Name: Ali Albayrak

Student ID: P304320

Date: 10/11/17

Portfolio Activity 1.7: Documentation

Lecturer: Stewart Godwin

GitHub URL: https://github.com/Hodkan/AT1.7-Calculator.git

Contents

[Data Structures 3](#_Toc498674359)

[Algorithms 4](#_Toc498674360)

[Flow Charts: 4](#_Toc498674361)

[Arithmetic Functions: 4](#_Toc498674362)

[Trigonometric Functions: 7](#_Toc498674363)

[Algebraic Functions: 8](#_Toc498674364)

[Error Handling Techniques: 10](#_Toc498674365)

[Recommended testing procedure: 10](#_Toc498674366)

[Arithmetic 10](#_Toc498674367)

[Trigonometric 11](#_Toc498674368)

[Algebraic 12](#_Toc498674369)

[Evidence of Testing/Results: 13](#_Toc498674370)

[Arithmetic: 13](#_Toc498674371)

[Trigonometric: 14](#_Toc498674372)

[Algebraic: 22](#_Toc498674373)

[Recommendations on upgrades and future enhancements 27](#_Toc498674374)

# Data Structures

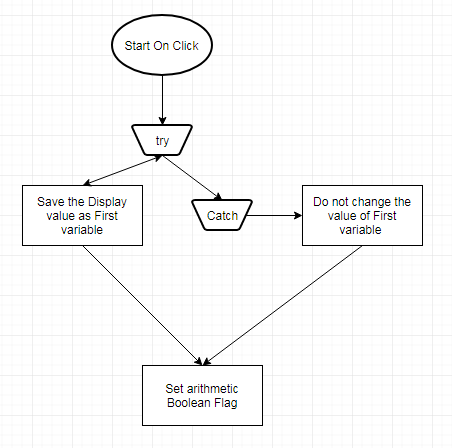
|  |  |  |
| --- | --- | --- |
| Name | Type | Purpose |
| Isplus | Boolean | Boolean flag for adding operation. |
| Issub | Boolean | Boolean flag for subtraction operation. |
| Isdiv | Boolean | Boolean flag for dividing operation. |
| Ismult | Boolean | Boolean flag for multiplication operation. |
| First | Double | To save the first input number. |
| Result | Double | To save and display the result of operations. |
| input | Double | To save the value for trigonometric operations. |
| remainder | Double | To check if input is multiple of 90 degrees |
| multiple | Double | To save the how many times input value multiples 90 |
| even | Double | To save if multiple value is even or odd. |

# Algorithms

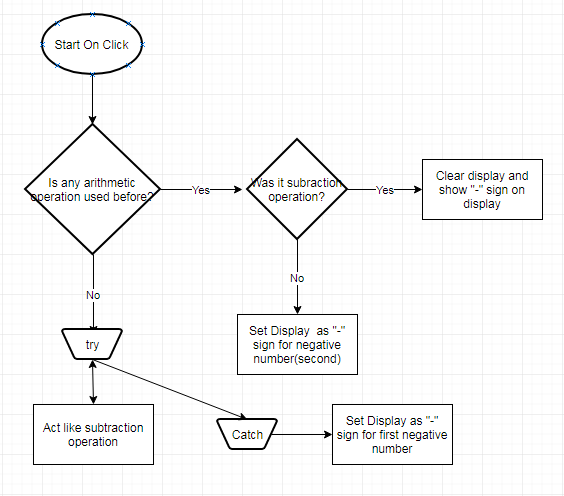
## Flow Charts:

