

Calculator Technical Report

Kaelan Oxby



June 15, 2019

SOUTH Metro TAFE

<https://github.com/KaelanOxby/Calculator>

Contents

[Data Structures 2](#_Toc11508605)

[Algorithms 2](#_Toc11508606)

[Number Buttons: 2](#_Toc11508607)

[Stop Button: 3](#_Toc11508608)

[Clear Button: 3](#_Toc11508609)

[Equals Button: 4](#_Toc11508610)

[Plus Button, Minus Button, Multiply Button: 5](#_Toc11508611)

[Divide Button: 6](#_Toc11508612)

[Trigonometric Buttons: 6](#_Toc11508613)

[Algebraic: 7](#_Toc11508614)

[Square Root Button: 7](#_Toc11508615)

[Invert Button: 8](#_Toc11508616)

[Cube Root Button: 8](#_Toc11508617)

[Switch Button: 9](#_Toc11508618)

[Backspace Button: 9](#_Toc11508619)

[Recommended Testing Procedure 10](#_Toc11508620)

[Future Upgrades 10](#_Toc11508621)

# Data Structures

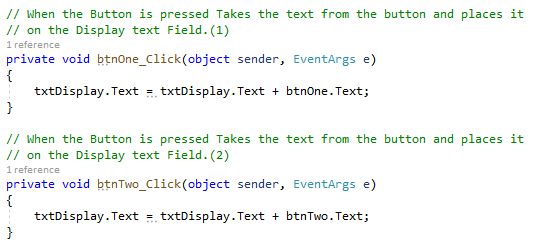
These structures are used to store variables and assign Boolean values for the equals method to use to define what it is calculating.

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Purpose** |
| total1 | double | Takes in the first value entered by the user to use for calculations |
| total2 | double | Takes the Second value entered and is used by calculations |
| result | double | It is used to assign the final calculation from total 1 and total 2 and give a result. |
| plusButtonClicked | bool | Holds the true/false values for the plus button. The equals method uses this to calculate the result |
| minusButtonClicked | bool | Holds the true/false values for the minus button. The equals method uses this to calculate the result |
| divideButtonClicked | bool | Holds the true/false values for the divide button. The equals method uses this to calculate the result |
| mutiplyButtonClicked | bool | Holds the true/false values for the divide button. The equals method uses this to calculate the result |
| num | double | This is a variable created in a method to take the text parsed from the txtDisplay it works the same as total1. |
| a | double | This is a variable created in a method to take the text parsed from the txtDisplay it works the same as total1. |

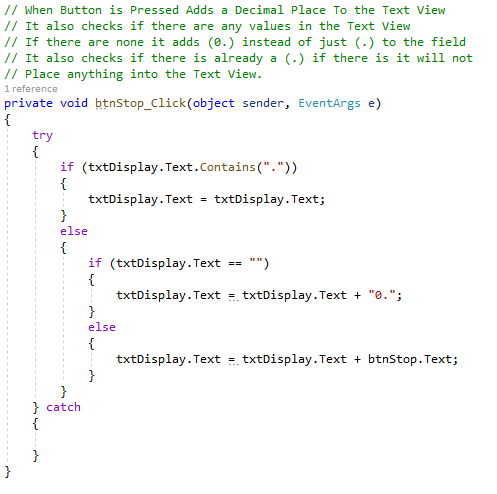
# Algorithms

## Number Buttons:

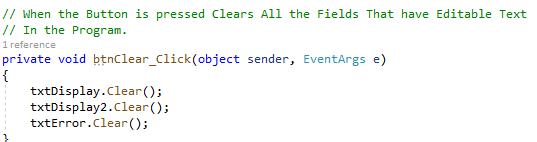
All the number buttons act the same they take the text from the button and adds it to the txtDisplay.



## Stop Button:



## Clear Button:



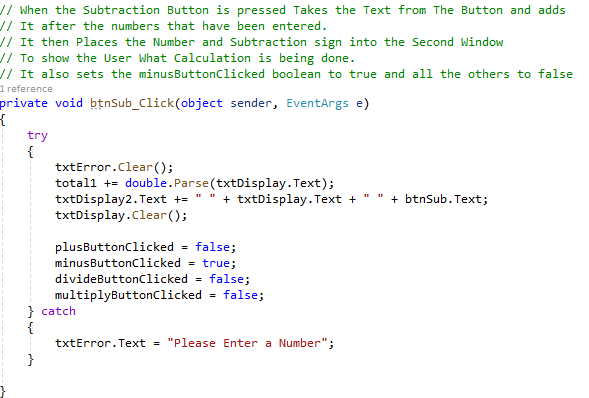
## Equals Button:

The equals button uses the Boolean arithmetic variables which will have 3 false and 1 true and uses that true one to do the calculations. It uses if else statements to state which one is true and do a calculation accordingly. It then resets total 1 at the end of the method. All methods are help in try catches to avoid any crashes.



