# About Calculator Project

Project specifications are provided in file named ‘C-SharpCalculatorTest[656].doc’.

Framework used: .Net framework 4.5.2

## The solution contains two (2) projects

### 1. SimpleCalc:

The Application is a Very simple Calculator program, the finished project is a WPF MVVM application using C# language.

* **Views**:
* CalculatorView is the default and the only view available in application this also contains the main application interface. Although screen currently supports limited digits and calculations but all digits can be easily supported by adding new buttons similar to existing buttons to number penal, by copy-paste existing and changing CommandParameter, and new command buttons for arithmetic operations can also be added similarly.
* **ViewModels**:
* CalculatorViewModel provides the data and behavior for the CalculatorView. It inherits from ViewModelBase which provides INotifyPropertyChanged implementation for any view model class choose to inherits from it. ViewModel exposes listed commands and properties.
* NumericInputCommand: This command can be used on numeric buttons to append data to current expression. CommandParameter can be any digit from 0-9 or decimal (.).
* OperatorCommand: This command can be used to select arithmetic logic to be applied to two operands. CommandParameter can be any of the selected arithmetic operation currently it supports plus (+), minus (-), divide (/), multiply (\*). But screen currently lists just two (+ and /).
* ClearExpressionCommand: This can be used to Clear contents of calculator display.
* CalculateCommand: This command is used by “=” (equals) button to calculate the result of the current expression.
* **Models**:
* ICalculator provides the blueprint for the calculator class. This currently requires just one function named Calculate.
* Calculator.cs class implements the ICalculator interface and provides the code logic to the application as per the expectations described in specifications document.
* **SimpleIoC**: This class provides a very simple implementation of dependency injector container, currently it just supports registering types and resolve dependencies to provide single instance across application, this behavior can be changed.

### 2. SimpleCalcTests:

* Contains Unit Tests to verify the expected output from the calculator program as per the instructions given the specification document.