**Object-Oriented Programming II Spring 2017**

**CIS 9410**

# Programming Assignment II

The **String** class, as currently implemented, is limited in its usefulness due to the approach used to store a **String** object’s value. This same approach often leads to the inefficient use of memory. The problem arises from the use of a *statically* allocated array – in our case, of 100 char elements/bytes – to store the **String** object’s value.

For this assignment you are to modify the **String** class by replacing the statically allocated array with one that is dynamically allocated. The size of the dynamically allocated array (i.e., the number of its elements) should, at all times, match the length of a **String** object’s value.

The revised String class should retain the same functionality as the one that it is replacing, with the result that some of the **String** class’s member functions and operators will need to be re-implemented. That is, the new implementation should retain the same functionality as the current one.

To accommodate the switch to the use of pointers in the revised implementation this version of the **String** class should include:

* An explicitly specified copy constructor
* A destructor
* An overload of the assignment operator (=);

In addition, any **String** Class member function that changes the value of a **String** class object or returns a **String** object may also have to be modified to accommodate the use of dynamically allocated arrays.

The revised **String** class should be used to reimplement the application from Programming Assignment 1, as well as the code that I had supplied.

Due Date: March 16, 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);

};