Laboratory 13: 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	✔	
Measurement and Analysis Exercise 1		
Measurement and Analysis Exercise 2		
Measurement and Analysis Exercise 3		
Analysis Exercise 1		
Analysis Exercise 2		
	Total	

Laboratory 13: Implementation Testing

Name ______________________________ Date ____________________

Section ______________________________

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

Question 1: What is the resolution of your implementation—that is, what is the shortest time interval it can accurately measure?

Test Plan 13-1 (Timer ADT operations)			
Test case	Actual time period (in seconds)	Measured time period (in seconds)	Checked

Laboratory 13: Measurement and Analysis Exercise 1

Name ____ Date ______________________

Section ____

In the table below, fill in values of N, $2N$, and $4N$: try 1000, 2000, 4000. If you do not obtain meaningful timing data—especially for `binarySearch`—change the value of N and try again.

Timings Table 13-2 (Search routines execution times)			
Routine	Number of keys in the list (`numKeys`)		
	N =	$2N$ =	$4N$ =
`linearSearch` O(N)			
`binarySearch` O(Log N)			
`STLSearch` O( )			

Please list times in seconds

Question 1: How well do your measured times conform to the order-of-magnitude estimates given for the `linearSearch` and `binarySearch` routines?

Question 2: Using the code in the file *search.cpp* and your measured execution times as a basis, develop an order-of-magnitude estimate of the execution time of the `STLSearch` routine. Briefly explain your reasoning behind this estimate.

Laboratory 13: Measurement and Analysis Exercise 2

Name ________________________________ Date ____________________

Section ________________________________

In the table below, fill in values of N, $2N$, and $4N$: try 1000, 2000, 4000. If you do not obtain meaningful timing data—especially for `quickSort`—change the value of N and try again.

Timings Table 13-3 (Execution times of a set of sorting routines)			
Routine	**Number of keys in the list (`numKeys`)**		
	N =	$2N$ =	$4N$ =
`selectionSort` $O(N^2)$			
`quickSort` $O(N \text{ Log } N)$			
STL `sort` O( )			

Please list times in seconds

Question 1: How well do your measured times conform with the order-of-magnitude estimates given for the `selectionSort` and `quickSort` routines?

Question 2: Using the code in the file *sort.cpp* and your measured execution times as a basis, develop an order-of-magnitude estimate of the execution time of the STL `sort` routine. Briefly explain your reasoning behind this estimate.

Laboratory 13: Measurement and Analysis Exercise 3

Name ______________________________ Date ________________

Section ______________________________

In the table below, fill in values for the various constructor tests. Try an initial value of N=1000. If you do not obtain meaningful timing data, change the value of N and try again.

Timings Table 13-4 (Timing constructor/initialization just before vs. inside loop)		
	Constructor/initialization location	
Your value of *N*: ____________	Outside loop	Inside loop
`int`		
`double`		
`vector`		
`TestVector`		

Please list times in seconds

Question 1: For each data type, how do your measured times for the constructor just before the loop compare to the times for the constructor inside the loop? What might explain any observed differences?

In the table below, fill in values for the various increment tests. Try an initial value of N=1000. If you do not obtain meaningful timing data, change the value of N and try again.

Timings Table 13-5 (Timing pre-/post-increment operators)		
Your value of N: ___________________	pre-increment	post-increment
`int`		
`double`		
`TestVector`		

Please list times in seconds

Question 2: For each data type, how do your measured times for the pre-increment operator compare to the times for the post-increment operator? What might explain any observed differences?

Laboratory 13: Analysis Exercise 1

Name ______________________________ Date ____________________

Section ______________________________

You are given another pair of searching routines. Both routines have order-of-magnitude execution time estimates of O(*N*). When you measure the actual execution times of these routines on a given system using a variety of different data sets, you discover that one routine consistently executes five times faster than the other. How can both routines be O(*N*), yet have different execution times when they are compared using the same system and the same data?

Laboratory 13: Analysis Exercise 2

Name ______________________________ Date ____________________

Section ______________________________

Why might the authors of the STL choose a search implementation that has the big-O performance that you observed?