```
1 /*****************************************************************************
2  * AUTHOR       : JOSHUA SALZEDO
3  * STUDENT ID   : 1094998
4  * Assignment #9: Intro to programming
5  * CLASS        : CS1A
6  * SECTION      : MW: 8AM
7  * DUE DATE     : 04/15/2018
8  *****************************************************************************/
9 #include "MyHeader.h"
10
11 /*****************************************************************************
12  * FUNCTION saneInputCinInt
13  * ---------------------------------------------------------------------------
14  * Gets a sanity checked int from the user
15  *  sanity check verifies the input is between (or equal to) the minimum
16  *          and the maximum.
17  * ---------------------------------------------------------------------------
18  * PRE-CONDITIONS
19  *      The following parameters must have defined values:
20  *          minimum,
21  *          maximum,
22  *          prompt
23  * POST-CONDITIONS
24         ==> returns input : sanity checked input
25  *****************************************************************************/
26 int SaneInputCinInt(int minimum,            // IN - minimum allowable value
27                     int maximum,            // IN - maximum allowable value
28                     const string &prompt)   // IN - user prompt
29 {
30     string fstring;     // CALC & OUT    - formatted string
31     bool valid;         // CALC          - input validity marker
32     int input;          // IN, CALC&  OUT- user input
33     do
34     {
35         cout << prompt;
36         cin >> input;
37
38         //process validity
39         valid = input >= minimum && input <= maximum;
40         if (!valid)
41         {
42             fstring += "Invalid input. Please enter a value between ";
43             fstring += to_string(minimum);
44             fstring += " and ";
45             fstring += to_string(maximum);
46             fstring += ".\n\n";
47             cout << fstring;
48         }
49     } while (!valid);
50
51     // flush the buffer
52     cin.ignore(1000, '\n');
53     // and return
54     return input;
55 }
56
57
```