**Software Design Document**

Application Name: Project Week 5 – Simple Calculator with Exception Handling  
Developer: Eric Martinez

**What classes will the program use?**

* CalculatorApp – This is the main part of the program that runs everything. It shows the menu, takes user choices, and uses other classes to do the work.  
    
  Main() – Runs the main program loop  
  Multi-number menu added to allow operations on more than 2 numbers
* Calculator – Handles the math. It performs addition, subtraction, multiplication, division, and modulus.  
  Now supports operations on multiple inputs like Add(int[]), Divide(double[]), etc.
* Memory- Handles the memory features. It stores a single value and a list of up to 10 integers. It lets the user store, view, update, or delete values in memory.  
  Now supports storing multiple values at once (up to 10) in one entry.
* **InputValidator** – Makes sure the user types in valid numbers or menu options. Exception handling to ensure the program does not crash on invalid input. It gracefully manages division-by-zero scenarios and invalid number formats during user input, allowing the user to reattempt input until it's valid.

What functions are in the program?  
  
Added support for multi-number input operations.

In CalculatorApp  
  
Main() – Runs the main program loop  
Multi-number menu added to allow operations on more than 2 numbers

* Main() – Starts the program and keeps it running
* ShowMenu() – Displays the math options
* HandleChoice() – Uses the user’s input to pick the right calculation or memory feature

**In Calculator**

* Add(int a, int b) – Adds two whole numbers
* Subtract(double a, double b) – Subtracts two decimal numbers
* Multiply(int a, int b) – Multiplies two whole numbers
* Divide(double a, double b) – Divides two decimal numbers (includes divide-by-zero check)
* Modulus(int a, int b) – Finds the remainder (also checks for divide-by-zero)

In MemoryManager

* StoreSingle(double value) – Stores a single number in memory
* RetrieveSingle() – Returns the stored value
* ClearSingle() – Clears the single stored value
* ReplaceSingle(double value) – Replaces the current stored value
* AddToList(int value) – Adds an integer to the list (max 10 values)
* RemoveFromList(int value) – Removes a value from the list
* DisplayList() – Shows all the stored list values
* CountList() – Shows how many values are in the list
* SumList() – Adds all numbers in the list
* AverageList() – Calculates the average of the list values
* DifferenceFirstLast() – Shows the difference between the first and last values

**In InputValidator**

* ReadInt(prompt) – Asks for a whole number and checks it's valid
* ReadDouble(prompt) – Asks for a decimal number and checks it's valid
* ReadMenuChoice() – Makes sure the user picks a valid menu option

How does the program flow?  
  
1. Shows a welcome message  
2. Repeatedly displays a main menu with options for math, memory, and multi-number operations  
3. Uses input validation to get user input  
4. Performs the selected operation using helper classes  
5. Displays results with proper formatting (2 decimal places where needed)  
6. Allows memory storage/retrieval with capped list size of 10  
7. Program continues until the user chooses to exit

1. Shows a welcome message and memory instructions
2. Shows the calculator and memory menu
3. The user picks a number for a calculation or memory feature
4. If math is selected, it asks for one or two numbers and uses calculator
5. If memory is selected, it asks what action to take (store, retrieve, etc.) and uses MemoryManager
6. Results are shown to the user
7. The user is asked if they want to continue
8. If yes, it loops. If not, it prints a thank you message and exits

**Controls and Error Handling**

* Uses switch to control menu choices
* Uses while loop to keep the program running
* Uses input validation to prevent crashes
* Checks for divide-by-zero
* Shows clear messages if something breaks.
* Makes sure the memory list never goes above 10 items
* Shows clear messages if something goes wrong