Design Document

Mathematical **E**xpression **C**alculator

10/30/2017

CS456

Version 1.0

Erika Vazquez

Chris Perry

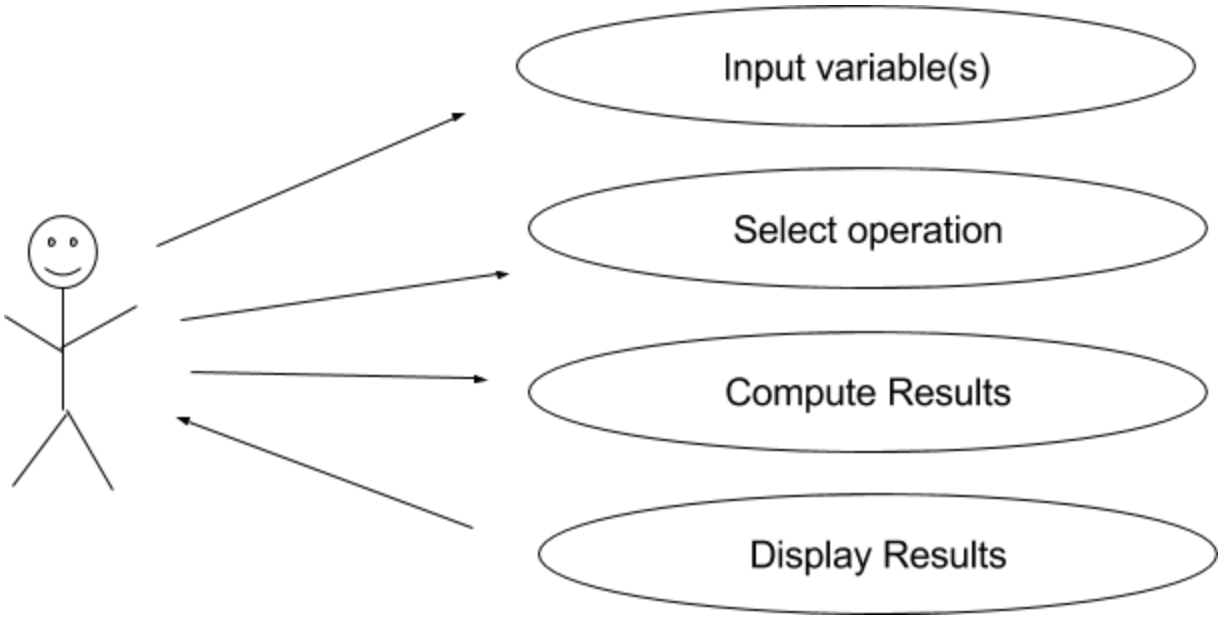
Abdullah Aldossary

Thamer Seyab

General Design Guidelines

- ➢ With the exception of overflow and underflow, no user input should cause the program to terminate abnormally
- ➢ The specification details minimum requirements for our program. As long as these requirements are met, any additional features/function may be added.
- ➢ The source language to be used is JAVA. Any system-dependent features shall be avoided if at all possible.
- ➢ Our program shall be usable on any system which supports the JAVA compiler, and shall not require any particular hardware or software.

