Problem 1:

Understanding:

I should say that when I looked at this problem, I was so confused. This problem looked identical to the one that we had last week. I had to get clarification on the discussion board. The difference between last week’s assignment and this week assignment is that we need to use functions. For this problem we need to write a simple function that accepts two strings and returns whether they have the same contents. We are allowed to reuse code from last week.

Design:

1. I’ve declared function: void stringCompare (string string1, string string2);
2. I went back to the last weeks code and moved code that involves string comparison to the function definition.
3. Finally, I called function stringCompare (string1, string2) in the main method.

Everything worked as expected. After all this was very straightforward problem.

Problem 2:

Understanding:

We need to write a function that acts like a random number generator after user enters min and max of the range. We are allowed to reuse last weeks code.

Design/Imprementation

1. I already have a code from last week, so my design and implementation will be done simulateously
2. Declare a function: int randInput(int min, int max);
3. Define function:

int randInput(int min1, int max1){

return rand() % (max1 - min1+ 1) + min1; // random will be in the range of min1-max1

////return myRand;

cout <<endl; // console output of the random number

}

1. While looking at my last week’s code for this problem, I’ve realized that I had a mistake in my random number generator. My code was not generating numbers in the range between min-max because I put extra brackets in the rand() % **(**(max1 - min1+ 1) + min1**)**. Because of the extra brackets ( in red color), I was generating random numbers between 0 and (max+1). This problem was fixed.
2. Next, I’ve called int randomNumber = randInput( min1, max1); in the main method
3. Looking at my code from last week, I saw that there was still a redundancy. The user is asked to enter 2 numbers, one more max and one for min. At the same time, I have a code for input error handling. Any time user enters a string, or some a character, they get an error message. The user needs to enter 2 numbers, the input error test is done x2, and thus there is a redundancy in the code. I’ve tried to come one with the way to move that part of the program into the function.
4. Using ideas from: <http://stackoverflow.com/questions/4604500/use-of-for-in-a-c-sharp-application>, the double getNumber() function was created to eliminate redundancy:

{

for(;;) // I found out that this is equivalent to while (true) {}

{

double value;

cin >> value;

if(cin.fail()) // if the cin failed

{

cin.clear(); // reset scin

string garbage; // declaration of the string this needs to be discarded

getline(cin, garbage); // ignores the rest of line

cout << "\nINVALID ENTRY. That wasn't a number, try again. ";

}

else

{

return value;

break;

}