**Arrays and Functions**

**Magic Squares**

A magic square is a square arrangement of positive integers such that all rows, all columns, and both diagonals add to the same sum. For example, here is a magic square:

8 3 4

1 5 9

6 7 2

Your assignment is to create a program to prompt the user for a filename that contains a proposed magic square, read the information from the file into a 3x3 two-dimensional array, and determine whether the arrangement is in fact a magic square. To get you started, [here](http://borax.truman.edu/180/lab10array/template.cpp) is the main function needed to make this program work. You are to use this exact main function, and not change it in any way.

Your program must produce output that looks like this:

Enter a filename for processing: foobar.txt

Could not open foobar.txt. Please try again.

Enter a filename for processing: goodsquare.txt

+---+---+---+

| 4 | 9 | 2 |

+---+---+---+

| 3 | 5 | 7 |

+---+---+---+

| 8 | 1 | 6 |

+---+---+---+

This is a valid magic square.

Your program is required to define and properly document exactly four functions described as follows:

1. validate\_square is a Boolean function that takes a 2-d array of unsigned values as a parameter and determines whether it is in fact a magic square by summing the rows, columns, and diagonals and verifying that they all total the same value
2. get\_input\_file has an ifstream reference parameter. The function prompts the user for a filename. If a file by that name can be opened for input, it is assigned to the parameter, still in its open state. The function must contain an input validation loop that continues to prompt until a valid filename is specified by the user. You should use the .fail() file method in this function, as described in the slides from class 17.
3. print\_square is a void function that takes a 2-d array of unsigned values as a parameter and outputs it to the screen as shown above
4. read\_square\_from\_file I have written this one for you, but you have to write the Javadoc for it.

Several items of particular note:

* All magic squares in the program and in the input files are strictly 3x3 squares that contain exactly the integers 1 through 9 and nothing else.
* Sometimes the array parameter in this program needs to be const, and other times not. It is very important that you get these right. Gaddis page 418 has information on this. The course slides also have information on all this.
* Be very explicit about the names of index variables, so that it is obvious which you consider the row and which the column. Do not use single-letter names such as r or c or, even worse, i or j.
* You should test your program with one or two *valid* magic squares and one or two *invalid* magic squares.