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

To address major issues in the creation of the program. There is no way to account for all possible bugs in the program, but it is possible to prove the program is tangible.

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

Refining, Production, Maintenance

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

Reusable components, Extensibility, Maintainability, it reduces large problems into smaller more manageable problems.

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

main method

5. What is a variable?

It's a sybol or name for a value/number. Example: 'a_used_number' can stand for any given number, and the programmer can refer to that number by using the variable name.

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

They can be declared globally just before the main method (but also outside of it)... or variables can be subject to only the method they're scoped within, but would still be declared at the beginning of that method (but inside of it).

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

A while statement will test the condition of the while loop FIRST - there is a chance the loop will never run. A do...while loop will ALWAYS run once, and then the while test will determine if it will run again.

8. What is typically included in a class definition?

data members and member functions.

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

A data member is part of a C++ class or structure that stores data of a certain type... A local variable inside a member function is only available inside that function, and can be used to derive a desired outcome.

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

Constructors cannot return values, so they cannot specify a return type. Normally, constructors are declared public.

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

When no parameters are set is when a default constructor is used.

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

There is no limit to the number of constructors in a class because like functions, constructors can be overloaded.

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

The FUNCTION PROTOTYPE is where the programmer declares that he/she is using a function other than main( ). This is like declaring a variable, the programmer knows that he/she will be using in the future, but has yet to say where they are going to use it, or how. This is answers the question who, it gives the function a name and character. The function prototype, by common practice is placed at the beginning of the program after the #includes and before main( ). The FUNCTION DEFINITION is the guts of the function. This is where the programmer decides what the function is going to do and tells it how to do it. It takes whatever information it is given and performs the operations. It works sort of like the brain, the brain takes in input and based upon that input performs in some way producing an output. The function defintion is placed outside of main( ) and any other functions. A function is its own entity and should be thought of as such.

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

the role of a header file list all the functions a class can do while hiding the inner workings of its functions

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

contains all the standard C++ library's i.e. string

16. What does a function signature include?

name of function and types of arguments

17. What is the scope of global variables?

the whole code

18. How does the compiler handle inline functions?

The complier includes copies of inline functions instead of making function calls, but usually only with very small functions.

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

large data items can be passed without copying the entire data point, reducing execution time and the amout of memory space needed

20. How are overloaded functions differentiated by the compiler?

They have the same name, just different parameters.

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

There is no base case. The recursion step doesn't reduce the problem during each recursive call.

22. What are the similarities between iteration and recursion?

both based on control statement, involve repetition, and involve a termination test within loops

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

manually inside the brackets or automatically via an initializer list

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

Stings can only be modified by functions of the String class, an array of characters can be modified by the user

25. How are arrays passed to functions?

the same way but with [] at the declaration and the actual function.

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

All elements are initialized to zero if not explicitly initialized for a static array, while a non-static array is not initialized to zero.

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

Size of subsequent dimensions are required. Compiler must know how many elements to skip to move to the second element in the first dimension

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

To sort the elements in an array by removing an element from the input data and inserting it at the correct position.

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

Selection sort finds the minimum value in the list, swaps it with the first value in the list, and repeats throughout the list.

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

n-1

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

O(nlog(n))

32. What is a pointer?

A pointer is a variable that contains the memory address of a variable that has a value.

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

the reference to a variable

34. What does the star (*) operator return?

Returns synonym for the object its operand points to

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

a[i] = *(a+i)

36. What does the sizeof operator return?

it yields the size in bytes of the operand, which can be an expression or the parenthesized name of a type

37. What is a function pointer?

Function Pointers are pointers, i.e. variables, which point to the address of a function.

38. What is a linked list?

a list of nodes where each node contains a data member and also a pointer to the next node.

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

Linked list is able to grow in size as needed. Does not require the shifting of items during insertions and deletions

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

Take up less memory and are contiguously stored

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

as a pointer node.

42. What is the difference between a circular linked list and a basic linked list?

No node in a circular linked list contains NULL.

