1. What is the difference between a size declarator and a subscript? The size declarator is used in a definition of an array to indicate the number of elements the array will have. A subscript is used to access a specific element in an array.

Size declarator: The number inside the brackets of an array. It indicates the number of elements , or values an array can hold. An array's size declarator must be a constant integer expression with a value greater than zero. It can either be a literal:
int days[6};
or a named constant:
const int NUM_DAYS = 6;
int days[NUM_DAYS];

Subscript: The number inside the brackets of an assignment operator or any statement that works with the contents of an array. A subscript is used as an index to pinpoint a specific element within an array. The first element will always be assigned subscript [0] or sub-zero.
2. Look at the following array definition. int values[10]; How many elements does the array have? What is the subscript of the first element in the array? What is the subscript of the last element in the array? Assuming that an int uses four bytes of memory, how much memory does the array use? - 10 elements
- 0
- 9
- 10 * 4 = 40 bytes.
3. Why should a function that accepts an array as an argument, and processes that array, also accept an argument specifying the array's size? Because, with the array alone the function has no way of determining the number of elements
it has.
4. Consider the following array definition: int values[5] = { 4, 7, 6, 8, 2 }; What does each of the following statements display? cout << values[4] << endl; _____ cout << (values[2] + values[3]) << endl; _____ cout << ++values[1] << endl; _____ 2

14

8
5. How do you define an array without providing a size declarator? By providing an initialization list. The array is sized to hold the number of values in the list.
6. Look at the following array definition. int numbers[5] = { 1, 2, 3 }; What value is stored in numbers[2] ? What value is stored in numbers[4] ? 3

0
7. Assuming that array1 and array2 are both arrays, why is it not possible to assign the contents of array2 to array1 with the following statement? array1 = array2; Because an array name without brackets and a subscript represents the array's beginning
memory address. The statement shown attempts to assign the address of array2 to array1,

which is not permitted. p398-399
8. Assuming that numbers is an array of double s, will the following statement display the contents of the array? cout << numbers << endl; No
Array's must have the data read into or extracted from each element on at a time. This can be done manually through multiple cin or cout statements or by stepping though the array using a loop. pg382
9. Is an array passed to a function by value or by reference? Reference

If the function were written to accept the entire array as an argument...ex: of parameter setup
```
void showValues(int nums[ ], int size)
{
 for (int index = 0; index < size; index++)
 cout << nums[index] << " ";
 cout << endl;
}
```
p408
10. When you pass an array name as an argument to a function, what is actually being passed? the beginning address of the array
pg409
11. How do you establish a parallel relationship between two or more arrays? By using the same subscript value for each array.
p404
12. Look at the following array definition. double sales[8][10]; How many rows does the array have? How many columns does the array have? How many elements does the array have? Write a statement that stores a number in the last column of the last row in the array. 8 rows across
10 colums down
80 Elements
data type "double" = 8 bytes => 80 * 8 = 160 bytes
cin >> col_list[9];
13. When writing a function that accepts a two-dimensional array as an argument, which size declarator must you provide in the parameter for the array? C++ requires the second size declator which is the number of columns to be specified in the function prototype and header because of the way the two-dimensional arrays are stored in memory. (one row follows another: |_|_|_||_|_|_||_|_|_|. p423

note: A two dimensional array (2D arrays) is like several identical arrays put together. It's useful for storing multiple sets of data (rows and columns).
14. What advantages does a vector offer over an array? - You don not have to declare the number of elements that the vector will have.
- If you add a value to a vector that is already full, the vector will automatically increase its size to accommodate the new value. (no out of bounds)
- Vectors can report the number of elements they contain. p430

note: to use a vector you must #include <vector>
syntax for defining a vector: vector<int> numbers;
A size declator is not required since a vector expands in size as you add to it. However, a starting size can be defined: vector<int> numbers(10).
15. The _____ indicates the number of elements, or values, an array can hold. size declarator.

p376
16. The size declarator must be a(n) _____ with a value greater than _____. constant integer expression

zero

p376
17. Each element of an array is accessed and indexed by a number known as a(n) _____. subscript
p377
18. Subscript numbering in C++ always starts at _____. zero
p377
19. The number inside the brackets of an array definition is the _____, but the number inside an array's brackets in an assignment statement, or any other statement that works with the contents of the array, is the _____. size declarator

subscript

p378
20. C++ has no array _____ checking, which means you can inadvertently store data past the end of an array. bounds

p385
21. Starting values for an array may be specified with a(n) _____ list. initialization
p388
22. If an array is partially initialized, the uninitialized elements will be set to _____. zero
p391
23. If the size declarator of an array definition is omitted, C++ counts the number of items in the _____ to determine how large the array should be. initialization list

p391
24. By using the same _____ for multiple arrays, you can build relationships between the data stored in the arrays. subscript

p404

25. You cannot use the _____ operator to copy data from one array to another in a single statement. assignment operator (=)

