Laboratory 2: Cover Sheet

Name _____ Date _____

Section _____

Place a check mark in the *Assigned* column next to the exercises your instructor has assigned to you. Attach this cover sheet to the front of the packet of materials you submit following the laboratory.

Activities	**Assigned:** Check or list exercise numbers	**Completed**
Implementation Testing	✔	
Programming Exercise 1		
Programming Exercise 2		
Programming Exercise 3		
Analysis Exercise 1		
Analysis Exercise 2		
	Total	

Laboratory 2: Implementation Testing

Name _____ Date _____

Section _____

Check with your instructor whether you are to complete this exercise prior to your lab period or during lab.

Test your implementation of the Text ADT using the program in the file *test2.cpp*. This program supports the following tests.

Lab 2 Online Test Plans	
Test	**Action**
2-1	Test the Date constructors
2-2	Test the Date getters
2-3	Test `Date::isLeapYear`
2-4	Test `Date::daysInMonth`
2-5	Test the << operator overloaded for Date
2-6	Test the BlogEntry constructors
2-7	Test the BlogEntry getters/setters

Test Plan 2-1 (Date constructors)			
Test case	**Parameters**	**Expected result**	**Checked**
First legal date	`1, 1, 1902`		
Current date			
A future date			

Test Plan 2-2 (Date getters)		
Test case	**Expected result**	**Checked**
Earliest legal date (January 1, 1902)	`January 1, 1902`	
Current date		
A future date		

Test Plan 2-3 (`Date::isLeapYear`)			
Test case	**Parameters**	**Expected result**	**Checked**
Normal leap year	`2012`	`True`	
Normal not a leap year	`2009`	`False`	
The multiple of 100 rule			
The multiple of 400 rule			
A future date			

Test Plan 2-4 (`Date::daysInMonth`)			
Test case	**Parameters**	**Expected result**	**Checked**
First month in year	`1, 2010`	`31`	
Last month in year	`12, 2011`	`31`	
A month with 30 days			
A leap-year February			
A non leap-year February			

Test Plan 2-5 (<< operator overloaded for Date)			
Test case	**Date to be tested**	**Expected result**	**Checked**

Test Plan 2-6 (BlogEntry constructors)			
Test case	**Parameters**	**Expected result**	**Checked**
Default constructor	`None`		

Test Plan 2-7 (BlogEntry getters/setters)			
Test case	**Current value or setter parameters**	**Expected result**	**Checked**
getAuthor			
getContents			
getCreateDate			
getModifyDate			
setAuthor			
setContents			

Laboratory 2: Programming Exercise 1

Name _____ Date _____

Section _____

Test Plan 2-8 (printHTML operation)		
Test case items	**Values**	**Checked**
Author value Contents Creation date Modification date		
Write Expected HTML output below		*Correct?*

Laboratory 2: Programming Exercise 2

Name _____ Date _____

Section _____

Test Plan 2-9 (getDayOfWeek)			
Test case	**Date to be tested**	**Expected result**	**Checked**
A Sunday A Wednesday A Saturday			

Laboratory 2: Programming Exercise 3

Name _____ Date _____

Section _____

Test Plan 2-10 (overloaded operators)			
Test case	**The two dates**	**Expected result**	**Checked**
==	Jan 1, 2009 Jan 1, 2009	 True	

Laboratory 2: Analysis Exercise 1

Name _____ Date _____

Section _____

Part A

Design another operation for the BlogEntry ADT and give its specification below. You need not implement the operation, simply describe it.

Function prototype:

Requirements:

Results:

Part B

Describe an application in which you might you use your new operation.

Laboratory 2: Analysis Exercise 2

Name _____ Date _____

Section _____

The BlogEntry class name suggests that it would be used by composition in a Blog class. Design the Blog ADT, specifying data items, structure, and operations. For each operation, indicate the prototype, any requirements, and the results.

Data Items

Structure

Operations

We provide one example operation to indicate the format and level of detail expected.

```
BlogEntry& operator[]( int entryNumber ) throw ( logic_error )
```

Requirements:
`entryNumber` must represent a valid entry.

Results:
Returns a reference to the specified blog entry.