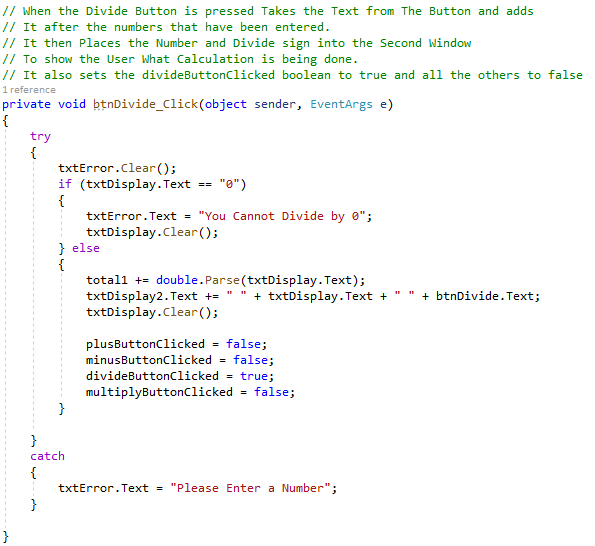
## Plus Button, Minus Button, Multiply Button:

These Three buttons do the exact same thing but just change the Boolean value of its individual type. If it is plus it makes the plusButtonPressed true and all the others false. Etc. Try Catches are used so that the application doesn’t crash.



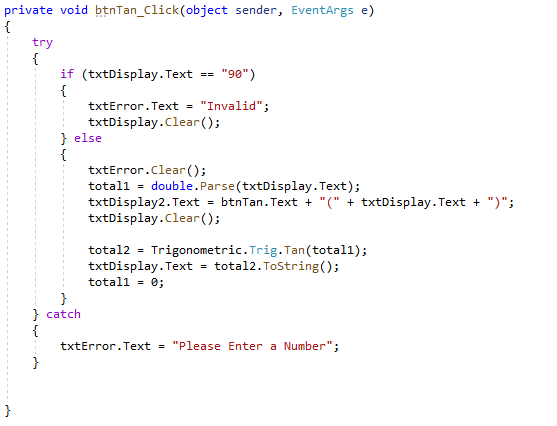
## Divide Button:

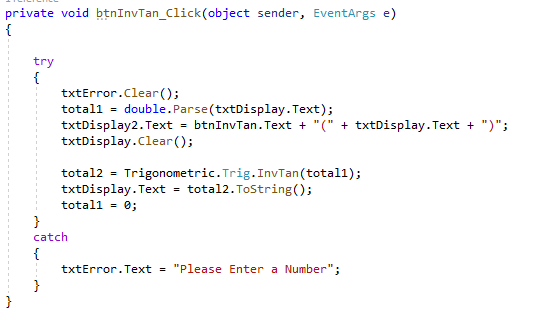
This is basically the same as the other three arithmetic buttons the only difference is that it checks if the value is 0, because you cannot divide by zero. The Button uses a try catch to ensure no crashes occur with incorrect values.



## Trigonometric Buttons:

All 6 of my Trigonometric Buttons use basically the same method just changing the actual process being used from sin to tan etc. The Method parses the value from the txtDisplay and feeds it into the calculation done in the method. It then places that text into txtDisplay2 and displays the answer in txtDisplay1. They are help in try catches so no crashing occurs on incorrect data entered.

