1. What is the role of a prototype program in problem solving?

A prototype program is a part of the Specification phase of Software Problem Solvin. It's employed to illustrate how the key problem or problems will be solved in a program, and sometimes serves as a base program to expand upon.

2. What stages in the software life cycle are influenced by the testing stage?

Directly: Refining, coding. Because Refining is right before the Testing Phase and Coding is right after the Testing Phase. Indirectly: Production, Maintenance. Because Refining occurs before these last two stages in the Software Life Cycle.

3. What are the main advantages associated with object-oriented programming?

Coding and Debugging programs are usually easier, as either the compiler will specify the object that is working incorrectly, or the function an object was assigned to will be easier to identify. The code itself also looks more organized and is easier to read and will help to avoid redundant coding.
Post-programming maintenance is also easier. Also, modules can be reused several times in other programs without too much hassle. Abstraction is the art of breaking down one big problem into smaller, simpler problems and solving them. Many of the smaller problems are shared between unique complex problems, and creating modules to these solve these smaller repeated problems can save time when you encounter them again.

4. Where do C++ programs begin to execute?

In the Main function..?

5. What is a variable?

A variable is a value that is subject to change in a computer's memory that can be used by programs. Programs can change the value of the variable and recall it later or act on it directly.

6. Where are variables declared in a C++ program?

In the main function, usually at the top of code. They can be declared almost anywhere, but must be declared before the code can use or act upon them.

7. What is the main difference between a while and a do...while statement?

A do...while statement's body always executes at least once. Where as a while statement will not execute at all unless the condition for the while is true.

8. What is typically included in a class definition?

the data and methods

9. What is the difference between a data member and a local variable inside a member function?

A data member exists permanently in that program's memory as long as it's run. A local variable inside a member function is temporal and only used to operate in the function and exists as long as the function is run.

10. What is the difference between a constructor and a function?

A constructor sets up the default values of variables when an object is instantiated, whereas a function allows interaction with that object.

11. When does C++ create a default constructor?

When non are provided

12. How many constructors can be created for a class?

infinite

13. What is the difference between a function prototype and a function definition?

Function definitions are just that, the definition. The prototype is what the compiler uses to check that calls to function are correct.

14. What is the role of a header-file?

the header file has access to the C++ Standard Library and makes certain classes like class string work.

15. What does a function signature include?

The portion of the function prototyp tha has the function name and the arguments but NOT the return type.

16. What is the scope of global variables?

They can be accessed by any part of the program. It can be referenced by any function that follows the declarations or definitions in the source file.

17. How does the compiler handle inline functions?

Placing the qualifier inline before a function's return type in the function definition "advises" the compiler to generate a copy of the function's code in place to avoid a function call.

18. What is the main advantage associated with function arguments that are passed by reference?

Passing by reference can eliminate the pass-by-value overhead of copying large amounts of data.

19. How are overloaded functions differentiated by the compiler?

The compiler selects the proper functions to execute based on number, types and order of arguments in the function call.

20. How are overloaded functions differentiated by the compiler?

by their signature (return type, argument list)

21. When defining a recursive function, what are possible causes for infinite recursion?

No easily reached base case and no base case at all

22. What are the similarities between iteration and recursion?

both are based on a control statement, involve repetition and involve a termination test. both gradually approach termination, both can occur infinitely

23. What are the two different ways of specifying the length of an array?

explicitly, by declaring it in brackets (i.e. int array[50];) and implicitly, by initializing several values (i.e. int array[] = {1, 2, 3};)

24. What is the main difference between strings declared using the type string versus strings declared using an array of characters?

an array is passed by reference therefore if an array of characters is changed, the memory is changed not just the variable

25. How are arrays passed to functions?

arrays are passed by reference, passing the starting address of array. programmer specifies the array name without brackets and passes the array size as another argument so the function can process the specific number of elements in the array

26. What is the difference between an array declared as static, and one that is not?

When we declare it static the array is not destroyed after the function is over. When an array is not declared as static, the array is created and initialized every time a function calls it.

27. How many dimensions need to be specified when passing a multi-dimensional array as an argument to a function?

all subsequent dimensions after the first dimension. first is not needed to be specified.

28. In one sentence, what is the main idea implemented by insertion sort?

a comparison sort in which the sorted array is built one entry at a time

29. In one sentence, what is the main idea implemented by selection sort?

Selection sort is an algorithm that selects the larges item the array and puts it in its place; then select the next largest until the array is sorted

30. What is the number of operations for insertion sort under a best-case scenario, and what is the best-case scenario?

N number of times. Where N equals the number of elements in the list. Best-case scenario is that the elements in the list are already sorted.

31. What is the base case for a recursive implementation of merge sort?

one element in an array

32. What is a pointer?

A variable that contains the memory address of another variable that contains a specific value.

33. What does the address (&) operator return?

returns a pointer

34. How can an array be addressed in pointer/offset notation?

Element b[n] can be accessed by *(bPtr + n).

35. What does the sizeof operator return?

it returns the size of the argument passed in

36. What are the different ways to pass a pointer to a function?

By argument and reference.

