Name: CAELAN KIMBALL **ID: 020022**
Date: 02/06/2020 **Class: Computer Programming 2**
Score: 38 / 42 (90.48%)

Computer Programming Chapter 7 Test

TRUE/FALSE

F 1. True/False: C++ limits the number of array dimensions to two.

Points: 1 / 1

F 2. True/False: In C++ 11, the range-based `for` loop is best used in situations where you need the element subscript for some purpose.

Points: 1 / 1

F 3. True/False: Although two-dimensional arrays are a novel idea, there is no known way to pass one to a function.

Points: 1 / 1

T 4. True/False: Each individual element of an array can be accessed by the array name and an element number, called a subscript.

Points: 1 / 1

T 5. True/False: If an array is partially initialized, the uninitialized elements will be set to zero.

Points: 1 / 1

T 6. True/False: A `vector` object automatically expands in size to accommodate the items stored in it.

Points: 1 / 1

Name: CAELAN KIMBALL **ID: 020022**

MULTIPLE CHOICE

C 7. Unlike regular variables, these can hold multiple values.
   a. constants
   b. named constants
   c. arrays
   d. floating-point variables
   e. None of these

   Points: 1 / 1

B 8. The individual values contained in array are known as _____.
   a. parts
   b. elements
   c. numbers
   d. constants
   e. None of these

   Points: 1 / 1

D 9. Which of the following is a valid C++ array definition?
   a. `int array[0];`
   b. `float $payments[10];`
   c. `void numbers[5];`
   d. `int array[10];`
   e. None of these

   Points: 1 / 1

D 10. The statement:

```
int grades[ ] = { 100, 90, 99, 80};
```

   shows an example of:
   a. default arguments
   b. an illegal array declaration
   c. an illegal array initialization
   d. implicit array sizing
   e. None of these

   Points: 1 / 1

C 11. By using the same _____ you can build relationships between data stored in two or more arrays.
a. array name
b. data
c. subscript
d. arguments
e. None of these

Points: 1 / 1

A 12. The name of an array stores the _____ of the first array element.
a. memory address
b. value
c. element number
d. data type
e. None of these

Points: 1 / 1

B 13. A two-dimensional array is like _____ put together.
a. an array and a function
b. several identical arrays
c. two functions
d. two arrays of different types
e. None of these

Points: 1 / 1

A 14. A two-dimensional array can be viewed as _____ and _____.
a. rows, columns
b. arguments, parameters
c. increments, decrements
d. All of these
e. None of these

Points: 1 / 1

A 15. If you leave out the size declarator in an array definition:
a. you must furnish an initialization list
b. you are not required to initialize the array elements
c. all array elements default to zero values
d. your array will contain no elements

Points: 1 / 1

D 16. Which of the following is a valid C++ array definition?

a. `int scores[0];`
b. `float $payments[10];`
c. `int readings[4.5];`
d. `int scores [10];`
e. None of these

Points: 1 / 1

B 17. An element of a two-dimensional array is referred to by _____ followed by _____.

a. the array name, the column number of element
b. the row subscript of the element, the column subscript of the element
c. a comma, a semicolon
d. the row subscript of element, the array name
e. None of these

Points: 1 / 1

B 18. When writing functions that accept multi-dimensional arrays as arguments, _____ must be explicitly stated in the parameter list.

a. all dimensions
b. all but the first dimension
c. the size declarator of the first dimension
d. all element values
e. None of these

Points: 1 / 1

A 19. An array can store a group of values, but the values must be:

a. the same data type
b. each of a different data type
c. constants
d. integers
e. None of these

Points: 1 / 1

B 20. An array's size declarator must be a _____ with a value greater than _____.

a. number, one
b. number, zero
c. constant integer expression, zero
d. variable, -1
e. None of these

Points: 0 / 1

D 21. Subscript numbering in C++_____.

a. can be set at runtime
b. can begin with a programmer-defined value
c. varies from program to program
d. begins with zero
e. None of these

Points: 1 / 1

C 22. Arrays may be _____ at the time they are _____.

a. resized, executed
b. re-scoped, deleted
c. initialized, declared
d. pre-compiled, typecast
e. None of these