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

Double Linked list are often easier to manipulate, because they allow sequential access to the list in both direction

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

using more pointers to enable the backward movement takes up more memory, having more pointers make it slightly more work to add or remove a node from the list

45. What is a stack?

Stacks are a type of container adaptors, specifically designed to operate in a LIFO context (last-in first-out), where elements are inserted and extracted only from the end of the container.

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

By implementing an array of a pre-defined size with the bottom stack element stored at element 0 and the last element stored in the array is the top. You increment top when an element is pushed and you decrement it when an item is popped.

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

Using nodes to keep track of the head of the stack. Then using Push( ) and Pop ( ) to create the stack as needed.

48. How are infix expressions evaluated by computers?

computers usually convert infix expressions to post fix expression and evaluate them using a stack.

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

Pop an element from one stack, check to see if it is the desired element, if not push it onto another stack. When finished, pop the items from the second stack and push them back onto the first stack (this will ensure the order of the elements is maintained).

50. What is a queue?

is a particular kind of collection in which the entities in the collection are kept in order and the principal (or only) operations on the collection are the addition of entities to the rear terminal position and removal of entities from the front terminal position. First in First Out (FIFO Method).

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

Enqueue and Dequeue

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

you can take the array and put it in a queue stack or you can just traverse the list and keep track of top and bottom

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

Use a linked list with 2 pointers, one to the front and one to the back. As long as back != front, the queue is not empty.

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

the Push method in the stack operations.

55. What is a tree?

a finite collection of nodes, where it starts with an element called the root, which has children, and its children have children until you get to the leaves which are the last elements and have to children

56. What is a leaf?

A leaf is a child of a parent node that has no children nodes of its own.

57. What is a binary tree?

a tree where each node has at most 2 children.

58. What is a binary search tree?

A binary search tree is a tree that also has the condition that each node may have at maximum 2 children and where the input data is compared to the data in the tree starting with the root. If the value is smaller than the root it traverses left, if it is larger it traverses right until it becomes a leaf.

59. What is the inorder traversal of a binary tree?

void inorder(ptNode ptr) { if (ptr) { inorder (ptr->left); cout<<ptr->data; inorder(ptr->right); } }

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

log n comparisons

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

a constructor, private and public variables, and function prototypes

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

data member or data functions are access specifiers

63. How are objects initialized when they are created?

Objects are assigned a space in memory when they are created. the name given to the object points to the memory address where the objects data resides.

64. What is a function signature?

the name of the function and the arguments passed to that function

65. What is a recursive function?

A function that divides and conquers a problem. There is a base case that will eventually be reached. The function calls itself over and over until the base case is satisfied.

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

using an iterative solution instead

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

a static array is one that keeps its values and is not initialized every time a class function is called, it's data is not lost at function end.

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 char[] adds a null value to the end of the string.

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

it is the taking of a larger problem and splitting it into simpler smaller problems.

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

It takes an array, splits itself in half, sorts the two halves (either by recursion or iteration) and compares them together giving a third array a full sorted list with both halves rejoined.

71. What is a pointer?

a form of storing data that keeps the location in memory of an data type or object.

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

For every call into memory, add/ multiply it into an equation of type O(n)

73. Briefly, how does selection sort work?

by selecting the first object and comparing it with the next objects to find one that is smaller, if it does it switches the two. then it comes back the other way and sees if the objects previous to the last one is greater, if it does it switches them. it does this while decreasing the range it looks at until it finishes in the middle.

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

Linkded list does not have a fixed size. Arrays have a fixed size.

75. What is a queue?

It is a "first in first out" data structure. It can be implemented with either a linked list or an array.

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

push, which adds an item to the top of the stack, and pop, which takes the first item off the top to the stack

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

The Euler tour visits every node in a tree, starting with the root and working around the tree counterclockwise. in-order, post-order, and pre-order traversals can be seen as variations of the Euler tour traversal.

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

not answered