



HND Calculator Application

Development Handbook

Contents

Development Handbook.....	1
Application overview	3
Design Sketches	4
Page Calculate.....	4
Design Thoughts.....	5
Expansion Concepts	6
UML Diagram	7
Flowchart Diagram.....	8
Entity list	9
Interface Designs	10
Test Plan.....	11
Evidence of running application	12
Test Log	18
APPENDIX A.....	19
XAML Code.....	19
MainWindow.....	19
PageCalculate.....	20
APPENDIX B.....	23
C# Code	23
CLASS Arithmetic.....	23
CLASS CalculationData	24
LOGIC MainWindow.....	25
LOGIC PageCalculate.....	26
Development Video List.....	29
Reference list	29
References and links	30
GitHub Repository.....	30

Application overview

This application will provide a simple calculator for arithmetic operations:

- Addition
- Subtraction
- Multiplication
- Division

The application will be developed using The **Windows Presentation Framework (WPF)**. This provides the tooling to create interfaces using **eXtensible Application Markup Language (XAML)**. The application logic will be implemented in Object Oriented Programming using C# as its language.

The application will be built using **Microsoft Visual Studio Community 2022**.

Future expansion is considered. The intent would be to provide application features that allow the user to convert a small range of value types from the Imperial to Metric measurement systems, and also in the opposite direction. This could include the ability to copy a value between the two calculation screens.