Points: 1 / 1

C 23. Given the following declaration, where is the value 77 stored in the scores array?

```
int scores[] = {83, 62, 77, 97};
```

a. `scores[0]`
b. `scores[1]`
c. `scores[2]`
d. `scores[4]`

Points: 1 / 1

A 24. An array can easily be stepped through by using a _____.

a. `for` loop
b. reference variable
c. named constant
d. null value
e. None of these

Points: 1 / 1

C 25. The range-based `for` loop, in C++ 11, is designed to work with a built-in variable known as the _____.

a. counter variable
b. `i` variable
c. iterator
d. range variable
e. None of these

Points: 0 / 1

C 26. To assign the contents of one array to another, you must use _____.

a. the assignment operator with the array names
b. the equality operator with the array names
c. a loop to assign the elements of one array to the other array
d. Any of these
e. None of these

Points: 1 / 1

C 27. To pass an array as an argument to a function, pass the _____ of the array.

a. contents
b. size, expressed as an integer
c. name
d. value of the first element
e. None of these

Points: 1 / 1

A 28. A two-dimensional array can have elements of _____ data type(s).

a. one
b. two
c. four
d. Any of these
e. None of these

Points: 1 / 1

D 29. A two-dimensional array of characters can contain _____.

a. strings of the same length
b. strings of different lengths
c. uninitialized elements
d. All of these
e. None of these

Points: 1 / 1

A 30. A(n) _____ can be used to specify the starting values of an array.

a. initialization list
b. array name
c. subscript
d. element
e. None of these

Points: 1 / 1

D 31. The _____ is automatically appended to a character array when it is initialized with a string constant.

a. array name
b. number of elements
c. value of the first element
d. null terminator
e. None of these

Points: 1 / 1

A 32. An array of `string` objects that will hold 5 names would be declared using which statement?

a. `string names[5];`
b. `string names(5);`
c. `string names5;`
d. `String[5] names;`
e. None of these will work.

Points: 1 / 1

B 33. It is _____ to pass an argument to a function that contains an individual array element, such as `numbers[3]`.

a. illegal in C++
b. legal in C++
c. not recommended by the ANSI committee
d. not good programming practice
e. None of these

Points: 1 / 1

A 34. How many elements does the following array have?

```
int bugs[1000];
```

a. 1000
b. 999
c. 1001
d. Cannot tell from the code

Points: 1 / 1

A 35. What will the following code display?

```
int numbers[] = {99, 87, 66, 55, 101 };
cout << numbers[3] << endl;
```

a. 55
b. 66
c. 101
d. 87

Points: 1 / 1

B 36. What will the following code do?

```
const int SIZE = 5;
double x[SIZE];
for(int i = 2; i <= SIZE; i++)
{
    x[i] = 0.0;
}
```

a. Each element in the array is initialized to 0.0
b. Each element in the array, except the first, is initialized to 0.0
c. Each element in the array, except the first and the last, is initialized to 0.0
d. An error will occur when the code runs

Points: 0 / 1

C 37. Which statement correctly uses C++ 11 to initialize a vector of `int`s named `n` with the values 10 and 20?

a. `vector n<int>(10, 20);`
b. `vector<int> n = {10, 20};`
c. `vector<int> n { 10, 20 };`
d. `int vector n ({10}, {20});`

Points: 1 / 1

B 38. What does the following statement do?

```
vector<int> v(10);
```

a. It creates a `vector` object and initializes all of its elements to the value 10.
b. It creates a `vector` object with a starting size of 10.
c. It creates a `vector` object and initializes the first element with the value 10.
d. It creates a `vector` object that can store only values of 10 or less.

Points: 1 / 1

C 39. What will the following C++ 11 code display?

```
vector<int> numbers { 3, 5 };

for (int val : numbers)
    cout << val << endl;
```

a. 5
   5
   5

b. 3
   3
   3
   3
   3

c. 3
   5

d. Nothing. This code has an error.

Points: 1 / 1

D 40. This `vector` function is used to insert an item into a `vector`.

a. `insert_item`
b. `add_item`
c. `store`
d. `push_back`

Points: 1 / 1

D 41. This `vector` function removes an item from a `vector`.

a. `remove_item`
b. `delete_item`
c. `erase`
d. `pop_back`

Points: 1 / 1

D 42. This `vector` function returns `true` if the `vector` has no elements.

a. `has_no_elements`
b. `null_size`
c. `empty`
d. `is_empty`

Points: 0 / 1