### Arithmetic Functions:

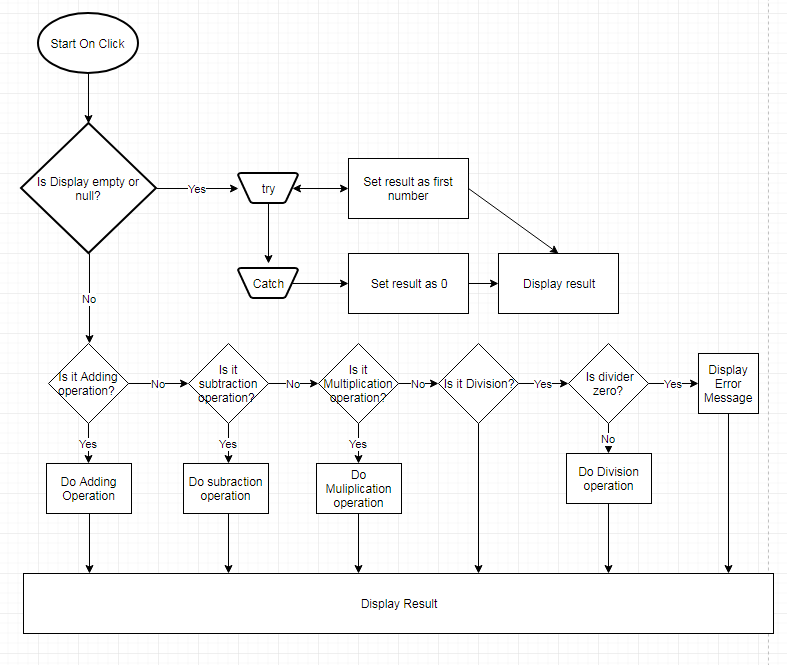
Addition/Multiplication/Division:



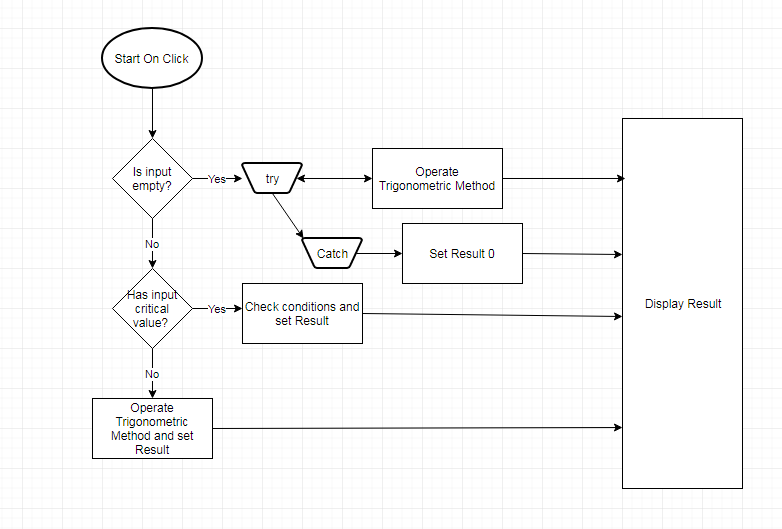
Subtraction:



Equals:

