LOGIC PageCalculate

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace McKillopCalculator
{
    /// <summary>
    /// Interaction logic for PageCalculate.xaml
    /// </summary>
    public partial class PageCalculate : Page
    {
        public PageCalculate()
        {
            InitializeComponent();
            ClearControls();
        }

        private void AddButton_Click(object sender, RoutedEventArgs e)
        {
            var calculationData = new CalculationData();
            calculationData = HarvestData();

            var result = Arithmetic.Addition(calculationData.FirstNumber,
calculationData.SecondNumber);
            ResultTextBlock.Text = result.ToString("#.####");
        }

        private void SubtractButton_Click(object sender, RoutedEventArgs e)
        {
            var calculationData = new CalculationData();
            calculationData = HarvestData();

            var result = Arithmetic.Subtraction(calculationData.FirstNumber,
calculationData.SecondNumber);

            ResultTextBlock.Text = result.ToString("#.####");
        }

        private void MultiplyButton_Click(object sender, RoutedEventArgs e)
        {
            var calculationData = new CalculationData();
            calculationData = HarvestData();

            var result = Arithmetic.Multiplication(calculationData.FirstNumber,
calculationData.SecondNumber);

            ResultTextBlock.Text = result.ToString("#.####");
        }

        private void DivideButton_Click(object sender, RoutedEventArgs e)
```

```
        {
            var calculationData = new CalculationData();
            calculationData = HarvestData();

            var result = Arithmetic.Division(calculationData.FirstNumber,
calculationData.SecondNumber);

            ResultTextBlock.Text = result.ToString("#.####");
        }

        private void ClearButton_Click(object sender, RoutedEventArgs e)
        {
            ClearControls();
        }

        private void ExitButton_Click(object sender, RoutedEventArgs e)
        {
            Application.Current.Shutdown();
        }


        // -- Utility
        private void ClearControls()
        {
            // Control
            FirstNumberTextBox.Text = "";
            SecondNumberTextBox.Text = "";
            ResultTextBlock.Text = "Result";
            // -- Focus
            FirstNumberTextBox.Focus();
        }

        private CalculationData HarvestData()
        {
            //-- Object to hold data while being gathered
            var tempData = new CalculationData();

            //-- Variables to hold text box string values and assign with
            //-- current values
            var firstNumberText = FirstNumberTextBox.Text;
            var secondNumberText = SecondNumberTextBox.Text;

            //-- Number variables to hold values for CalculationData object
            double firstNumber;
            double secondNumber;

            //-- Check there is data
            //-- This method has a weakness! If the user doesn't enter
            //-- text that can be converted to a number.
            if (!string.IsNullOrEmpty(firstNumberText))
            {
                try
                {
                    //-- Convert to a number
                    firstNumber = double.Parse(firstNumberText);
                    tempData.FirstNumber = firstNumber;
                }
                catch (FormatException e)
                {
                    //-- Tell the user what happened as feedback
                    MessageBox.Show(e.Message);
                }
```

```
            }
            else //-- Error
            {
                MessageBox.Show("You must enter a value for First Number");
                //-- drop out of the method
                return tempData;
            }

            //-- Let's try a better method for the second number. This is
sophisticated code!
            if (Double.TryParse(secondNumberText, out secondNumber))
            {
                //-- value is good to go so assign to object
                tempData.SecondNumber = secondNumber;
            }
            else //-- Error
            {
                //-- Tell them the bad news
                MessageBox.Show($"Could not convert {secondNumberText} to a
number");
                //-- Break out of the method
                return tempData;
            }

            return tempData;

        }
    }
}
```