Use Case Model

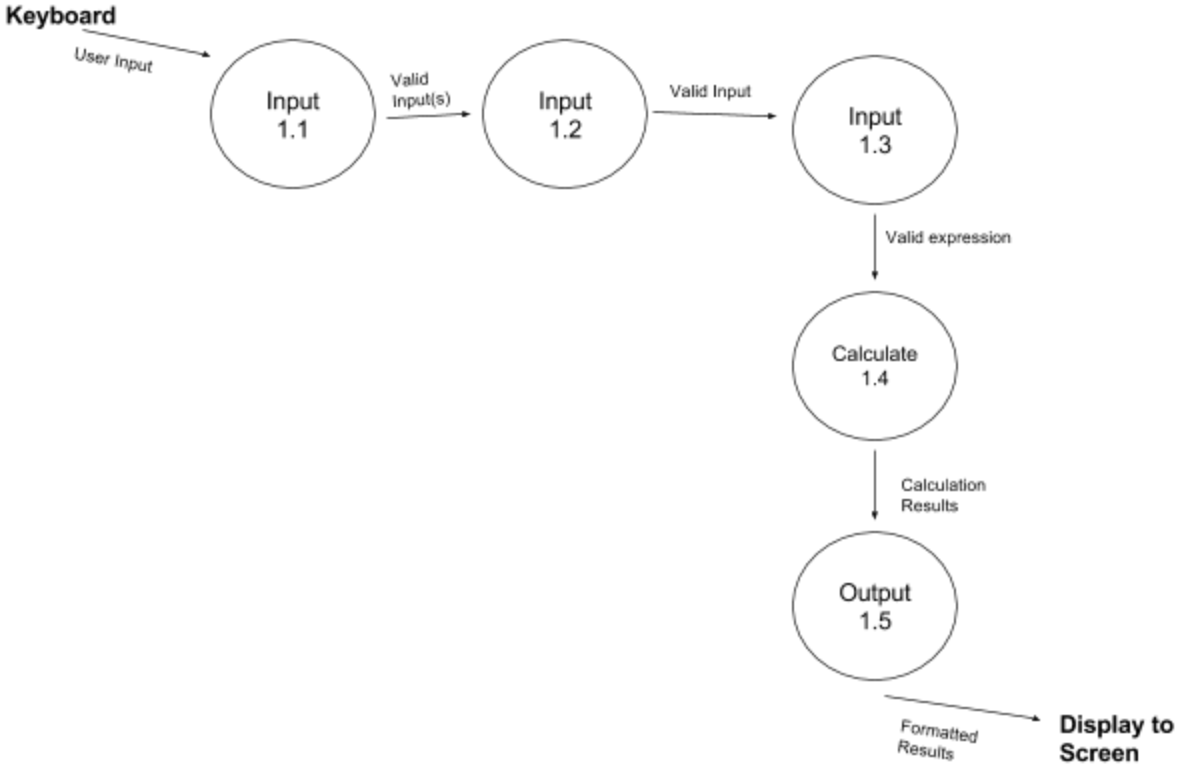


Data Flow Diagram

Data (the mathematical expression - its components) comes from the users of the calculator and the answer to it eventually flows back to the user once the calculation of the expression is completed.

Data flows from the user to the screen, where input is given. Once the number of variables to be used is entered, the application validates that it is a number less than or equal to 4. If the number is valid, the program will ask user to enter numerical values for each variable one at a time. If it is not a valid number (0-4), program will display error message and ask user to type a valid number. Once program has numbers stored in variable(s), it will ask user to enter a mathematical expression that consists of the variables. Program will do necessary calculation and provide the user with a correct and accurate answer to the math expression.

Data flow diagram that identifies the major functions required of the program and shows their relationship to each other. The numbers included with each function are keyed to the subsections below that describe that function.



Major Function Descriptions

Input 1.1 shall accept numerical value 0-4 from the user via the output window, validate the input, and use that input to determine how many variables the user will need. This is then sent to Input 1.2.

Ex)

Input: 3

Input 1.2 shall accept a numerical value from the user via output window, for each variable that the user selected in the previous input.

Ex)

Input: a: 3
b: 1
c: 2

Input 1.3 shall accept a mathematical expression from the user via output window, that may include variable(s) that the user previously assigned a value to. Valid mathematical expression includes the basic operations: addition, subtraction, multiplication, and division.

For **addition** the user must use **+** sign, for **subtraction -** sign, for **multiplication** use ***** sign, for **division** use **** sign. When inputting the expression, no spaces should be between numbers, variables, and operands. Also, no **=** sign to be typed, user should just press enter when the expression is typed and ready to be calculated.

Ex)

a+b
10-c
a*c
12/a

- If an illegal sequence is detected, an error message will be printed. User will be asked to input a valid typed expression, quit the program, or start over.
- To quit the program, user shall type Q or q in the input line.
- To start over, user shall type S or s in the input line.

Calculate 1.4 will accept correct input from Input 1.3 and will perform the arithmetic operation requested by that input. The four arithmetic functions are listed above.

- Division by zero should be detected by the program, and an error message should be displayed.
- Overflow and underflow do not need to be detected.
- The order of calculation need only proceed from left to right; no hierarchy placed on the order of calculations.

Output 1.5 will display the results of a calculation. Results should be displayed only when:
The ENTER key is pressed, after inputting the equation in Input 1.3.
Results shall be displayed in base 10.