case “b” that we will be passing as a parameter . Once we complete doing so we save it in result String so we are able to return the value to the caller.

## Scalability

Describe with words and diagrams how you would scale the Calculator service component when deployed to public cloud provider like Amazon’s AWS or Microsoft’s Azure.

In case this application will be used extensively, it will be possible to have multiple instances of it with load-balancer in front of the application so it will accommodate the request from multiple clients. I do believe this will be the best approach in order to avoid latency as well as crashing of the system time to time.