Design Sketches

Page Calculate

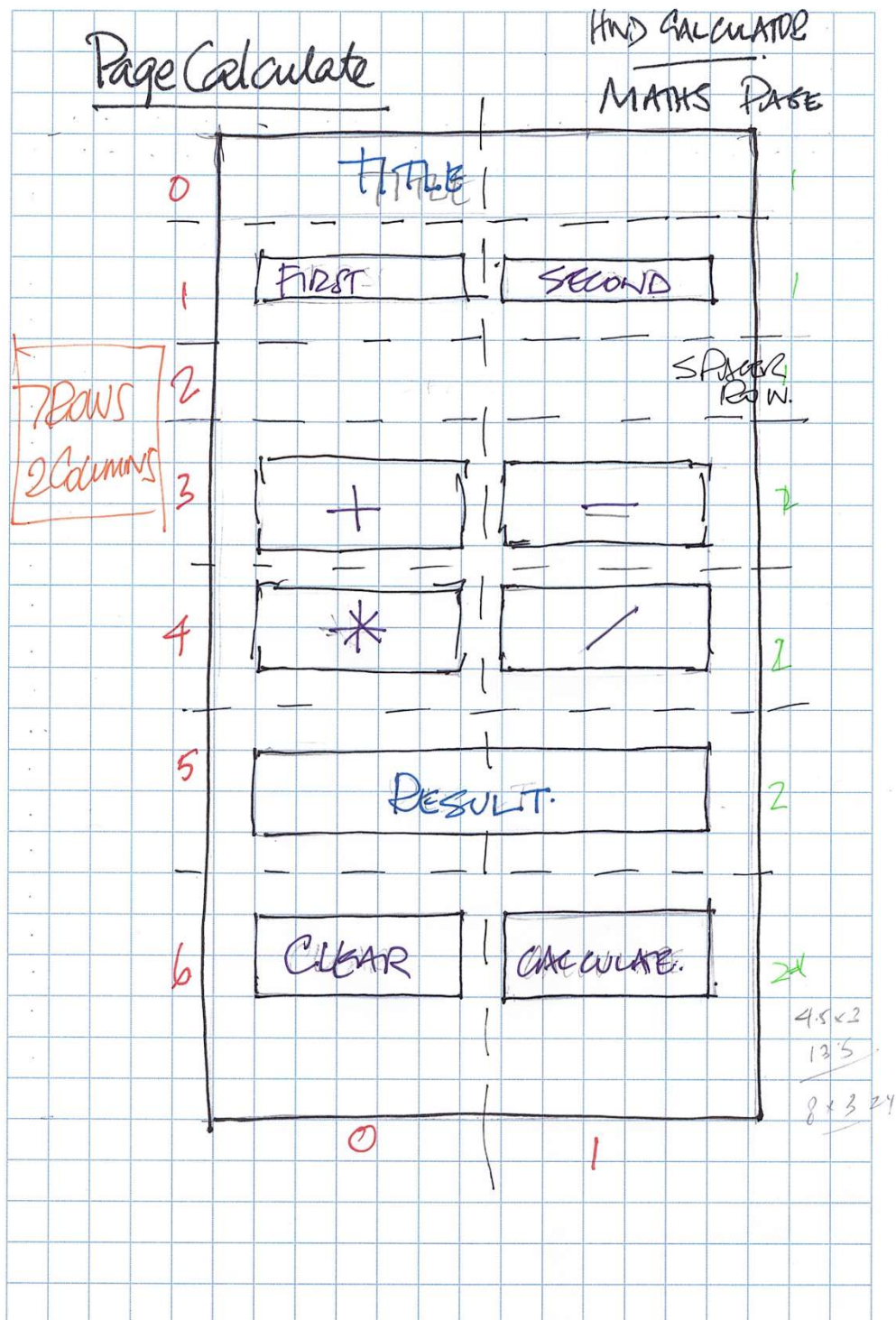
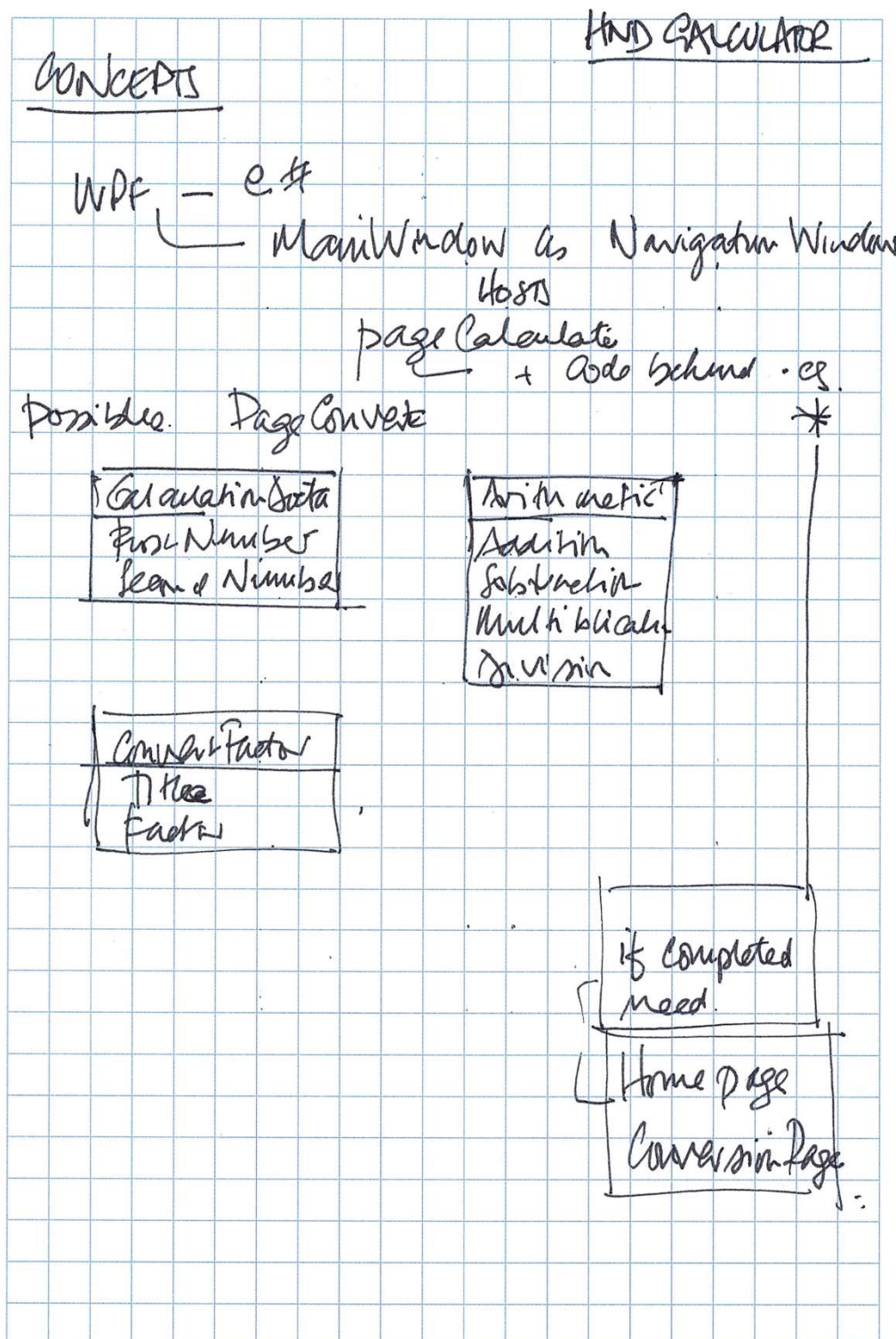


