1. These are operators that add and subtract one from their operands.
   a. plus and minus
   b. **++** and **--**
   c. binary and unary
   d. conditional and relational
   e. binary and ternary

2. This operator increments the value of its operand and then uses the value in context.
   a. prefix increment
   b. postfix increment
   c. prefix decrement
   d. postfix decrement
   e. None of these

3. This is a control structure that causes a statement or group of statements to repeat.
   a. decision statement
   b. **loop**
   c. **cout** object
   d. selection structure
   e. None of these

4. The two important parts of a **while** loop are the expression that is tested for a **true** or **false** value and
   a. a statement or block that is repeated as long as the expression is true
   b. a statement or block that is repeated only if the expression is false
   c. one line of code that is repeated once, if the expression is true
   d. a statement or block that is repeated once, if the expression is true

5. Something within a **while** loop must eventually cause the condition to become **false** or a(n) _____ results.
   a. null value
   b. infinite loop
   c. unexpected exit
   d. compiler error
   e. None of these

6. The **while** loop is a _____ loop.
   a. post-test
   b. pre-test
   c. infinite
   d. limited
   e. None of these

7. If you place a semicolon after the test expression in a **while** loop, it is assumed to be a(n)

a. pre-test loop
b. post-test loop
c. null statement
d. infinite loop
e. None of these

8. The statements in the body of a **`while`** loop may never be executed while the statements in the body of a **`do-while`** loop will be executed

a. at least once
b. at least twice
c. never
d. as many times as the user wishes
e. None of these

9. When the increment operator precedes its operand, as **`++num`**, the expression is in _____ mode.

a. postfix
b. prefix
c. preliminary
d. binary
e. None of these

10. This means to increase a value:

a. decrement
b. increment
c. modulus
d. parse
e. None of these

11. A collection of statements that performs a specific task is a(n)

a. infinite loop
b. variable
c. constant
d. function
e. decision

12. A function _____ contains the statements that make up the function.

a. prototype
b. definition
c. call
d. expression
e. parameter list

13. A function can have no parameters, one parameter, or many parameters and can return _____ value(s).

    a. zero to many
    b. no
    c. only one
    d. a maximum of ten
    e. None of these

14. A function is executed when it is

    a. defined
    b. prototyped
    c. declared
    d. called
    e. None of these

15. Functions are ideal for menu-driven programs. When the user selects a menu item, the program can _____ the appropriate function.

    a. call
    b. append
    c. define
    d. declare
    e. None of these

16. This type of variable is defined inside a function and is not accessible outside the function.

    a. global
    b. reference
    c. local
    d. counter
    e. None of these

17. The value in this type of variable persists between function calls.

    a. global
    b. internal
    c. static
    d. dynamic
    e. None of these

18. These types of arguments are passed to parameters automatically if no argument is provided in the function call.

    a. local
    b. default
    c. global
    d. reference
    e. None of these

19. When used as parameters, these types of variables allow a function to access the parameter's original argument:

    a. reference
    b. floating-point
    c. counter
    d. undeclared
    e. None of these

20. _____ functions may have the same name as long as their parameter lists are different.

    a. Only two
    b. Two or more
    c. No
    d. Un-prototyped
    e. None of these

21. Unlike regular variables, _____ can hold multiple values.

    a. constants
    b. named constants
    c. arrays
    d. floats
    e. None of these

22. The individual values contained in an array are known as

    a. parts
    b. items
    c. constants
    d. elements
    e. None of these

23. To access an array element, use the array name and the element's _____.

    a. data type
    b. subscript
    c. value
    d. name
    e. None of these

24. By using the same _____ you can build relationships between data stored in two or more arrays.

    a. array name
    b. data types
    c. subscript
    d. arguments
    e. None of these

25. The name of an array stores the _____ of the first array element.

    a. value
    b. memory address
    c. element number
    d. data type
    e. None of these

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

    a. `int nums[0];`
    b. `float $payments[10];`
    c. `void numbers[5];`
    d. `int sizes[10];`
    e. None of these

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

    a. `int array[0];`
    b. `float $payments[10.23];`
    c. `int numbers[5.6];`
    d. `int scores[25];`
    e. None of these

28. 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 array elements
    c. all array elements default to zero values
    d. your array will contain no elements
    e. None of these