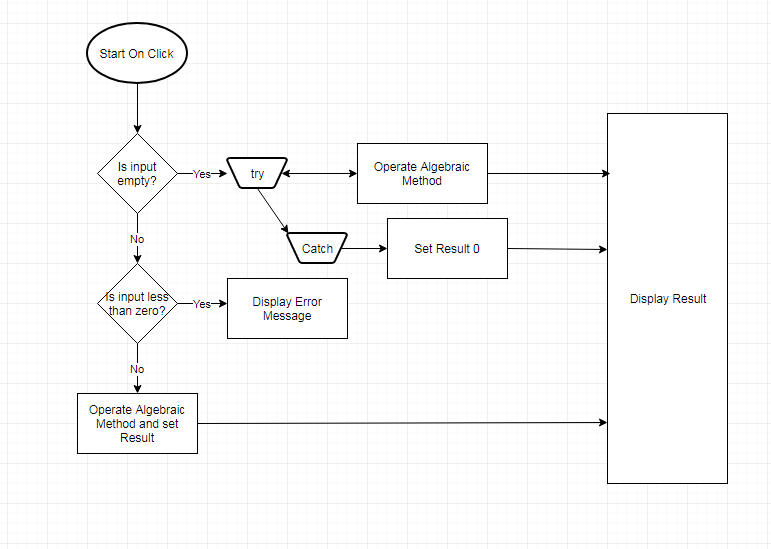


### Trigonometric Functions:

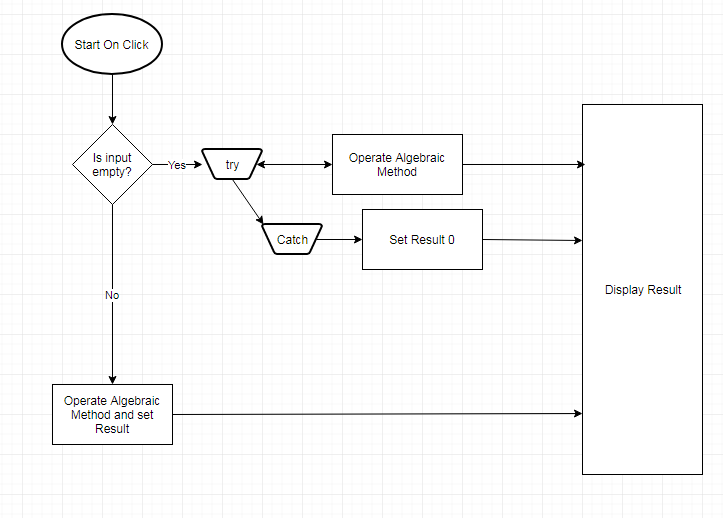


### Algebraic Functions:

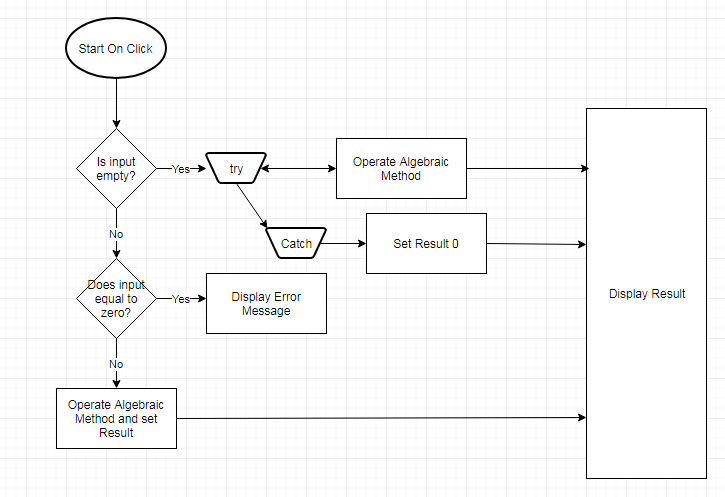
Square Root:



Cube Root:



Inverse:



## Error Handling Techniques:

Try and catch methods with error messages are used in code in purpose of error handling techniques. Flow chart of each sections contains the try and catch methods to display error handling techniques that used in code.

# Recommended testing procedure:

This program should be tested by using test table. Also, calculator should be tested by entering inappropriate and unexpected inputs to see error handling techniques.

This example of test table could be used for each section it has:

## Arithmetic

|  |  |  |
| --- | --- | --- |
| **Test** | **Expected** | **Actual / Comment** |
| **Addition** | | |
| -5 + -5 | -10 | -10 / Correct |
| -5 + 0 | -5 | -5 / Correct |
| 0 + 5 | 5 | 5 / Correct |
| 5 + 5 | 10 | 10 / Correct |
| **Subtraction** | | |
| -5 - -5 | 0 | 0 / Correct |
| 5 - 0 | 5 | 5 / Correct |
| 0 - 5 | -5 | -5 / Correct |
| 10 - 15 | -5 | -5 / Correct |
| **Multiplication** | | |
| 5 \* 5 | 25 | 25 / Correct |
| 5 \* 0 | 0 | 0 / Correct |
| 0 \* 5 | 0 | 0 / Correct |
| 5 \* -5 | -25 | -25 / Correct |
| **Division** | | |
| 10 / 5 | 2 | 2 / Correct |
| 10 / 3 | 3,33333 | 3,33333 / Correct |
| 10 / 0 | Error | Error / Correct (Check Screenshot) |
| 0 / 10 | 0 | 0 / Correct |
| 10 / -2 | -5 | -5 / Correct |