ex: //wrong newVaules = oldValues;
reason, anytime the name of an array is used without brackets and a subscript, it is seen as the array's beginning memory address. p398
26. Any time the name of an array is used without brackets and a subscript, it is seen as _____. beginning memory address
p398
27. To pass an array to a function, pass the _____ of the array. address or name
p409-410
28. A(n) _____ array is like several arrays of the same type put together.
two-dimensional

p418
29. It's best to think of a two-dimensional array as having _____ and _____. rows colums
p419
30. To define a two-dimensional array, _____ size declarators are required. two
p419
31. When initializing a two-dimensional array, it helps to enclose each row's initialization list in _____. braces
p422
32. When a two-dimensional array is passed to a function the _____ size must be specified. column
p422
V33. The _____ is a collection of programmer-defined data types and algorithms that you may use in your programs. (vectors) Standard Template Library (STL)

p429
V34. The two types of containers defined by the STL are _____ and _____.
sequence
associative
p429
V35. The vector data type is a(n) _____ container. sequence
p429
V36. To define a vector in your program, you must #include the _____ header file.
<vector>
p430
V37. To store a value in a vector that does not have a starting size, or that is already full, use the _____ member function. push_back

p435

V38. To determine the number of elements in a vector , use the _____ member function. size

p437
V39. Use the _____ member function to remove the last element from a vector . pop_back

p438
V40. To completely clear the contents of a vector , use the _____ member function. clear

p439
50. T F An array's size declarator can be either a literal, a named constant, or a variable. F not a variable
p376
51. T F To calculate the amount of memory used by an array, multiply the number of elements by the number of bytes each element uses. T
ex: int mailes[84]
# of elements = 84 * size of each element = 4 bytes = 336 bytes.
p377
52. T F The individual elements of an array are accessed and indexed by unique numbers. T
subscripts
p377
53. T F The first element in an array is accessed by the subscript 1. F
zero
54. T F The subscript of the last element in a single-dimensional array is one less than the total number of elements in the array. T
55. T F The contents of an array element cannot be displayed with cout . F
56. T F Subscript numbers may be stored in variables. T

p380
57. T F You can write programs that use invalid subscripts for an array. T

No bounds checking in C++
p384
58. T F Arrays cannot be initialized when they are defined. A loop or other means must be used. F

p387
59. T F The values in an initialization list are stored in the array in the order they appear in the list. T

p388
60. T F C++ allows you to partially initialize an array. T

p390-391
61. T F If an array is partially initialized, the uninitialized elements will contain "garbage." F

they will be set to zero

(if a local array is completely uninitialized, its elements will contain garbage, just like other local variables)

p391
62. T F If you leave an element uninitialized, you do not have to leave all the ones that follow it uninitialized. F

C++ does not provide a way to skip elements in the initialization list. Ex: //not legal
int nums[12] = {2, 4, , 8 , , 12};

p391
63. T F If you leave out the size declarator of an array definition, you do not have to include an initialization list. F

note: you must provide an initialization list if you leave out an array's size declarator. Otherwise C++ doesn't know how to make an array.
p392
64. T F The uninitialized elements of a string array will automatically be set to the value "0" . The uninitialized elements of a string array will contain empty strings. p392
65. T F You cannot use the assignment operator to copy one array's contents to another in a single statement. T

because the name of an array w/o the brackets and subscript stands for the array's starting memory address.
newValues = oldValues;
8012 = 8024;
p398
66. T F When an array name is used without brackets and a subscript, it is seen as the value of the first element in the array. F

it is seen as the array's starting memory address.
p398
67. T F To pass an array to a function, pass the name of the array.T
p398

68. T F When defining a parameter variable to hold a single-dimensional array argument, you do not have to include the size declarator. T

Because the
69. T F When an array is passed to a function, the function has access to the original array.
T
p411
70. T F A two-dimensional array is like several identical arrays put together. T
p418
71. T F It's best to think of two-dimensional arrays as having rows and columns. T
p419
72. T F The first size declarator (in the declaration of a two-dimensional array) represents the number of columns. The second size definition represents the number of rows. F

Rows
P419
73. T F Two-dimensional arrays may be passed to functions, but the row size must be specified in the definition of the parameter variable. F

Column
p422
74. T F C++ allows you to create arrays with three or more dimensions. T

C++ does not limit the number of dimensions that an array may have.
p425
75. T F A vector is an associative container. F

sequence
p429
76. T F To use a vector , you must include the vector header file. T

#include <vector>
p430
77. T F vectors can report the number of elements they contain. T
p430 and 437
78. T F You can use the [] operator to insert a value into a vector that has no elements. F

You cannot use the [] operator to access a vector element that does not exist. To store a value in a vector that does not have a starting size, or that is already full, use the push_back member function.

key phrase: elements that already exist
P435

79. T F If you add a value to a vector that is already full, the vector will automatically increase its size to accommodate the new value. T

p430