Figure 1 PageCalculate Sketch

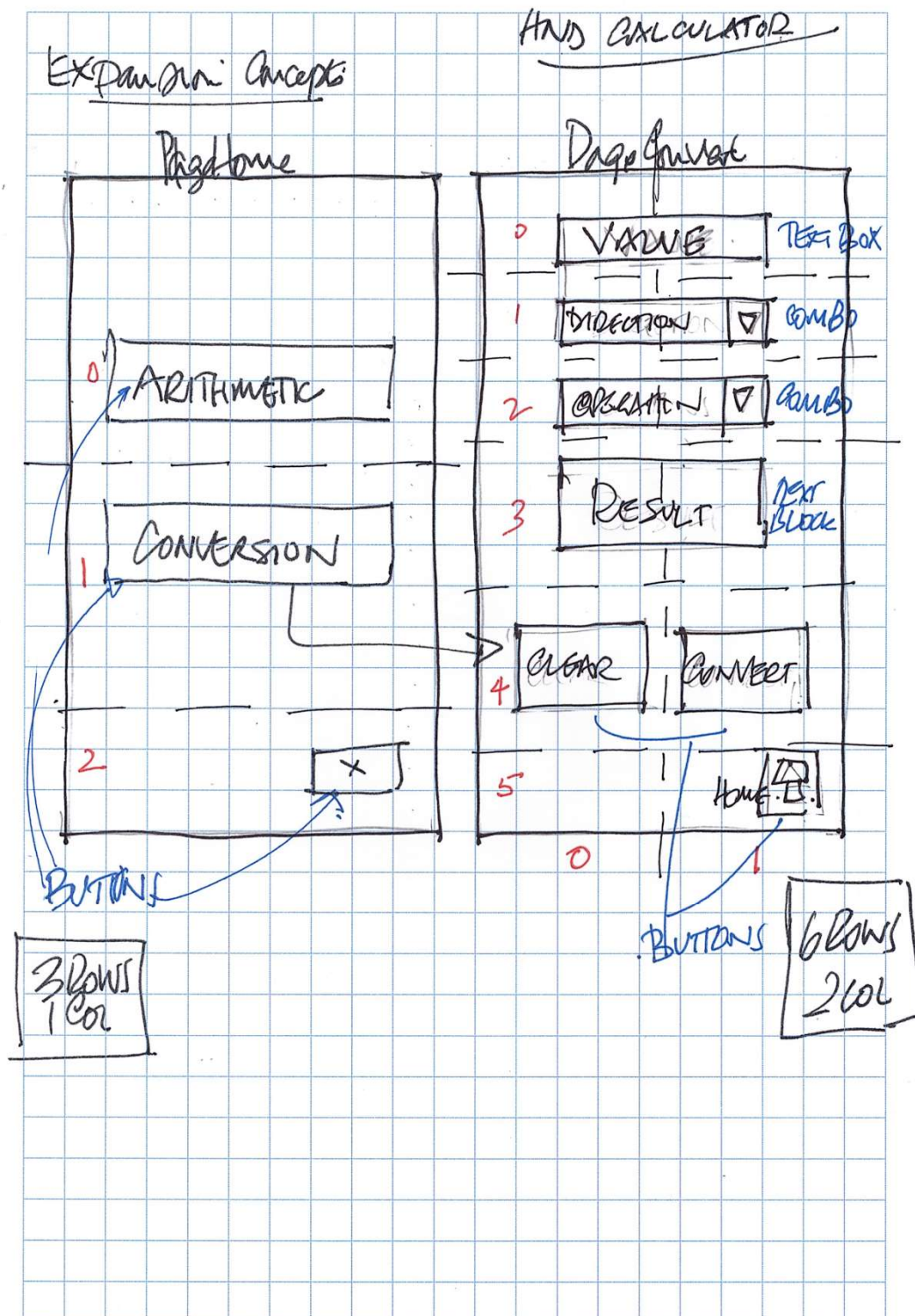
Design Thoughts



Free Plain Graph Paper from <http://incompetech.com/graphpaper/plain/>

Figure 2 Design thoughts

Expansion Concepts



Free Plain Graph Paper from <http://incompetech.com/graphpaper/plain/>

Figure 3 Expansion Concepts

UML Diagram

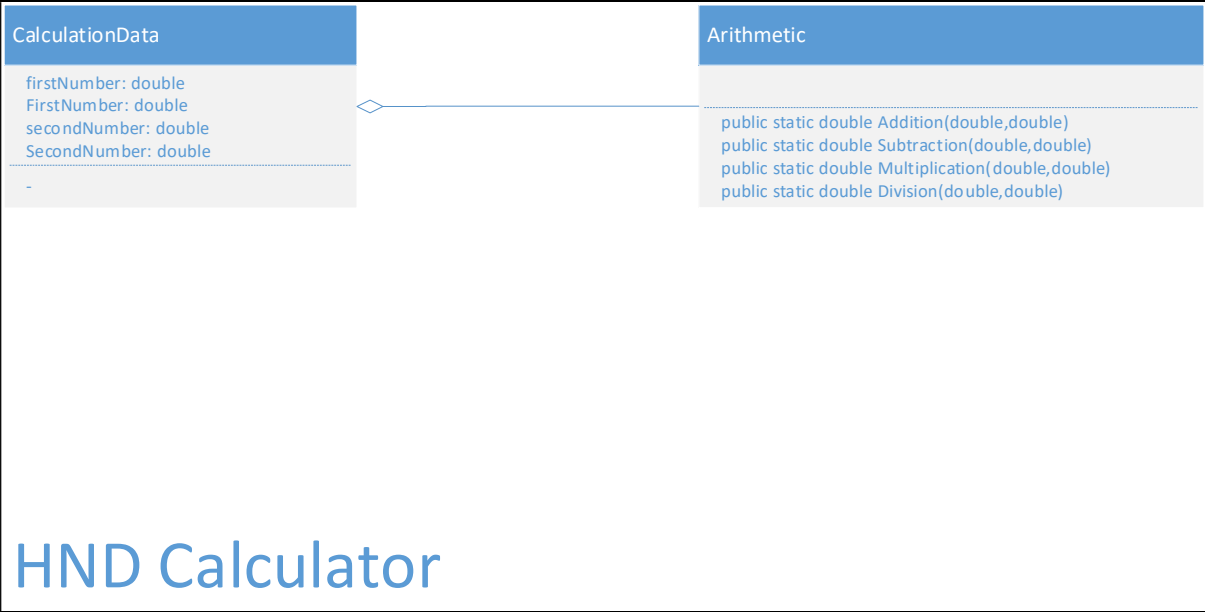


Figure 4 UML Diagram

Flowchart Diagram

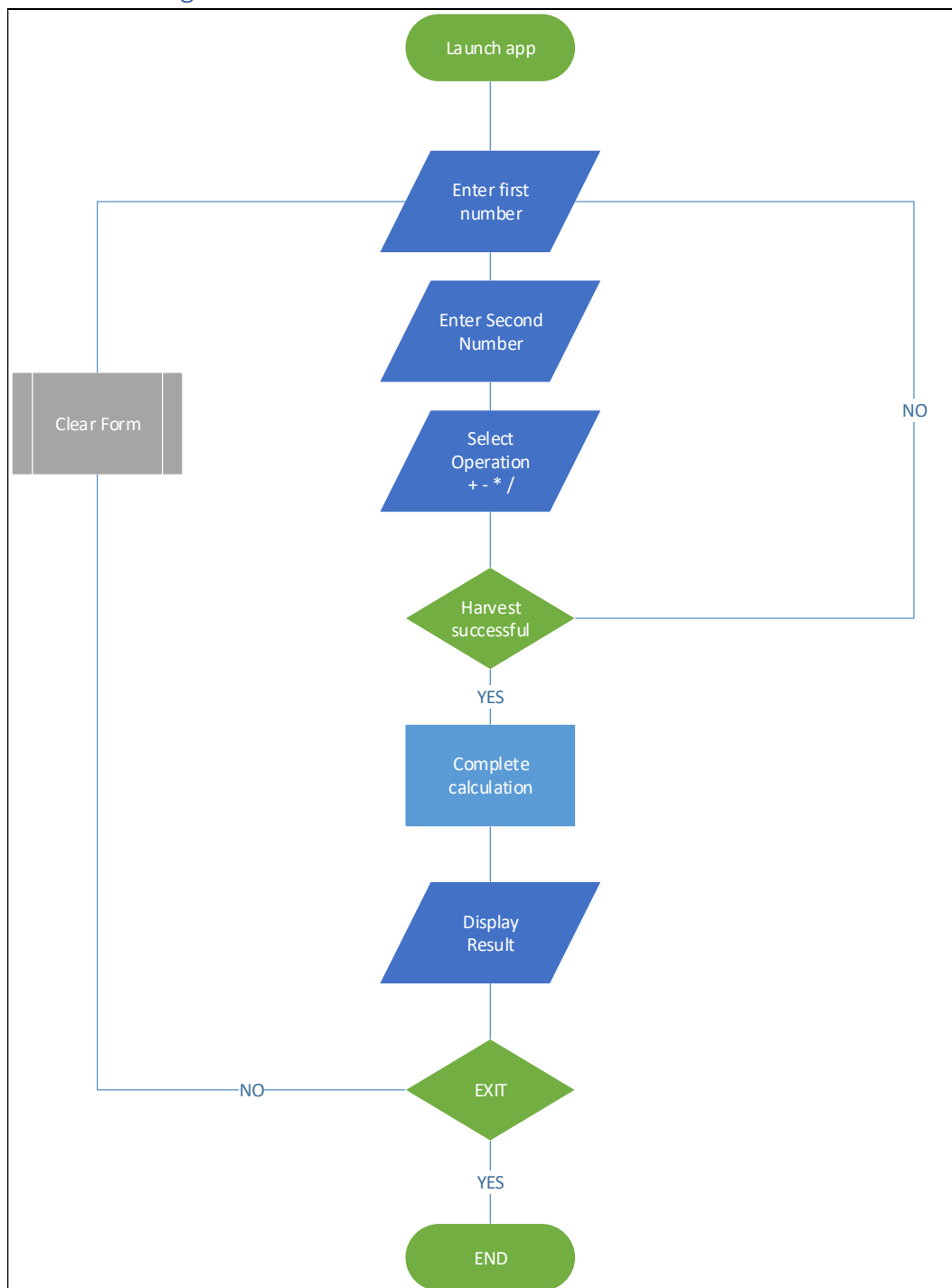


Figure 5 Flowchart application algorithm

Entity list

- CLASS **Arithmetic**
- CLASS **CalculationData**
- NavigationWindow: **MainWindow**: Navigation window to host pages
- PAGE: **PageAbout**: Program information page
- PAGE: **PageCalculate**: Process interface page

The image shows a Visual Studio design of a calculator interface. The design is a rectangular window with a light gray background and a blue border. It is divided into several sections by horizontal and vertical lines. At the top, the title "Calculator" is centered in a large, bold, blue font. Below the title, there are two input fields: "First Number" on the left and "Second Number" on the right, both in a bold, blue font. Below these input fields, there are four operation buttons: "Add" (yellow), "Subtract" (cyan), "Multiply" (red), and "Divide" (green). Below the operation buttons, there is a large, light blue rectangular box labeled "Result" in a bold, blue font. At the bottom, there are two buttons: "CLEAR" and "EXIT", both in a bold, blue font. The design includes dimension lines on the left side indicating the height of various sections: 3* for the title bar, 5* for the input fields, 2* for the operation buttons, 5* for the result box, and 7* for the bottom buttons. The width of the input fields and operation buttons is marked as 1*.

Calculator	
First Number <input type="text"/>	Second Number <input type="text"/>
Add	Subtract
Multiply	Divide
Result	
CLEAR	EXIT

Visual Studio

Figure 6 PageCalculate design

Test Plan

Item Number	Item	Test	Method	Expected outcome
001	Test app launch	Launch app	Launch in Visual Studio	Launch with PageCalculate
002	Control layout	Control positions and colours	Launch in Visual Studio	All controls in correct positions in form
003	Clear	Clear controls with button	Click CLEAR button	First and second numbers cleared
004	Clear to focus	Focus goes to FirstNumber	Click CLEAR button	Cursor focus in First Number control
005	Exit app	Make sure app shuts down	Click EXIT button	Application closes
006	ETC.	ETC	ETC	ETC

Evidence of running application

The screenshot shows a window titled "McKillop Calculator". Inside, the title "Calculator" is centered at the top. Below it are two input fields: "First Number" containing "25.74" and "Second Number" containing "16.59". Between the input fields and the result field are four operation buttons: "Add" (yellow), "Subtract" (cyan), "Multiply" (red), and "Divide" (green). Below these is a large light blue "Result" field. At the bottom are two grey buttons: "CLEAR" and "EXIT".

Figure 7 Data Entry

The screenshot shows a window titled "McKillop Calculator". Inside, the word "Calculator" is displayed in a large, bold, blue font. Below it, there are two input fields: "First Number" containing "25.74" and "Second Number" containing "16.59". Below these are four operation buttons: "Add" (yellow), "Subtract" (cyan), "Multiply" (red), and "Divide" (green). A large light blue display box shows the result "42.33". At the bottom are "CLEAR" and "EXIT" buttons, and the text "Addition Test" is centered below them.

First Number	Second Number
25.74	16.59

Add Subtract

Multiply Divide

42.33

CLEAR EXIT

Addition Test

Figure 8 Addition

The screenshot shows a window titled "McKillop Calculator" with a standard Windows title bar. The application has a light gray background and a blue title "Calculator" at the top. Below the title, there are two input fields labeled "First Number" and "Second Number". The "First Number" field contains the value "25.74" and the "Second Number" field contains "16.59". Below these fields are four operation buttons: "Add" (yellow), "Subtract" (cyan), "Multiply" (red), and "Divide" (green). The "Subtract" button is highlighted with a blue border. Below the operation buttons is a large light blue display field showing the result "9.15". At the bottom, there are two buttons: "CLEAR" and "EXIT", both in gray. Below these buttons is the text "Subtraction Test".

First Number	Second Number
25.74	16.59

Add Subtract

Multiply Divide

9.15

CLEAR EXIT

Subtraction Test

Figure 9 Subtraction

The screenshot shows a window titled "McKillop Calculator". Inside, the title "Calculator" is centered at the top. Below it are two input fields: "First Number" with the value "25.74" and "Second Number" with the value "16.59". There are four operation buttons: "Add" (yellow), "Subtract" (cyan), "Multiply" (red), and "Divide" (green). The "Multiply" button is highlighted with a blue border. Below the operation buttons is a large light blue display box showing the result "427.0266". At the bottom are "CLEAR" and "EXIT" buttons, and the text "Multiply Test" is centered below them.

First Number	Second Number
25.74	16.59

Add Subtract

Multiply Divide

427.0266

CLEAR EXIT

Multiply Test

Figure 10 Multiplication

The screenshot shows a window titled "McKillop Calculator" with a standard Windows-style title bar (minimize, maximize, close buttons). The application has a white background and a blue title "Calculator" at the top. Below the title, there are two input fields: "First Number" containing "25.74" and "Second Number" containing "16.59". Below these are four colored buttons: a yellow "Add" button, a cyan "Subtract" button, a red "Multiply" button, and a green "Divide" button. Below the buttons is a large light blue display box showing the result "1.5515". At the bottom, there are two grey buttons: "CLEAR" and "EXIT". Below the "EXIT" button is the text "Divide Test".

First Number	Second Number
25.74	16.59

Add Subtract

Multiply Divide

1.5515

CLEAR EXIT

Divide Test

Figure 11 Division

The screenshot shows a window titled "McKillop Calculator" with standard Windows window controls (minimize, maximize, close). The main content area has a title "Calculator" in blue. Below it are two input fields labeled "First Number" and "Second Number". There are four operation buttons: "Add" (yellow), "Subtract" (cyan), "Multiply" (red), and "Divide" (green). Below these is a large light blue button labeled "Result". At the bottom are two grey buttons labeled "CLEAR" and "EXIT". Below the "EXIT" button is the text "CLEAR Test".

Calculator

First Number **Second Number**

Add **Subtract**

Multiply **Divide**

Result

CLEAR **EXIT**

CLEAR Test

Figure 12 Clear controls

Test Log

Item Number	Item	Test	Method	Expected outcome	Date	Actual Outcome	Action taken
001	Test app launch	Launch app	Launch in Visual Studio	Launch with PageCalculate	28/10/2022	Lunch function OK	NA
002	Control layout	Control positions and colours	Launch in Visual Studio	All controls in correct positions in form	28/10/2022	Layout as planned	NA
003	Clear	Clear controls with button	Click CLEAR button	First and second numbers cleared	28/10/2022	Controls are cleared	NA
004	Clear to focus	Focus goes to FirstNumber	Click CLEAR button	Cursor focus in First Number control	28/10/2022	Focus not set	Method fixed and works
005	Exit app	Make sure app shuts down	Click EXIT button	Application closes	28/10/2022	Application closes	NA
006	ETC.	ETC	ETC	ETC			

APPENDIX A

XAML Code

Interface code

MainWindow

```
<NavigationWindow x:Class="McKillopCalculator.MainWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:local="clr-namespace:McKillopCalculator"
    mc:Ignorable="d"
    Title="McKillop Calculator"
        Height="800" Width="450"
        Source="PageCalculate.xaml"
        WindowStartupLocation="CenterScreen">
</NavigationWindow>
```

PageCalculate

```
<Page x:Class="McKillopCalculator.PageCalculate"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
      xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
      xmlns:local="clr-namespace:McKillopCalculator"
      mc:Ignorable="d"
      d:DesignHeight="800" d:DesignWidth="450"
      Title="Calculate" ShowsNavigationUI="False">

    <Grid Margin="5">
        <!-- Define rows and columns -->
        <Grid.RowDefinitions>
            <RowDefinition Height="3*" />
            <RowDefinition Height="5*" />
            <RowDefinition Height="2*" />
            <RowDefinition Height="5*" />
            <RowDefinition Height="5*" />
            <RowDefinition Height="5*" />
            <RowDefinition Height="7*" />
        </Grid.RowDefinitions>
        <Grid.ColumnDefinitions>
            <ColumnDefinition />
            <ColumnDefinition />
        </Grid.ColumnDefinitions>

        <!-- Title row index 0 -->
        <StackPanel
            Grid.ColumnSpan="2"
            HorizontalAlignment="Center">
            <TextBlock
                Style="{StaticResource PageHeaderStyle}">
                Calculator
            </TextBlock>
        </StackPanel>

        <!-- Data entry of numbers Row Index 1 -->
        <!-- First number data entry -->
        <StackPanel
            Grid.Row="1"
            Orientation="Vertical">
            <TextBlock
                Style="{StaticResource ControlLabelStyle}"
                TextAlignment="Center">
                First Number
            </TextBlock>
            <TextBox
                Style="{StaticResource ValueTextBox}"
                x:Name="FirstNumberTextBox">

        </TextBox>

        </StackPanel>

        <!-- Second number data entry -->
        <StackPanel
            Grid.Row="1"
            Grid.Column="1"
            Orientation="Vertical">
            <TextBlock
                Style="{StaticResource ControlLabelStyle}"
                TextAlignment="Center">
```



```
        Second Number
    </TextBlock>
    <TextBox
        Style="{StaticResource ValueTextBox}"
        x:Name="SecondNumberTextBox">

    </TextBox>

</StackPanel>

<!-- Spacer/Spare Row Index 2 -->

<!-- Buttons Addition Subtraction Row Index 3 -->
    <Button
        Grid.Row="3"
        Style="{StaticResource BigButton}"
        Background="Yellow"
        x:Name="AddButton"
        Click="AddButton_Click">
        Add
    </Button>

    <Button
        Grid.Row="3"
        Grid.Column="1"
        Style="{StaticResource BigButton}"
        Background="Aqua"
        x:Name="SubtractButton"
        Click="SubtractButton_Click">
        Subtract
    </Button>

<!-- Buttons Multiplication Division Row Index 4 -->

    <Button
        Grid.Row="4"
        Style="{StaticResource BigButton}"
        Background="LightCoral"
        x:Name="MultiplyButton"
        Click="MultiplyButton_Click">
        Multiply
    </Button>

    <Button
        Grid.Row="4"
        Grid.Column="1"
        Style="{StaticResource BigButton}"
        Background="LightGreen"
        x:Name="DivideButton"
        Click="DivideButton_Click">
        Divide
    </Button>

<!-- Display result Row Index 5 -->
    <StackPanel
        Grid.Row="5"
        Grid.ColumnSpan="2"
        Margin="10">
        <Border
            Margin="5"
            Padding="5"
            Background="LightBlue"
            BorderBrush="DarkBlue"
```

```
        BorderThickness="3"
        CornerRadius="5">
        <TextBlock
            Background="AliceBlue"
            VerticalAlignment="Center"
            HorizontalAlignment="Stretch"
            TextAlignment="Center"
            Foreground="DarkBlue"
            FontFamily="TW Cen MT"
            FontSize="36"
            x:Name="ResultTextBlock">Result</TextBlock>
        </Border>
    </StackPanel>
<!-- Buttons Clear and Calculate Row Index 6-->
    <StackPanel
        Grid.Row="6"
        Grid.ColumnSpan="2">
        <Grid>
            <Grid.ColumnDefinitions>
                <ColumnDefinition/>
                <ColumnDefinition/>
            </Grid.ColumnDefinitions>

            <Button
                Style="{StaticResource FormButton}"
                x:Name="ClearButton"
                Click="ClearButton_Click">
                CLEAR
            </Button>

            <Button
                Grid.Column="1"
                Style="{StaticResource FormButton}"
                x:Name="ExitButton"
                Click="ExitButton_Click">
                EXIT
            </Button>
        </Grid>
    </StackPanel>

</Grid>

</Page>
```