## Trigonometric

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test** | **Expected** | | | **Actual / Comment** |
| **Tan** | | | | |
| 0o | | 0 | 0 / Correct | |
| 30o | | 0.577350269189626 | 0.577350269189626 / Correct | |
| 45o | | 1 | 1 / Correct | |
| 60o | | 1,73205080756888 | 1,73205080756888 / Correct | |
| 90o | | Invalid / Error | Invalid / Error / Correct | |
| **Sin** | | | | |
| 0o | | 0 | 0 / Correct | |
| 30o | | 0.5 | 0.5 / Correct | |
| 45o | | 0.70710678118654 | 0.70710678118654 / Correct | |
| 60o | | 0,866025403784439 | 0,866025403784439 / Correct | |
| 90o | | 1 | 1 / Correct | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Cos** | | | | |
| 0o | | 1 | 1 / Correct | |
| 30o | | 0,866025403784439 | 0,866025403784439 / Correct | |
| 45o | | 0,70710678118654 | 0,70710678118654 / Correct | |
| 60o | | 0.5 | 0.5 / Correct | |
| 90o | | 0 | 0 / Correct | |
|  |  |  |  |  |

## 

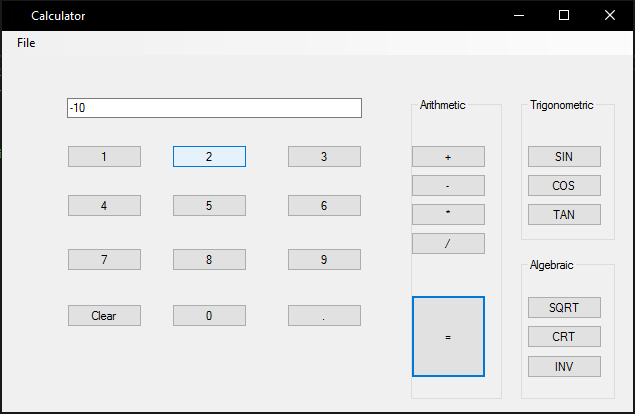
## Algebraic

|  |  |  |
| --- | --- | --- |
| **Test** | **Expected** | **Actual / Comment** |
| **Square Root** | | |
| 0 | 0 | 0 / Correct |
| 1 | 1 | 1 / Correct |
| 2 | 1,4142135623731 | 1,4142135623731 / Correct |
| -1 | Error | Error / Correct |
| **Cube Root** | | |
| 0 | 0 | 0 / Correct |
| 1 | 1 | 1 / Correct |
| 2 | 1,25992104989487 | 1,25992104989487 / Correct |
| -1 | -1 | -1 / Correct |
| **Inverse** | | |
| 0 | Error | Error / Correct |
| 1 | 1 | 1 / Correct |
| 2 | 0.5 | 0.5 / Correct |
| -1 | -1 | -1 / Correct |

