
COM S 213 – Fall 2002

ASSIGNMENT #7: A Better IntArray

DATE GIVEN: 10/24/02

DATE DUE: 10/31/02—BOO!

PURPOSE:

To apply what we've learned about Exceptions to the IntArray class.

ASSIGNMENT:

Assignment #6 had you build an IntArray class. This week we are going to improve upon it! You will need to build off of last week's assignment. If you aren't sure you got Assignment #6 right, I'll post the solution Monday morning.

When doing assignment #6, the overload of the array index operator (`operator[]`) should have been bothering you. Since we are returning a reference in that function we need to fabricate something to return in the case where an error occurs. This is much better handled with an exception. Your assignment is to create a new *Exception class* (see notes from Lecture 15) suitable for throwing when errors occur in your IntArray class. The errors you should be detecting and throwing exceptions for are:

- Array index out of bounds in `IntArray::operator[]()`
- Unable to allocate memory in `IntArray::resize()`

Your exception class should be generic enough to be applied to both error cases, and should store the following information:

- The offending index (when applicable)
- A string message explaining what happened
- An enumerated error code (make up a enum for “negative index”, “index too big” and “no memory available”)

Finally, add a new member function called “print” to IntArray which prints out the array in a fashion similar to this (assuming I have an array of size 4 with the following elements: 5,2,4,3):

Array Dump: Size is: 4

0. 5

1. 2

2. 4

3. 3

SUGGESTIONS

This one is fairly straightforward, email me with questions.