COM S 213 – Fall 2002

ASSIGNMENT #9: **CoolArray**

DATE GIVEN: **11/7/02**

DATE DUE: **11/14/02—**

PURPOSE:
To apply what we've learned about templates

ASSIGNMENT:
Today we learned that template arguments can be used with classes in one of two ways:

- To allow a data type used in a class to be specified at declaration time
- To allow a "constant" used in a class to be specified at declaration time

For this assignment, you will need to create an array class template similar to the "MyArray" class I used in lecture today. The array class will need to have a constructor, destructor and an overloaded array index operator.

But that's no all! The template class will take two template arguments. One will be the data type to be stored int the array. The second will be the initial number of elements in the array.

Here's the twist. You template array class should use a static declaration of the internal data store (like we saw in today's demos) utilizing the integer template argument. However, instead of simply throwing an exception if there is an out of bounds array access attempt, you should grow your storage *dynamically* to accommodate the integer index passed (so long as it is a positive 32-bit value). So if I have an array of 5 elements an attempt to access the 10th element, your array would automatically grow to 10 spaces "on the fly". Obviously at this point we've switched to dynamic memory allocation.

Your task is to implement this array and test it. If dynamic memory becomes involved during the course of your object's lifetime, you will need to free it in the destructor.

SUGGESTIONS
This one is a little tougher than the last one… email me with questions.