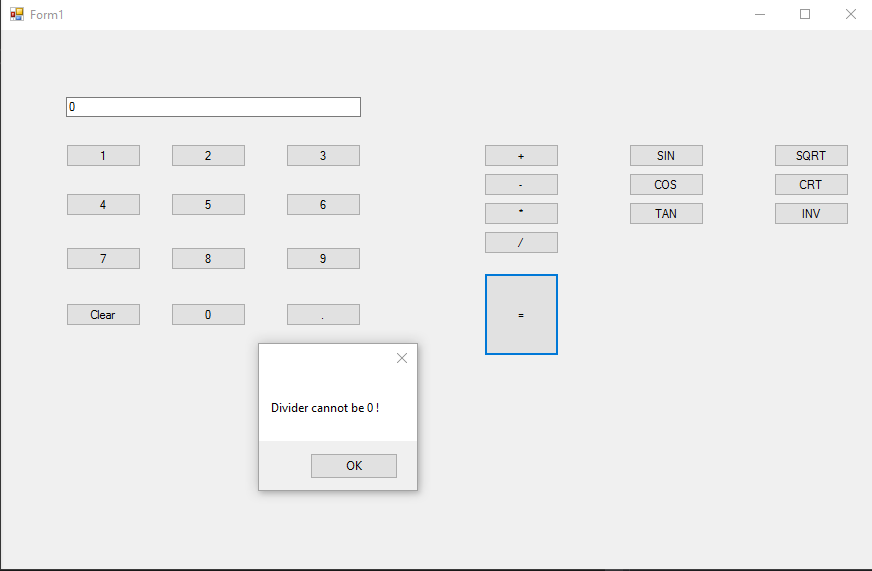
# Evidence of Testing/Results:

## Arithmetic:

-5 + -5:



Division Error (Divided by 0):



## 

## Trigonometric:

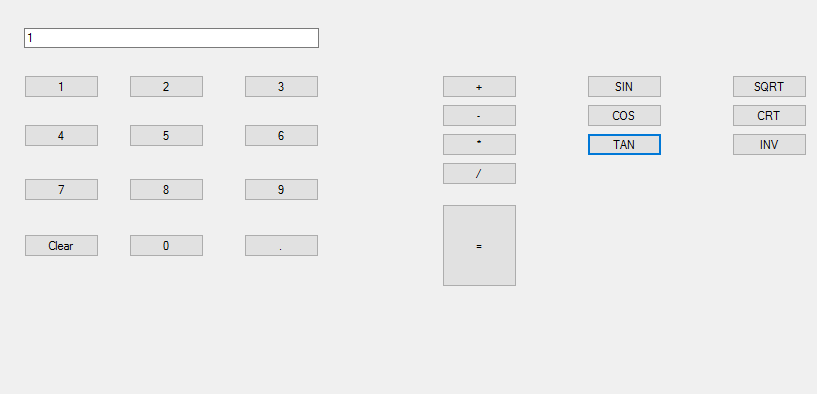
TAN 0:



TAN 30:



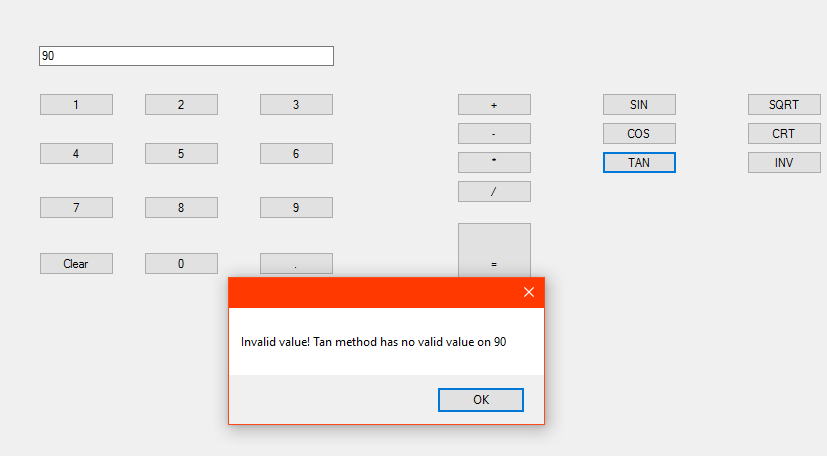
TAN 45:



