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

High risk problems are address in the prototype program to make sure that the program is feasible. A prototype may also be used to show a company that the software can be possibly programmed.

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

Refining and possibly the design if the testing phase reveals problems in the design. Production can be affected if the program is unworkable in its current form which will lead to a later production time than originally estimated. Also affects coding because after testing you may need to rewrite the code for the program to remove errors.

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

Modularability, the ability to reuse parts of the program later in another program sometimes with completely different goals for the program. Also it makes it easier to debug code by dividing up the code into classes that each do a specific job and when the program fails at one job you only have one class to debug. Good for security purposes because it allows you to let someone use a program which sorts lists without having to give them access to the source code. ALso allows you to use inheritance and polymorphism.

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

At function ‘int main ()’

5. What is a variable?

Variable can be a integer or a string in a program.

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

They can be declared right before they are used, but it is a good practice to declare them at the beginning of the program and label what they are food.

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

"The “do” statement first evaluates the condition and then executes the lines of code in the statement 0 or more times. The “do while” statement executes the lines of code and then it evaluates the condition. "

8. What is typically included in a class definition?

For us it is usually a set, get, and displayMessge or other messages that tell you what properties a class has but does not reveal its implementations. You could have anything you want in your own class definition.

9. What is typically included in a class definition?

A class is an expanded concept of a data structure, it holds both the data and the functions being executed

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

A data member is accessible anywhere in the class, while i local variable in a member function is only accessible in that particular function, and is freed after the function exits.

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

constructors initialize the data that is then used in the calculations of a function

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

When one is not explicitly stated.

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

you can overload constructors as you need

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

A functgion prototype is a declaration of a function , while function definition specifies what a function does

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

header files contain code which can be used in multiple files

16. What does a function signature include?

the name of the function and the types of its arguments

17. What is the scope of global variables?

Global variables have a file scope

18. How does the compiler handle inline functions?

It generates a copy of the function's code by by inserting it in the program.

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

Gives access to the caller data directly, also it allows for modifying the data.

20. How are overloaded functions differentiated by the compiler?

Overloaded functions are differentiated by their parameters.

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

Not having the proper case to leave the recursion

22. What are the similarities between iteration and recursion?

Both are based on a control statement. Both involve repetition. Both 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 int array[10]; or by variable const int size = 10; int array[size];

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

string is a class in the standard library and has method that modify it. Whereas the char array is on able to be modifie by the user.

25. How are arrays passed to functions?

Usually the same way you pass anything else... you do not have to specify the length of a char array, just pass it to the function by sending the name of the array. Example: char str[20]; isPalindrome(str); // this sends the char array to the function isPalindrome. Otherwise, would need to send the length of the array as well.

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

a static array will only be initilized once, a non static array will be re-initilized once the program reaches the initilization line again.

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

The first dimension is not required however the subsequent dimension(s) are needed for the compiler.

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

It starts with the second element and checks it to see if it is less than the element(s) to the left of it and if it is it inserts it into its corrected position.

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

Find the lowest value and place it at the front, then find the next lowest and place it in front of the last value, etc.

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

It will do (n) operations where (n) is the number of elements in the array. Best case is when all the elements are already sorted.

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

O(nlog(n))

32. What is a pointer?

A variable that stores the address of another variable

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

the address of the variable in memory

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

The * operator returns the value of the object's memory address it precedes.

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

by using the *(pointer + element-index)

36. What does the sizeof operator return?

returns the size in bytes of the specified data

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

nonconstant pointer to nonconstant data nonconstant pointer to constant data constant pointer to nonconstant data constant pointer to constant data

38. What is a function pointer?

When dereferenced, a function pointer invokes a function, passing it zero or more arguments just like a normal function. In programming languages like C, function pointers can be used to simplify code, such as replacing large switch statements.

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

Array size is fixed,but Linked is not fixed.

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

On the other hand, arrays allow random access, while linked lists allow only sequential access to elements. Singly-linked lists, in fact, can only be traversed in one direction.

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

You have to pass the head pointer to a function since it has access to the entire list.

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

in a circular linked list the the last item points to the first item.

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

ability to backtrack through a list.

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

each node requires an extra pointer, taking up more storage. insertions and deletions require more pointer operations.

45. What is a stack?

A stack is a data structure that allows data members to be added or removed one at a time, and in order. They are a "Last in First Out" structure.

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

pop - to remove item push - to insert an element

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

using arrays, you are pushing one int at a time into the stack until you get to the end of your array

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

Using a linked list is one way to implement a stack so that it can handle essentially any number of elements. It is usually convenient to put a data structure in its own module, thus, you'll want to create files stack.h and a stack.c

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

You would have to keep using peek or getTop to see what the top element is then popping the top element until peek or getTop provides the element you are looking for.

50. What is a queue?

A queue stores a set of elements in a particular order. Its principle of operation is FIFO(first in first out), which means the first element inserted is the first one to be removed.

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

Enqueue and Dequeue

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

Utilizing indexes for the front and the back. Front is 0 and as you add items the back index is incremented.

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

Check if list is empty and add elements to the list.

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

push

55. What is a tree?

A tree is a branched data structure with nodes, and leaves... a special node called root has no parent node, but however like all other nodes can and does have 2 nodes that it links to, namely one on the right, and one on the left.

56. What is the height of a tree?

The height of a tree is the depth of its subtrees, that is, how many layers deep it is.

57. What is a leaf?

The elements at the very bottom of an inverted tree (that is, those that have no elements below them) are called leaves.

58. What is a binary search tree?

A tree which is split based on values. This makes it very easy to search. One can compare the desired value to the root, and if the root is greater than, we search the left side of the tree, if it is less than, we search the right side... and do the same thing recursively.

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

postorder

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

O(n)

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

data members and function definitions

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

private, public

63. How are objects initialized when they are created?

They are initialized with a constructor.

64. What is a function signature?

the name of the function and what it accepts/ returns

65. What is a recursive function?

A recursive function is a function that breaks down a large problem into two or more smaller problems and calls itself to solve those smaller problems until a base case or terminating point is reached. At the base case, a definite answer is called and the function no longer recurses, but unwinds itself back through function calls to determine the answer to the original large problem.

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

Many recursive solutions may also be solved with looping control statements, such as while, for, do-while, etc.

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

An array they is declared as static can be accessed outside of the method it was declared without passing it as an argument to other functions in the same class.

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[]?

character arrays can call on individual letters were a string array can only call words.

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

The divide-and-conquer paradigm divides the problem into two pieces and works on each piece separately, allowing for a faster approach to 2 separate smaller problems.

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

Merge sort breaks the array in half, and continues to do so until it has 2 elements to compare and sorts them, after doing so it "merges" back as it keeps on sorting the algorithm as it does so.

71. What is a pointer?

A data type that points to a specific memory address.

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

keep a value of how many operations it takes and add to this value each time a function is called.

73. Briefly, how does selection sort work?

selection sort finds the lowest element in the data set and places it behind the pivot point.

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

Linked lists may be dynamically grown.

75. What is a queue?

A queue is a First in First out data structure much like a line for a movie theatre. The first object in line is the first object to be handled or enacted upon.

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

pop, push

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

it starts node on the left of the root and then proceeds to visits each node in a left to right order, visits the root, and then proceeds to repeat the previous step on the right side of the tree.

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

Create a temp Node Set temp's values to the Node after head or NULL in the case of only head Node in the list. Set head equal to temp. Delete temp