37. What is a function pointer?

a function pointer is a pointer to a function. It can be used in substitution for calling a function.

38. What is a linked list?

A data type that contains a pointer to at least the next element in a list.

39. What is the main advantage of linked lists over arrays?

Linked lists can grow and shrink as you need them to. Arrays have a fixed size.

40. What is the main advantage of arrays over linked lists?

Array based implementation requires less memory than lined lists based implementation

41. How are linked lists passed as arguments to a function?

linked lists are passed by using pointers

42. What is the main advantage of a doubly-linked list over a basic linked list?

they can traverse in both directions

43. What is the main disadvantage of a doubly-linked list over a basic linked list?

Insertion and deletion with a doubly linked list are more involved than with a singly linked list leaving more room for error and complex algorithms.

44. What is a stack?

A data structure that stores data using LIFO.

45. What are the two main functions defined by a stack?

push and pop.

46. How can you implement a stack with an array?

Make an array, make the bottom at spot 0, make the last spot in the array the top, and increment top when you ad

47. How can you implement a stack with a list?

Use head as the top and push and pop nodes from the head.

48. How are infix expressions evaluated by computers?

i have an hard time explaining this so i'll show how infix is evaluated instead. Start with an infix expression, like, (((5+2)*5)+(400/(2+3))), and push items until you get a ")" and once that happens, perform the operations until you reach an "("... with that complete, you will now have ((7*5)+(400/(2+3))) as now the expression that will be evaluated... perform last step again... (35+(400/(2+3))) is now the stack.... repeat agian... (35+(400/5)) is now the stack after that.... repeat... (35+80) is now the stack, repeat again... 115 is now the stack, and is returned.

49. What operations would you need to perform to find a given element on a stack?

could traverse through the list/array to find the element.

50. What is a queue?

a finite ordered list with zero or more elements, based upon First In First Out

51. What are the two main functions defined by a queue?

Enqueue and dequeue, which add and remove elements from the queue

52. How can you implement a queue with an array?

remove the element then shift the elements one space back

53. How can you implement a queue with a list?

you would have your list and add a back pointer so that your program knows where the end is

54. What is the stack operation corresponding to the enqueue operation in queues?

push

55. What is the height of a tree?

Height = the number of times you must progress down the levels of a tree to reach a leaf of the greatest distance from the root.

56. What is a leaf?

the node with degree 0

57. What is a binary tree?

Starting at the root node, it branches off into one or two subsets that are binary subtrees of the root. Each node has at most two children, the left child and the right child.

58. What is a binary search tree?

a binary tree with a special organization of data. where the left child of the element is less than it, and the right child is larger than it.

59. How many comparisons does it take to find an element in a binary search tree?

This is on the runtime order of log(n) where n is the number of nodes in the binary tree.

60. What are the elements typically included in a class definition?

the functions and variables used when the object is defined for the class.

61. What are the access-specifiers that can be used in a C++ class definition?

Public or Private.

62. How are objects initialized when they are created?

Objects are initialized by a value given. When an object is defined for a certain task.

63. What is a function signature?

A function signature is the return type of a function, its name, and the number and type of its parameters.

64. What is a recursive function?

A function that calls itself to perform an opperation such as a factorial recursive function.

65. What is the alternative way to solve a problem that could be solved through recursive functions?

Iteration, performers a series of checks and loops.

66. What is the alternative way to solve a problem that could be solved through recursive functions?

the alt, way is to use a functions parameter in a call "to" function.

67. What is the difference between an array that is declared as static and one that is not?

An array that is declared static is allocated when the program begins to run, and is not freed until the program exits, but has limited scope. A non-static array is allocated and freed every time that it comes in and out of scope.

68. What is the main difference between a string of characters that is read into a variable of type string versus a variable of type char[]?

a string of characters will have a null character or value at the end of the string

69. Briefly describe the divide-and-conquer paradigm.

You take a class and break down everything it is supposed to do into easily managed functions.

70. Briefly describe in one sentence how does merge sort work?

divides an array in half and sorts each half by calling its self, thus dividing each half again and again and sorting it until the array is sorted. then putting the elements back in the original array sorted.

71. What is a pointer?

a variable that holds the address of another variable and can access the contents of the variable.

72. What is the experimental approach for measuring the running time of an algorithm?

the experimental approach would be to run through the algorithm and see how long it takes

73. Briefly, how does selection sort work?

it searches through the entire data set for the best element to go in the current position, then it moves up a position and searches through the remainder of the set for the best element for the new position.

74. What is the advantage of linked lists over arrays?

linked list many be dynamically grown. It has not limit

75. What are the main operations associated with a stack?

push and pop

76. What is the Euler tour traversal of a tree?

The Euler tour traversal of a tree begins at the root and visits each node in the tree from the left, then bottom, then right sides.

77. How do you delete a node from a binary search tree?

first attaching the elements from the node to be deleting to alternate nodes and then deleting that node.
delete node;

78. How many steps does it take to search a node in a binary search tree?

It is the same as the height of the tree.