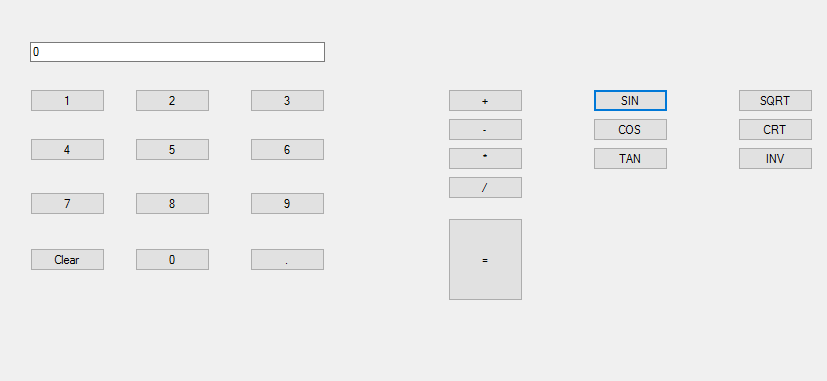
TAN 60:



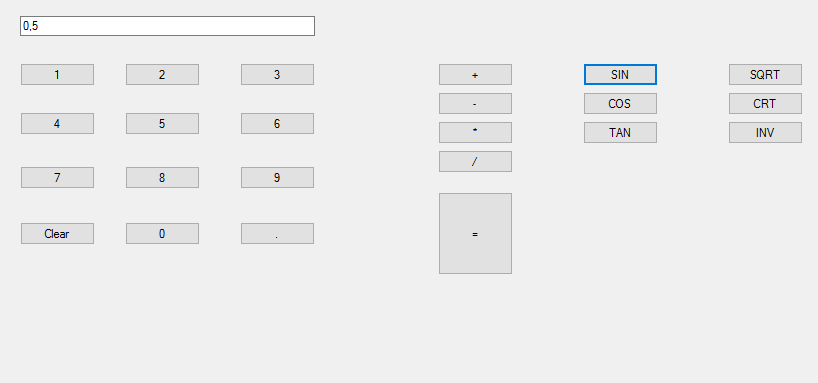
TAN 90:



Sin 0:



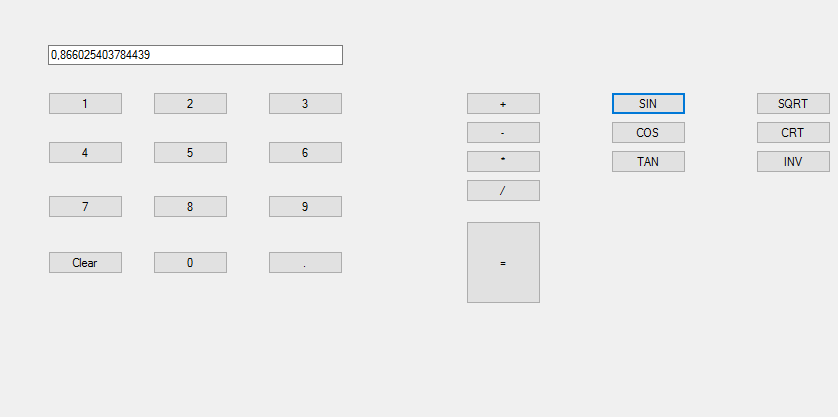
Sin 30:



Sin 45:



Sin 60:



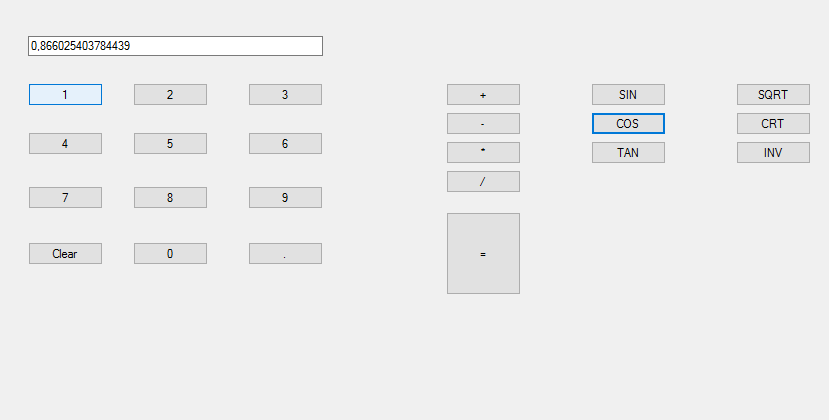
Sin 90:



