CSC 2111 **Spring 2013**

Name(s): _____

Lab 2: Abstraction and Classes

Reference: *C++ Plus Data Structures*
Chapter 2 - "Date Design and Implementation"
Section 2.3 - "Higher-Level Abstraction and the C++ Class Type"

Objective: To help you understand classes and abstraction, as well as implementing an algorithm and testing.

In the Text: Reference sub-sections "Class Specification" (pgs. 88-89), "Class Implementation" (pgs. 89-91), and "Member Functions with Object Parameters" (pgs. 91-92). (Not needed, but there for you, just in case.)

Description: This exercise demonstrates the difference between classes and structs. In this lab, you are going to fill in the missing parts of an abstract date type called `DateType`, found in *DateType.h* and *DateType.cpp*. Your mission, should you choose to accept it, will be to use the provided test-driver (*DateDr.cpp*) to validate your `DateType` class implementation.

Source File(s): *DateType.h*, *DateType.cpp*, and *DateDr.cpp*

IMPLEMENTATION

Whenever you receive "working" code from someone else, you should first **compile** their code so as to make sure that what you have received is "good".

When you compile all of the .cpp files (using the command "g++ DateType.cpp DateDr.cpp"), does it compile without any warnings or errors? Y/N

Next, you will need to **implement the code** for the following simple methods:

- ✓ `DateType::Initialize`
- ✓ `DateType::GetMonthAsString`

Once you have implemented the above two methods, **compile** your code again.

Does it compile without warnings or error? Y/N