Portfolio Activity 1.7

**Technical Document**

Challenger Institute of Technology

Mathew Lowson

Table of Contents

[Data Structures 2](#_Toc478565638)

[Name, type and purpose of each variable (in a table format) 2](#_Toc478565639)

[Algorithms 2](#_Toc478565640)

[Pseudo code for each method 2](#_Toc478565641)

[btnNumbers\_Click 2](#_Toc478565642)

[SignClicked 2](#_Toc478565643)

[btnSQRT\_Click 2](#_Toc478565644)

[btnInvert\_Click 3](#_Toc478565645)

[btnSine\_Click 3](#_Toc478565646)

[btnTan\_Click 3](#_Toc478565647)

[btnEquals\_Click 3](#_Toc478565648)

[btnCubeRT\_Click 4](#_Toc478565649)

[btnDot\_Click 4](#_Toc478565650)

[Error handing techniques 5](#_Toc478565651)

[Recommended testing procedure (how should this software be tested before commercial release) 5](#_Toc478565652)

[Recommendations on upgrades and future enhancements 5](#_Toc478565653)

# Data Structures

## Name, type and purpose of each variable (in a table format)

|  |  |  |
| --- | --- | --- |
| Name | type | purpose of each variable |
| a | string | Keeps the value user enter for “a” |
| b | string | Keeps the value user enter for “b” |
| da | double | Converts the value of “a” to a double |
| db | double | Converts the value of “b” to a double |

# Algorithms

## Pseudo code for each method

### btnNumbers\_Click

* make button named “but” equals (button)sender
* does sign label equals “Sign” or “±” or ”√” or “∛ “
  + if so then make “a” equal the number on the button
  + does label “A” contain “.0”
    - if so make label “A” equal itself minus the trailing “0”
  + does label “A” equals “A”
    - if so make label “A” equal “a”;
  + else make label “A” equal itself plus “a”;
* else make “B” equal the number on the button
* does label “B” contain “.0”
  + if so make label “B” equal itself minus the trailing “0”
* does label “B” equals “B”
  + if so make label “B” equal “b”;
* else make label “B” equal itself plus “b”;

### SignClicked

* make button named “b” equals (button)sender
* make sign label the text of b

### btnSQRT\_Click

* does label “A” equal a number
  + if not then return
* else make “da” equal label “A”
* make sign label equal “√”
* make textbox “result” equal Maths.Algebraic.SqrRt(da) then convert to string

### btnInvert\_Click

* does label “A” equal a number
  + if not then return
* else make “da” equal label “A”
* make sign label equal ±
* dose textbox “result” equal “Result”
  + if so do nothing
* else dose textbox “result” not equal “a”
  + make “a” equals textbox “result”
* make “a” equals Maths.Algebraic.Inverse(da) then convert it to string

• Make textbox “result” equal “a”

* Make label “A” equal “a”

### btnSine\_Click

* does label “A” equal a number
  + if not then return
* else make “da” equal label “A”
* make textbox “result” equals Maths.Trigonometric.Sine(da)then convert to string

BntCosine\_Click

* does label “A” equal a number
  + if not then return
* else make “da” equal label “A”
* make textbox “result” equals Maths.Trigonometric.Cosine(da)then convert to string

### btnTan\_Click

* does label “A” equal a number
  + if not then return
* else make “da” equal label “A”
* does “da” equal “90”
  + if so make textbox “result” equals “Invalid”
* Else make textbox “result” equals Maths.Trigonometric.Tan(da)then convert to string

### btnEquals\_Click

* While sign Label is equal to “+”, ”/”, “\*”, “-”
* Make “da” then sign label then “db”

### btnCubeRT\_Click

* does label “A” equal a number
  + if not then return
* else make “da” equal label “A”
* make sign label equal ∛
* make textbox “result” equals Maths.Algebraic.CubRt(da) then convert to string

### btnDot\_Click

* does sign label equals “Sign”
  + Dose “a” not contain “.”
    - Then make “a” equals “.0”
    - Make label “A” equal itself plus “a”
* Else Dose “a” not contain “.”
  + Then make “a” equals “.0”
  + Make label “A” equal itself plus “a”

## Error handing techniques

If there is a dot (.) don’t add another on

if (!a.Contains("."))

{

a = ".0";

lbA.Text += a;

}

If lbA.Text is not a double don’t do nothing if it is a double make da equal it

if (double.TryParse(lbA.Text, out da) == false)

{

return;

}

If label “a“ is 90 then present invalid for tan else show tan

if (da == 90)

{

tbResult.Text = "Invalid";

}

else

{

tbResult.Text = Maths.Trigonometric.Tan(da).ToString();

}

# Recommended testing procedure (how should this software be tested before commercial release)

Push all buttons to make sure nothing breaks if it fails fix code and try again

If all buttons work compare results with a calculator to make sure all results are the same

# Recommendations on upgrades and future enhancements

For upgrading purposes add more buttons and more mathematical functions to calculator