Cos 0:



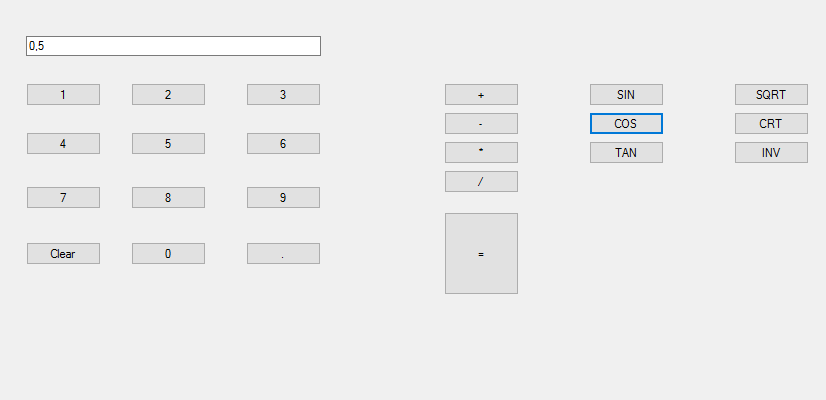
Cos 30:



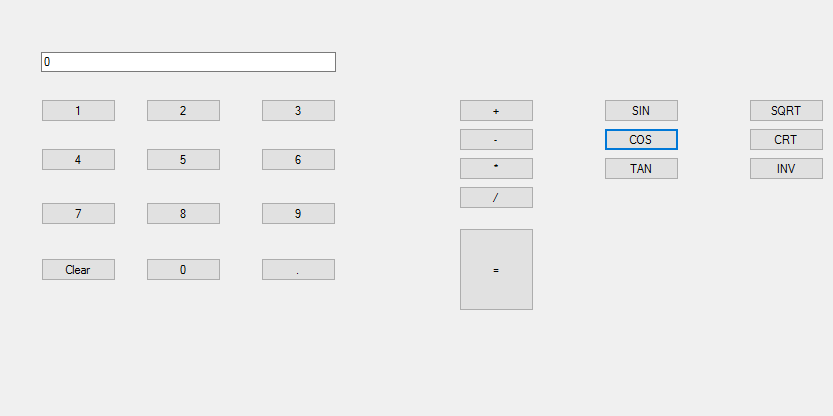
Cos 45:



Cos 60:



Cos 90;



## 

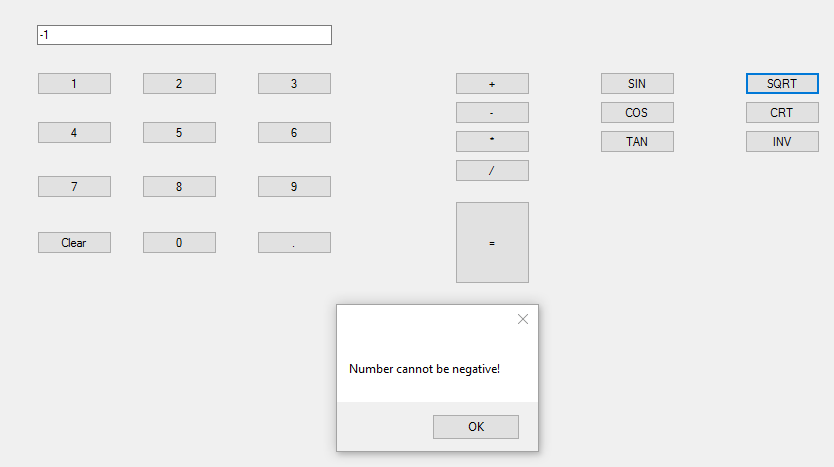
## Algebraic:

