

Laboratory 6: 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 6: 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 Plan 6-1 (Stack ADT operations)			
Test case	Commands	Expected result	Checked
Series of pushes	+a +b +c +d	a b c d	
Series of pops	- - -	a	
More pushes	+e +f	a e f	
More pops	- -	a	
Empty? Full?	E F	False False	
Empty the stack	-	Empty stack	
Empty? Full?	E F	True False	

Note: The topmost data item is shown in bold.

Laboratory 6: Programming Exercise 1

Name _____ Date _____

Section _____

Test Plan 6-2 (postfix arithmetic expression evaluation)			
Test case	Arithmetic expression	Expected result	Checked
One operator	34+		
Nested operators	34+52/ *		
Uneven nesting	93 * 2+1 -		
All operators at end	4675 -+ *		
Zero dividend	02/		
Single-digit number	7		

Laboratory 6: Programming Exercise 2

Name _____ Date _____

Section _____

Test Plan 6-3 (Eight queens)								
	1	2	3	4	5	6	7	8
1								
2								
3								
4								
5								
6								
7								
8								

Mark the queen locations and visually verify that the printed solution is valid.

Laboratory 6: Programming Exercise 3

Name _____ Date _____

Section _____

Test Plan 6-4 (delimitersOk operation)			
Test case	Commands	Expected result	Checked
Valid expression with parentheses	3 * (a+b)	true	
Valid expression with mixed delimiters	F[3 * (a+b)]	true	
Invalid expression with mixed delimiters	(f[b] - (c+d] / 2;	false	
Empty expression	<i>Empty string<Newline></i>		
Improperly paired brace	a = f [+ 3		

Note: The improperly matched delimiters are shown in **bold**.

Laboratory 6: Analysis Exercise 1

Name _____ Date _____

Section _____

Given the input string “abc”, which permutations of this string can be output by a code fragment consisting of only the statement pairs

```
cin >> ch;    permuteStack.push(ch);
```

and

```
ch = permuteStack.pop();    cout << ch;
```

where `ch` is a character and `permuteStack` is a stack of characters? Note that each of the statement pairs may be repeated several times within the code fragment and that the statement pairs may be in any order. For instance, the code fragment

```
cin >> ch;    permuteStack.push(ch);  
cin >> ch;    permuteStack.push(ch);  
cin >> ch;    permuteStack.push(ch);  
ch = permuteStack.pop();    cout << ch;  
ch = permuteStack.pop();    cout << ch;  
ch = permuteStack.pop();    cout << ch;
```

outputs the string “cba”.

Part A

For each of the permutations listed below, give a code fragment that outputs the permutation or a brief explanation of why the permutation cannot be produced.

<code>"abc"</code>	<code>"acb"</code>
<code>"bac"</code>	<code>"bca"</code>
<code>"cab"</code>	<code>"cba"</code>

Part B

Given the input string “`abcd`”, which permutations beginning with the character ‘`d`’ can be output using the same code fragment combinations (e.g., `cin/push`, `pop/cout`) described previously? Why can only these permutations be produced?

Laboratory 6: Analysis Exercise 2

Name _____ Date _____

Section _____

For each of the stack implementations, identify the performance order of magnitude (big-O value) for the listed operations. Then provide a justification for your big-O value.

Operation	Array-based	Linked
<div>Push</div> <div>Justification</div>	<div>O()</div>	<div>O()</div>
<div>Pop</div> <div>Justification</div>	<div>O()</div>	<div>O()</div>
<div>Clear</div> <div>Justification</div>	<div>O()</div>	<div>O()</div>

