Building a simple calculator

Now, this is just plain silly! Let’s write a program that pretends to be a calculator. Do not get worried about this, it has a few parts, but you can put it together without a lot of work. It looks more complex than it really is! Just read this description closely, and it should come together.

The basic program structure

In the description below, think about how a calculator works. (Surely, you have used such a beast somewhere in your life!) If not, on both the PC and the Mac, there is a *Calculator Program* that you can start up to see how it works. Our calculator will be pretty simple, read the description closely.

The main control loop

Your program basically be just a simple loop that asks the user what button they want to press. This “button” will be an integer number between 1 and 6 (see below). You need to display the list of buttons as the program starts, so the user will know what to do.

Each time a button is “pressed” by the user, your program should test that input number (using a decision statement) and do whatever is required by that button. Once it is done, you should print out the current value stored in a variable named “display” that we will describe below.

Calculator “memory”

Internally, we need to manage a few variables that will simulate the calculator’s “memory:

You will need to manage two variables in the main module in your program:

* one named “display” that holds the floating point number representing the calculator display.
* one named “button” which will hold a user input integer button number (see below).

What the buttons do

Our calculator will only have five buttons, plus one to stop the calculator (the program):

Calculator buttons

Each “button” is supposed to cause one simple thing to happen in the calculator. For example, your code might include these lines:

if(button == 1) {

 dout << "enter a number ";

 cin >> userval;

 display = display + userval;

}

Here are the buttons you will set up:

1. The plus key, which asks for a number from the user, than adds that number to the number in “display”.

2. The minus key which asks for a number from the user, then subtracts that number from the number in “display”.

1. The multiply key which asks for a number from the user, then multiplies that number by the number in “display”.
2. The divide key which which asks for a number from the user, then divides the number in “display by the number entered.

Finally, we have another button that will cause a prompt to be displayed asking the user to enter a number. That number will be stored in “entered.

1. The clear button will zero out the display

And the last button to exit the program.

1. The exit key which terminated the program

Example of using the calculator

Here is an example of how your calculator should work, showing the key number the user will enter, followed by the input number and resulting display. In this table, a dash means no input required from the user, or nothing is displayed because the calculator is “off”. All other user input is a floating point number:

| **number** | **key** | **user number** | **display** |
| --- | --- | --- | --- |
| 5 | clr |  | 0 |
| 1 |  | 123.0 | 123.0 |
| 2 |  | 5.0 | 118.0 |
| 3 |  | 2 | 236.0 |
| 4 | / | 2 | 118.0 |
| 5 | clr |  | 0 |
| 6 | exit |  |  |

The program would end after the six button was “pressed”

Here is an example of the output that sequence of keys should display on the screen. You can modify this as you wish:

Calculator program

Here are the keys on your calculator:

1 = add, 2 = subtract, 3 = multiply, 4 = divide, 5 = clear, 6 = exit

Enter your button: 5

Display shows: 0

Enter your button: 1

Enter the number to add: 123.0

Display shows: 123.0

Enter your button: 2

Enter the number to subtract: 5.0

Display shows: 118.0

And so on

Add another button:

1. The sqrt button which takes the number in “display” and replaces it with the square root of that number.

Use your calculator to find the value of SQRT(3\*3 + 4\*4). Type out (or copy and paste) the text displayed by your program as you evaluate this expression. Upload this text in a file named “extra.txt” with your final program source code on Blackboard.