**Program Specifications**

You are to write a program to manipulate strings using pointers. The program will start out by  displaying a menu to ask user what to do as shown below:

                Simple String Manipulation Program

1. Load string
2. Show string
3. Character count
4. Trim string
5. Convert to upper case
6. Quit

Based on user selection the original string will be manipulated and output will be displayed to show the resulted string after manipulation (or character count in case option 3 is chosen). This menu will then be re-displayed for another user selection until user selects option 6 to quit.

Note: when display a string you must use this format: [string content] so that if there is any leading white-space characters in the string we can recognize them. Example: if input string is

“   Welcome to C++ programming world!!!”

Option 2: [   Welcome to C++ programming world!!!]

Option 3: Total character count:   40

Option 4: [Welcome to C++ programming world!!!]

Option 5: [   WELCOME TO C++ PROGRAMMING WORLD!!!]

**Class Design**

You need at least two classes.

class **SmartString**

        Private:

              Data members:  a pointer to a char , namely  str

        Public:

              Constructors:

* Default constructor: initialize the pointer str to NULL
* Non-default constructor: take a pointer to a charater as its only parameter and dynamically allocate memory for str and copy the character array to str. Do not use strlen or strcpy. Use a loop to get the length and to copy instead.

              Destructor: if str is not NULL delete the array of characters

                 Member functions:

* LoadString: ask user to enter a string (may include white space characters). Store it in a character array of maximum 1024 characters (use cin.getline). Dynamically allocate memory for str based on input string's length then copy input string to str.

                        Note: Must use a loop to get the string's length and copying. strlen and strcpy are not

                    allowed

* ShowString: display the string in format [<string content>]
* Size: return the number of characters in str. If str is NULL return 0.
* Trim: remove all leading white-space characters in the original string str
* ToUpper: Converting all letters in the str string to upper case letters
* GetString: return str. This is dangerous and should not be done. It simply illustrates how to return a pointer from a function.

**Important note:** since SmartString class contains a pointer member we should provide a copy

constructor and overload the assignment operator. However you're not required to do it now.

We will discuss about this later on when learning how to overload C++ operators.

class **SmartStringApp**

            Private

                  Data member:  a pointer to a SmartString object, namely ss

             Public

                  Constructors:

* Default constructor: initialize ss to NULL
* Non-default constructor: take a pointer to character as its only parameter. The constructor will dynamically create the SmartString object ss.

                    Destructor: if ss is not NULL then de-allocate ss

                    Member functions:

* Init:  invoke function LoadString from SmartString object ss to load a string from user input
* Start: display the menu shown above and execute SmartString's member functions based on user input

**Implementation Requirements**

* Must use pointers notation when processing the string. No array notation is allowed except for array declarations if needed.
* No string C/C++ library is allowed (except for isspace and toupper)
* Must use the this pointer in at least 4 different places
* Must use member initialization syntax in default constructors
* Main program will simply:
  + declare a pointer to SmartStringApp and dynamically create a SmartStringApp's default object
  + Invoke Start function for the newly created SmartStringApp object
  + free memory for the SmartStringApp object