CSC 201 – Computer Science I

Madeline Navarro

9/13/2016

**Program #1 – Chess Board Implementations**

**1) Using a function to initialize a row using its colors**

I used two string variables as parameters, and these would be the colors of the row that the function initializes. The color lines and the border line were already initialized as constants, available for use as arguments. The function returns a string value; it is given two string parameters that correspond to the first color of the row and the second color, and it assigns the row string in an alternating pattern of the two colors until the row is filled to the board length. This function is more convenient because it can be used for not only both color rows but also the border row string variable. Instead of entering two color lines, the arguments for the border row initialization can be two border lines.

**2) Using a function to print a row until it forms a complete row of blocks**

This function prints out the amount of rows it takes to complete a block. This way, either color may be used as an argument to finish one row of complete chess squares for each color needed. This is made more convenient using a for loop, which is discussed in the next implementation.

**3) Using for loops to reduce cout statements**

In this program, there are many items that need to be printed out, but the repetition means that the use of a for loop in several instances can reduce the cout statements to even as few as three. First, to initialize a row, I used a for loop, because each colored row is an easily repeated pattern. Second, to print a row of complete chess squares, a for loop can again be used to print each row 5 times to make a whole block. The last use of the for loop is to print the correct amount of complete block rows to make an entire chess board. Four iterations of printing a white-beginning row of squares and then a black-beginning row completes the board vertically.