APPENDIX B

C# Code

CLASS Arithmetic

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

/*
 * Title:   Arithmetic
 * Author:  Paul McKillop
 * Date:    October 2022
 * Purpose: Arithmetic methods - Method only class
 */

namespace McKillopCalculator
{
    public class Arithmetic
    {
        // -- Calculation rules for subtraction and division:
        // --- first - second
        // --- first / second

        public static double Addition(double first, double second)
        {
            return first + second;
        }

        public static double Subtraction(double first, double second)
        {
            return first - second;
        }

        public static double Multiplication(double first, double second)
        {
            return first * second;
        }

        public static double Division(double first, double second)
        {
            return first / second;
        }
    }
}
```

CLASS CalculationData

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

/*
 * Title: CalculationData
 * Author: Paul McKillop
 * Date: October 2022
 * Purpose: Hold data for calculations
 */

namespace McKillopCalculator
{
    public class CalculationData
    {
        // -- Data member for first number
        private double firstNumber;

        public double FirstNumber
        {
            get { return firstNumber; }
            set { firstNumber = value; }
        }

        private double secondNumber;

        public double SecondNumber
        {
            get { return secondNumber; }
            set { secondNumber = value; }
        }
    }
}
```

Paul McKillop
HND Calculator

LOGIC MainWindow

```
using System.Windows.Navigation;

namespace McKillopCalculator
{
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    public partial class MainWindow : NavigationWindow
    {
        public MainWindow()
        {
            InitializeComponent();
        }
    }
}
```

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;
}
}
```

Development Video List

Reference list

« R:\22-23 HND Applications\040 Calculator\07 Videos to publish		
Name	Extension	Size
 040 01 What we will build.mp4	mp4	18,798,551
 040 02 Create the WPF Application.mp4	mp4	62,827,245
 040 03 XAML Styles.mp4	mp4	34,877,597
 040 04 Class CalculationData.mp4	mp4	22,523,504
 040 05 Class Arithmetic.mp4	mp4	17,991,567
 040 06 GUI PageCalculate Part 1.mp4	mp4	35,063,152
 040 07 GUI PageCalculate Part 2.mp4	mp4	64,648,234
 040 08 GUI PageCalculate Part 3.mp4	mp4	31,804,922
 040 09 LOGIC PageCalculate Part 1.mp4	mp4	22,292,838
 040 10 LOGIC PageCalculate Part 2.mp4	mp4	25,609,421
 040 11 LOGIC PageCalculate Part 3.mp4	mp4	21,288,214

References and links

The complete WPF tutorial. (2022). Retrieved 28 October 2022, from <https://wpf-tutorial.com/>

Windows Presentation Foundation for .NET 5 documentation. Windows Presentation Foundation for .NET 5 documentation | Microsoft Learn. Retrieved October 28, 2022, from <https://learn.microsoft.com/en-us/dotnet/desktop/wpf/?view=netdesktop-6.0>

GitHub Repository

<https://github.com/pgmckillop/HND-Calculator>