Square Root:

2:

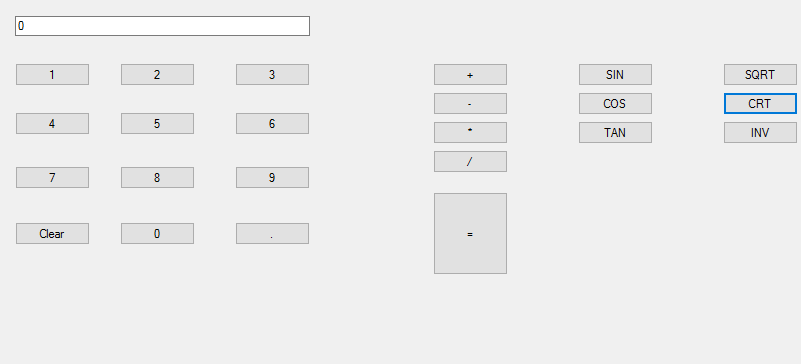


-1:



Cube Root:

0:



1:



2:

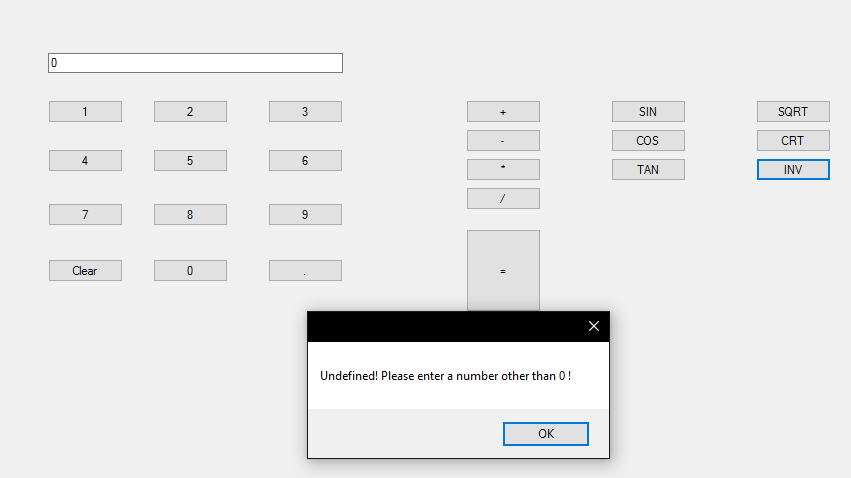


-1:

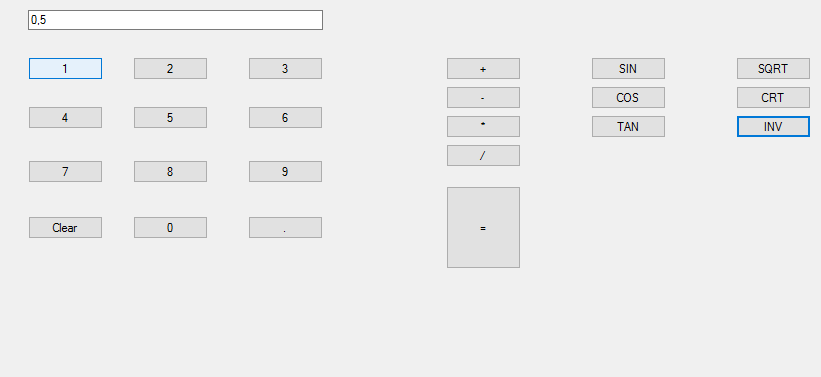


Inverse:

0:



2:



-1:



# Recommendations on upgrades and future enhancements

I would recommend to add more function on calculator. There can be more trigonometric and algebraic methods and functions such as cot, sec, cosec. Also, I would recommend to add an ability that performs more than one operation in a single time. Additionally, history of operations could be displayed. Moreover, visual and design of the calculator can be improved.