**https://github.com/TArnoldTafe/Portfolio1.7Calculator**

Tobey Arnold

Tafe 2017

Portfolio 1.7 Technical Document

Contents

[Data Structures 2](#_Toc481548601)

[Explanation 2](#_Toc481548602)

[Variables Table 2](#_Toc481548603)

[Algorithms 6](#_Toc481548604)

[BasicMath 6](#_Toc481548605)

[AdvArithmetic 6](#_Toc481548606)

[TrigArithmetic 6](#_Toc481548607)

[Error Handling 7](#_Toc481548608)

[Test Procedure 8](#_Toc481548609)

[Upgrades 8](#_Toc481548610)

## Data Structures

### Explanation

This calculator program primarily uses doubles to store the numerical data, which is then converted to a string for purposes of displaying in the textbox. The numerical buttons 0-9 as well as the decimal button all send the text inside the button to the display, which is treated as a double. The basic operators have a Boolean attached that stores which operation button was clicked. This is useful for the Inverse function, which needs to know which button was clicked so it can do the opposite. The more advanced operators all make use of the external libraries imported in, which in turn make use of the Math library.

The external libraries get the value from the textbox, and calculate and return the result of the calculations as a double, which can then be displayed on the calculator as a string.

### Variables Table

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Variable Name** | **Type** | **Purpose** |
| **TobeyArnold\_Portfolio1.6** |  |  |  |
| Global Variables | leftNumber | Double | Stores the text from the display for the first half of the sum |
|  | Result | Double | The calculated number |
|  | plusButtonClicked | Bool | Stores if the Plus button was clicked |
|  | minusButtonClicked | Bool | Stores if the Minus button was clicked |
|  | divideButtonClicked | Bool | Stores if the Divide button was clicked |
|  | multiplyButtonClicked | Bool | Stores if the Multiply button was clicked |
| Btn0Click | txtResult.Text | String | The text in the display |
|  | btn0.Text | String | The text found in the button |
| Btn1Click | txtResult.Text | String | The text in the display |
|  | btn0.Text | String | The text found in the button |
| Btn2Click | txtResult.Text | String | The text in the display |
|  | btn0.Text | String | The text found in the button |
| Btn3Click | txtResult.Text | String | The text in the display |
|  | btn0.Text | String | The text found in the button |
| Btn4Click | txtResult.Text | String | The text in the display |
|  | btn0.Text | String | The text found in the button |
| Btn5Click | txtResult.Text | String | The text in the display |
|  | btn0.Text | String | The text found in the button |
| Btn6Click | txtResult.Text | String | The text in the display |
|  | btn0.Text | String | The text found in the button |
| Btn7Click | txtResult.Text | String | The text in the display |
|  | btn0.Text | String | The text found in the button |
| Btn8Click | txtResult.Text | String | The text in the display |
|  | btn0.Text | String | The text found in the button |
| Btn9Click | txtResult.Text | String | The text in the display |
|  | btn0.Text | String | The text found in the button |
| BtnDecimalClick | txtResult.Text | String | The text in the display |
|  | btn0.Text | String | The text found in the button |
| BtnAddClick | leftNumber | Double | Stores the text from the display for the first half of the sum |
|  | plusButtonClicked | Bool | Stores that the Plus button was clicked |
|  | minusButtonClicked | Bool | Stores that the Minus button was not clicked |
|  | divideButtonClicked | Bool | Stores that the Divide button was not clicked |
|  | multiplyButtonClicked | Bool | Stores that the Multiply button was not clicked |
| BtnSubtractClick | leftNumber | Double | Stores the text from the display for the first half of the sum |
|  | plusButtonClicked | Bool | Stores that the Plus button was not clicked |
|  | minusButtonClicked | Bool | Stores that the Minus button was clicked |
|  | divideButtonClicked | Bool | Stores that the Divide button was not clicked |
|  | multiplyButtonClicked | Bool | Stores that the Multiply button was not clicked |
| BtnDivideClick | leftNumber | Double | Stores the text from the display for the first half of the sum |
|  | plusButtonClicked | Bool | Stores that the Plus button was not clicked |
|  | minusButtonClicked | Bool | Stores that the Minus button was not clicked |
|  | divideButtonClicked | Bool | Stores that the Divide button was clicked |
|  | multiplyButtonClicked | Bool | Stores that the Multiply button was not clicked |
| BtnMultiplyClick | leftNumber | Double | Stores the text from the display for the first half of the sum |
|  | plusButtonClicked | Bool | Stores that the Plus button was not clicked |
|  | minusButtonClicked | Bool | Stores that the Minus button was not clicked |
|  | divideButtonClicked | Bool | Stores that the Divide button was clicked |
|  | multiplyButtonClicked | Bool | Stores that the Multiply button was not clicked |
| BtnEqualsClick | result | Double | The calculated result |
|  | txtResult.Text | String | The calculated result displayed on the display |
|  | leftNumber | Double | Clears the sum |
| BtnSquareRootClick | num | Double | The number from the txtResult textbox |
|  | txtResult.Text | String | The calculated result displayed on the display |
| BtnCubeRootClick | num | Double | The number from the txtResult textbox |
|  | txtResult.Text | String | The calculated result displayed on the display |
| BtnInverseClick | result | Double | The calculated Result |
|  | leftNumber | Double | The number from the txtResult textbox |
|  | plusButtonClicked | Bool | Stores if the Plus button was clicked |
|  | minusButtonClicked | Bool | Stores if the Minus button was clicked |
|  | divideButtonClicked | Bool | Stores if the Divide button was clicked |
|  | multiplyButtonClicked | Bool | Stores if the Multiply button was clicked |
| BtnTanClick | num | Double | The number from the txtResult textbox |
|  | txtResult.Text | String | The calculated result displayed on the display |
| BtnSinClick | num | Double | The number from the txtResult textbox |
|  | txtResult.Text | String | The calculated result displayed on the display |
| BtnCosClick | num | Double | The number from the txtResult textbox |
|  | txtResult.Text | String | The calculated result displayed on the display |
|  |  |  |  |
| **BasicMath** |  |  |  |
| Add | a | Double | Left number of the sum |
|  | b | Double | Right number of the sum |
| Subtract | a | Double | Left number of the sum |
|  | b | Double | Right number of the sum |
| Multiply | a | Double | Left number of the sum |
|  | b | Double | Right number of the sum |
| Divide | a | Double | Left number of the sum |
|  | b | Double | Right number of the sum |
|  |  |  |  |
| **AdvArithmetic** |  |  |  |
| SquareRoot | a | Double | Number to get the square root of |
| CubeRoot | a | Double | Number to get the cube root of |
|  |  |  |  |
| **TrigArithmetic** |  |  |  |
| Tan | a | Double | Number to be calculated |
|  | result | Double | The calculated result |
| Sin | a | Double | Number to be calculated |
|  | result | Double | The calculated result |
| Cos | a | Double | Number to be calculated |
|  | result | Double | The calculated result |

