**Object-Oriented Programming II Spring 2017**

**CIS 9410**

# Programming Assignment I

For this assignment you are to implement the class **String** whose specification is given below.

Once you have completed the implementation of the **String** class, you are to use this class to implement a multifile application that reads a sequence of up to 100 words from a user, outputs the words, one per line, in ascending order along with an indication for each word that is a palindrome that it is a palindrome. All string variables used in this application should be implemented using the **String** class.

To read the words from the user, the program should repeatedly ask the user if there is a word to be input. Each time the user answers in the affirmative by entering the value “yes” (case should be ignored), the program should then prompt the user to enter a word. (You may assume that these input words do not contain any embedded whitespace characters.) Although the user may enter words in any combination of uppercase and lowercase characters, your program should convert all of the characters to lowercase. To indicate that there are no additional words to be processed, the user should enter the value “no” (case should be ignored). A value other than “yes” and “no” to the prompt for additional data should be ignored, and cause a re-prompting of the question.

After all of the words that are to be processed have been input, they should be sorted in ascending order, and output one per line. For each word that is a palindrome, that word should be followed on the same output line with an indicator, in the form of the string “*(palindrome)*,” that it is a palindrome.

The multifile implementation should include the two files *mystring.h* and *mystring.cpp*, containing the **String** class specification/declaration and the **String** class member and friend function definitions, respectively. It should also include a *source* file containing *main()* and other functions that you use to implement your program.

I will also supply code to test your implementation of the **String** class. For this assignment this code will be in the form of a file that you are to use as the third file – in addition to the files *mystring.h* and *mystring.cpp* – composing your multifile solution.

The **String** class that you are implementing for this assignment is intended to supplant the string class available in *C++*, and as such the *C++* string class may not be used for this assignment.

The non-Sting class member function *bool IsPalindrome(String)* should be used to test a String value to determine if it is a palindrome.

Due Date: March 2, 2017

**String Class Specification**

class String

{

private:

char strval[100]; //string value

int strln; //length of string value

public:

//constructors

String(): strln(0)

{}

String(char s[]); //initialize string value to s

//accessor functions/operators

int Length() const

{return strln;}

int SubString(String s, int startpos=0) const;

String SubString(int startpos, int endpos) const;

void ToCstring(char cs[]); //pass back string value in cs as Cstring

bool operator ==(String s) const;

bool operator >(String s) const;

bool operator <(String s) const;

String operator +(String s) const;

//modifier/mutator functions/operators

void ToLower();

void ToUpper();

void Append(String s);

String operator +=(String s);

bool Replace(int startpos, int endpos, String s);

bool Replace(String targetstr, String replacestr);

bool Insert(String s, int pos=0);

char& operator [](int indx)

{return strval[indx];}

//input/output functions

friend istream& operator >>(istream &strm, String &strng);

friend ostream& operator <<(ostream &strm, String strng);

};