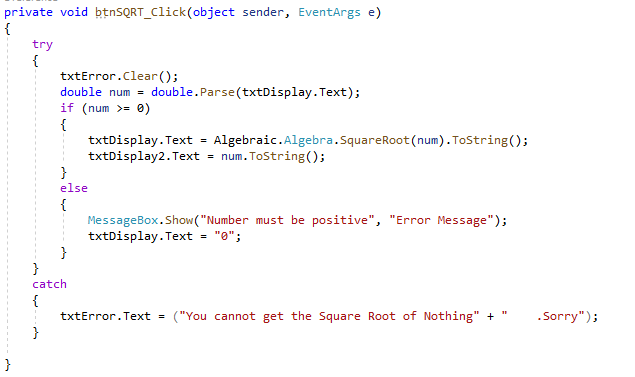




## Algebraic:

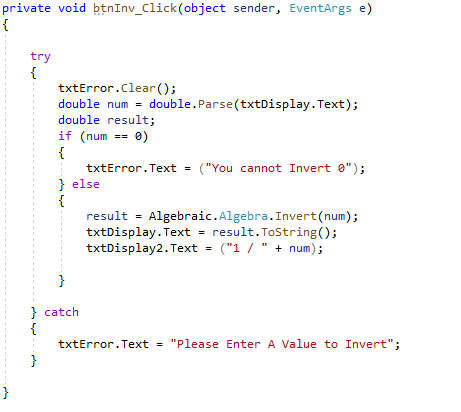
### Square Root Button:

When the button is pressed it parses the text from txtDisplay and assigns it to num variable. It then uses the SQRT Algebraic Function from the Library and calculates the result. It also assigns the value entered and the calculation algorithm to txtDisplay2.



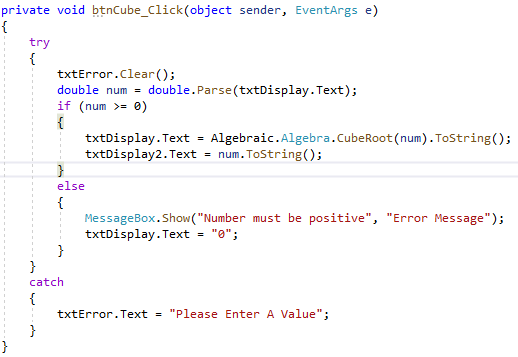
### Invert Button:

Parses the number from the display and places it in the num variable. It then uses the Inv function from the Algebraic library to do the Inv Calculation. It is in a try catch to avoid crashes. It also assigns the value entered and the calculation algorithm to txtDisplay2.



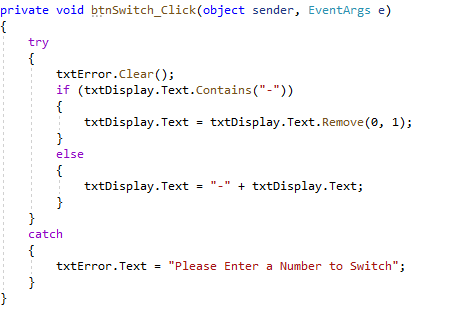
### Cube Root Button:

Parses the number from the display and places it in the num variable. It then uses the Cube Root function from the Algebraic library to do the Cube Root Calculation. It is in a try catch to avoid crashes. It also assigns the value entered and the calculation algorithm to txtDisplay2.



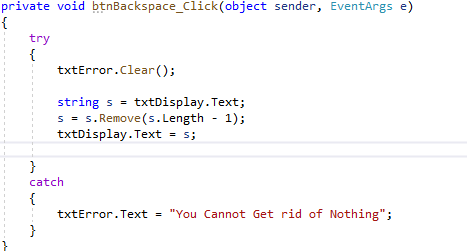
## Switch Button:

Converts the negative values to positive values and positive values to negative. It does this by checking if the txtDisplay contains a (-) it uses the Text.Remove method to remove the first value taking away the -. Else it adds a – to the front. It is all held in a try catch to ensure the program does not crash.



## Backspace Button:

When the button is pressed it uses Text.Remove to remove 1 value from the string. The string is taken from txtDisplay.Text. It is in a try catch to make sure the program does not crash.



# Recommended Testing Procedure

For this application I recommend focusing on calculation accuracy. Using singular calculations as this is all this program can handle. Using decimals, random numbers and zeros and making sure that calculations are done correctly. The same goes for Trigonometry functions, test against things that do not work and use random numbers including negatives to make sure that it all works correctly and that the calculations are accurate. Testing against an already created calculator would be a good way to test this program for accuracy. Another form of testing that could be done just randomly pressing buttons to make sure that the application displays error messages and does not crash.

# Future Upgrades

The main future upgrade that I would implement would be to allow multiple calculations to be done at once. I would also want to introduce BIMDAS, as even introducing multiple calculations the calculation would still be incorrect as it would not follow BIMDAS and follow the input order when calculating. I would also implement on key listeners so that the user can use the number pad and number keys on their keyboard to input numbers as well as symbols like (-, +, +, /).