29. An array can store a group of values, but the values must be

    a. the same data type
    b. integers and floating-point numbers
    c. integers
    d. constants
    e. None of these

30. 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

31. A _____ algorithm is a method of locating a specific item of information in a larger collection of data.

    a. sort
    b. search
    c. standard
    d. linear
    e. None of these

32. The advantage of a linear search is its

    a. complexity
    b. efficiency
    c. simplicity
    d. speed
    e. None of these

33. A _____ search is more efficient than a _____ search.

    a. character, string
    b. integer, double
    c. binary, linear
    d. linear, binary
    e. None of these

34. A binary search begins with the _____ element of an array.

    a. first
    b. last
    c. largest
    d. middle
    e. None of these

35. The _____ sort usually performs fewer exchanges than the _____ sort.

    a. bubble, selection
    b. binary, linear
    c. selection, bubble
    d. ANSI, ASCII
    e. None of these

36. Array elements must _____ before a binary search can be performed.

    a. summed
    b. set to zero
    c. positive integers
    d. sorted
    e. None of these

37. Using a linear search to find a value that is stored in the last element of an array that contains 20,000 elements, _____ elements must be compared.

    a. 20,000
    b. only the first two
    c. only half
    d. 2,000
    e. None of these

38. A(n) _____ search uses a loop to sequentially step through an array.

    a. binary
    b. unary
    c. linear
    d. relative
    e. None of these

39. Data that is to be sorted in ascending order is ordered

    a. from lowest value to highest value
    b. from highest value to lowest value
    c. with a binary search algorithm
    d. by identifying the middle value and going up and down from there
    e. None of these

40. Regardless of the algorithm being used, a search through an array is always performed

    a. from lowest to highest element
    b. from highest to lowest element
    c. beginning with the middle element
    d. using a binary search algorithm
    e. None of these

41. The _____, also known as the address operator, returns the memory address of a variable.

    a. asterisk (*)
    b. ampersand (&)
    c. percent sign (%)
    d. exclamation point (!)
    e. None of these

42. With pointer variables you can _____ manipulate data stored in other variables.

    a. never
    b. seldom
    c. indirectly
    d. All of these
    e. None of these

43. When you work with a dereferenced pointer, you are actually working with

    a. a variable whose memory has been allocated
    b. a copy of the value pointed to by the pointer variable
    c. the actual value of the variable whose address is stored in the pointer variable
    d. None of these

44. A function may return a pointer but the programmer must ensure that the pointer

    a. still points to a valid object after the function ends
    b. has not been assigned an address
    c. was received as a parameter by the function
    d. has not previously been returned by another function
    e. None of these

45. What does the following statement do?

```
double *num2;
```

    a. Declares a **double** variable named **num2**
    b. Declares and initializes a pointer variable named **num2**
    c. Initializes a pointer variable named **num2**
    d. Declares a pointer variable named **num2**
    e. None of these

46. The _____ keyword represents the address **0.**

    a. **nullptr**
    b. **NULL**
    c. **weak_ptr**
    d. **shared_ptr**
    e. None of these

47. When the less than operator (**<**) is used between two pointer values, the expression is testing whether

    a. the value pointed to by the first is less than the value pointed to by the second
    b. the value pointed to by the first is greater than the value pointed to by the second
    c. the address of the first variable comes before the address of the second variable in the computer's memory
    d. the first variable was declared before the second variable
    e. None of these

48. Use the **delete** operator only on pointers that were

    a. never used
    b. not correctly initialized
    c. created with the **new** operator
    d. dereferenced inappropriately
    e. None of these

49. Which of the following is true about this statement:

```
sum += *array++;
```

a. This statement is illegal in C++.
b. This statement will cause a compiler error.
c. This statement assigns the dereferenced pointer's value, then increments the pointer's address.
d. This statement increments the dereferenced pointer's value by one, then assign that value.
e. None of these

50. In the following statement, what does **int** mean?

```
int *ptr = nullptr;
```

a. The variable named ***ptr** will store an integer value.
b. The variable named ***ptr** will store an asterisk and an integer value
c. **ptr** is a pointer variable and will store the address of an integer variable.
d. The variable named ***ptr** will store the value in **nullptr**.
e. None of these