

## 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		
<del>Programming Exercise 2</del>		
<del>Programming Exercise 3</del>		
<del>Analysis Exercise 1</del>		
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 <code>Date::isLeapYear</code>
2-4	Test <code>Date::daysInMonth</code>
2-5	Test the <code>&lt;&lt;</code> 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			

SVisa: modify these entries to match the test cases from your *test2.cpp*

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

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		
<i>Write Expected HTML output below</i>		<i>Correct?</i>

## 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.

