**CIS 2275 C++ Programming Part II Name:**

**Program 1 The Color Swap 100 pts Due: Tuesday, February 2 by 11:59 pm**

# Review functions, pointers/references, multi-file construction, vectors, arrays and strings

**Turn In Requirements:**

1. **5 pts Name your Visual C++ 2015 project LastnameP1, such as NelsonP1.**
2. **5 pts Upload your project to Visual Studio Online. Be sure you add your three files as Resources so VSO won’t delete them.**
3. **5 pts If you upload to Blackboard, print out the \*.h and \*.cpp files, staple this page to the front of your printed source code when you turn it in for grading. 5 pts. Also, remove BOTH debug folders and the .sdf file from your project before uploading it to Blackboard.**

**Program Requirements:**

1. **3 pts Write your name, email address and file name at the top of your source code in a comment.**
2. **5 pts Your main function should have cout statements that write “header” information to the screen. The header info includes your name, course and program information, as well as a 1-2 line description of the program.**
3. **5 pts. Use good C++ programming style and formatting for your program. Use appropriate comments to explain what you are doing.**

**Program Requirements:**

The goal of this program is to read in several sentences. If the sentence contains a color--that is found in the Colors.txt file, that color is replaced with the corresponding phrase found in the Phrases.txt file.

Colors.txt contains names of colors. The names are only one word, such as red, blue, etc. Sentences.txt contain up to 10 sentences, that \*may\* contain at least one color in it that is found in the Colors.txt file. SwapPhrases.txt contains short phrases that will correspond to the colors in Colors.txt. There is one phrase for each color.

So there are three files to be read: Colors (7 of them), sentences ( up to 10) and phrases (7, to substitute for the colors). I will provide the three files, but you may write your own files and name them accordingly.

The main function starts with displaying the class *Header*, either as a cout from main or by calling a function. Main contains the variable declarations:

int numSwapped[SIZE];

vector<string> vColors;

string sentences[SIZE];

vector<string> vPhrases;

plus other variables nee3ded to keep track of items. Main also contains the default filename assignments into strings,

string colorFile = "Colors.txt";

string sentenceFile = "Sentences.txt";

string swapPhrasesFile = "SwapPhrases.txt";

string outputFile = "Out.txt"; //or whatever you like here

and all of the calls to the functions. Also, it has the output file ofstream object declared and the output file opened and checked.

Ask the user if he/she wants to enter a filename for each file or if he/she wants to use the default value. Don’t make the user type in all of the filenames if he/she wants to use the default. You may do this in main, or in a function (or more than one).

Call the *Read* function for each file. Pass in the vectors by reference, and pass in the array as usual. In main, check the bool return from each *Read* function to make sure the file was opened and read. If not, tell the user and exit the program.

Call the *WriteOriginal* function, which writes to the output file. It writes the original sentences, colors, phrases to the output file. It is passed the information to be written as well as the output filestream object.

In the branch where all three files were successfully read, call a function *SwapColors,* which returns an int, the number of sentences that were modified. There will be five arguments to this function. This is where you check all of the sentences to see if they contain any of the colors. Any individual color will only be in sentence one time, however, a sentence may have more than one color in it. A sentence may begin with a color, hence the first letter is capitalized. You will have to take that into consideration for your search. If a color is located in the sentence, that color is replaced with a corresponding phrase. If the color is the first word in the sentence, the corresponding phrase must also be capitalized so the sentence is grammatically correct. The total number of replacement actions for each sentence is stored in the NumSwapped vector.

Ask for the output filename, open the output file, and call the function *WriteResults*. It also writes to the output file. In *WriteResults*, write the total number of sentences analyzed, the total that were modified, and the modified sentences. For each sentence state how many colors were swapped out. The output file must contain your name and program #.

In addition to writing the results to the file, display the results to the user in a cout.

When the program has finished executing, display a Good-bye message, either with a cout from main or by calling a function.