## Algorithms

### BasicMath

#### Add

double Add (double a, double b){

return a + b

}

#### Subtract

double Subtract (double a, double b){

return a - b

}

#### Divide

double Divide (double a, double b){

return a / b

}

#### Multiply

double Multiply (double a, double b){

return \* b

}

### AdvArithmetic

#### SquareRoot

double SquareRoot (double a){

get squareroot of a

return a

}

#### CubeRoot

double CubeRoot (double a){

get a to the power of (1 / 3) //C# doesn’t have a built in cube root function

return a

}

### TrigArithmetic

#### Tan

Double Tan (double a){

convert a to radians

calculate tan based off of radians value

return calculated tan

}

#### Sin

Double Tan (double a){

convert a to radians

calculate sin based off of radians value

return calculated sin

}

#### Cos

Double Tan (double a){

convert a to radians

calculate cos based off of radians value

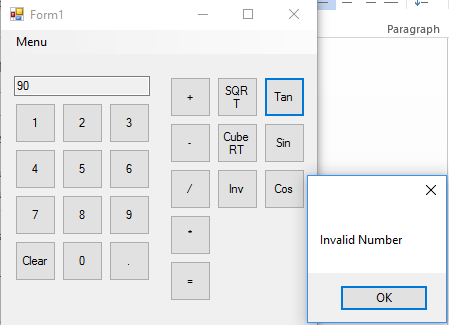
return calculated cos

}

### Error Handling

The main calculator program does some very basic error handling by making sure that the double pulled from the calculator is below 90, this ensures that no negative values (and therefore invalid) numbers are calculated. If the number is 90 or above or is below 0 then the calculator outputs an error message.

#### Example



## Test Procedure

To test this program before release every button must be tested with both valid and invalid entries to ensure bug free use. If possible, all button combinations should be tested, as well as many different operating systems and computers as possible. This will ensure consistent performance with many computers.

## Upgrades

The calculator program as it is now is fairly basic and could do with many upgrades and improvements for a better product. Some possible upgrades include:

* Keyboard input as well as button input
* History of previous calculations and the results
* Ability to store numbers in algebraic symbols
* Ability to display percentages and fractions, not just decimals
* Button to clear most recent entry as well as clear all

Many of these upgrades would be fairly easy to add on to the calculator in the future if needed.