

**Lab 1**

<b>Requirement</b>	<b>Possible Points</b>	<b>Points Earned</b>
Header filled in (name, etc.)	1	
Builds	1	
Uses appropriate variable names/types	1	
Displays proper username	2	
Correctly allows input with multiple dots	1	
Correctly determines site type	2	
Uses 3 functions appropriately	2	
<b>Total</b>	<b>10</b>	

Comments:

**Lab 2**

<b>Requirement</b>	<b>Possible Points</b>	<b>Points Earned</b>
Header filled in (name, etc.)	1	
Uses appropriate variable names/types	1	
At least three functions	1	
Function that reads in text file	1	
-Adjusts to any size playlist	2	
-Playlist title displays first	1	
Function that calculates the time	3	
Function that displays each line appropriately	1	
-Correctly calculates and displays playlist songs	3	
-Correctly number each line	1	
Total time is correctly calculated and displayed	2	
Output is correctly spaced and formatted	3	
<b>Total</b>	<b>20</b>	

Comments:

**Lab 3**

<b>Requirement</b>	<b>Possible Points</b>	<b>Points Earned</b>
Header filled in (name, etc.)	1	
Uses appropriate variable names/types	1	
Input - works with 30 - 50 integers	2	
Displays the data as requested	2	
Sorts the data properly and redisplay	1	
Menu Function	2	
Mean Function correct	2	
Median Function correct	2	
Mode Function correct	3	
Range Function Correct	2	
Standard Deviation Function Correct	2	
Proper usage of vectors as parameters	2	
<b>Total</b>	<b>22</b>	

Comments:

**Lab 4**

<b>Requirement</b>	<b>Possible Points</b>	<b>Points Earned</b>
Header filled in (name, etc.)	1	
Uses appropriate variable names/types	1	
Three functions	3	
Input of data correct	1	
Correct output	3	
Correct calculation of high score	1	
Correct calculation of who won	3	
Proper usage of 2-D Array	2	
<b>Total</b>	<b>15</b>	

Comments:

**Lab 5**

<b>Requirement</b>	<b>Possible Points</b>	<b>Points Earned</b>
Header filled in (name, etc.)	1	
Uses appropriate variable names/types	1	
Three functions	3	
Randomly place ship	2	
Display Board	2	
Game Simulation (input, hit/miss)	3	
Properly end game and state if they won	3	
Proper usage of 2-D Array implemented as vector	2	
<b>Total</b>	<b>17</b>	

Comments:

**Lab 6**

<b>Requirement</b>	<b>Possible Points</b>	<b>Points Earned</b>
Header filled in (name, etc.)	1	
Uses appropriate variable names/types	1	
Proper usage of shell program	2	
Only proper menu input accepted	2	
Only proper values accepted	2	
Correct recursive calculation of factorial	3	
Correct recursive calculation of fibonacci	3	
Correct recursive calculation of triangular	3	
<b>Total</b>	<b>17</b>	

Comments:

**Lab 7**

Requirement	Possible Points	Points Earned
Header filled in (name, etc.)	1	
Three functions	3	
Prompt for filename	2	
Appropriately read in text file and set-up matrix	4	
Display matrix to screen	2	
Ask for starting location	1	
Correct use of recursive AreaFill function	3	
Display of proper new matrix	2	
<b>Total</b>	<b>18</b>	

Comments:

**Lab 8**

Requirement	Possible Points	Points Earned
Header filled in (name, etc.)	1	
Three functions	3	
Proper use of the 2-d array	2	
Recursive Method	3	
Display a proper solution of 8 queens	4	
<b>Total</b>	<b>13</b>	

Comments:

**Lab 9**

Requirement	Possible Points	Points Earned
Header filled in (name, etc.)	1	
Prompt for number of marbles	2	
Loop appropriate number of times	2	
Correct output format	2	
Answered analysis questions	3	
<b>Total</b>	<b>10</b>	

Comments:

**Lab 10**

Requirement	Possible Points	Points Earned
proper setCode function(s) ints and vectors	3	
proper dialCode function(s) ints and vectors	3	
proper isOpen function(s) ints and vectors	2	
Correct declaration all private data members	3	
Correct creation of constructors	3	
Proper testing program showing all the capabilities of your Lock class	6	
<b>Total</b>	<b>20</b>	

Comments:

**Lab 11**

Requirement	Possible Points	Points Earned
Correct default constructor	2	
Correct 2 int parameter constructor	2	
Correct get function	2	
Correct show function	2	
Correct implementation of + operator	4	
Correct implementation of * operator	4	
Correct declaration all private data members	2	
<b>Total</b>	<b>18</b>	

Comments:

**Lab 12**

<b>Requirement</b>	<b>Possible Points</b>	<b>Points Earned</b>
Correct declaration all private data members	3	
Correct default constructor	2	
Correct 2 int parameter constructor	2	
Correct get function	2	
Correct show function	2	
Correct isAfter function	3	
Correct implementation of == operator	3	
Correct implementation of > operator	2	
Correct implementation of < operator	2	
Correct implementation of + operator	5	
Correctly overloading the << operator	3	
Correctly overloading the >> operator	3	
Testing program	8	
<b>Total</b>	<b>40</b>	

Comments:

**Lab 13**

Requirement	Possible Points	Points Earned
An appropriate calculator-testing program	12	
Default Constructor	1	
Constructor BigNumber(int)	2	
Constructor BigNumber(String)	2	
Copy Constructor BigNumber(BigNumber)	2	
De-constructor	1	
Overloaded operator <<	2	
Overloaded operator >>	2	
Overloaded operator * (int num)	3	
Overloaded operator * (BigNumber)	3	
Overloaded operator +	3	
Overloaded operator -	3	
Overloaded operator -=	3	
Overloaded operator +=	3	
Overloaded operator *= (BigNumber)	3	
Overloaded operator *= (int)	3	
Overloaded operator ==	2	
Overloaded operator !=	2	
Overloaded operator >	2	
Overloaded operator <	2	
Overloaded operator >=	2	
Overloaded operator <=	2	
Overloaded operator =	3	
IsNegative function	2	
IsPositive Function	2	
AddSigDigit Function	2	
ChangeDigit	2	
GetDigit	2	
NumDigits	2	
Normalize	2	
Private variables	3	
<b>Total</b>	<b>80</b>	

Comments: