

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

Simulating the behavior of only a portion of the desired software product.

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

Refining and Coding

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

Re-usability and ease of maintenance

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

main

5. What is a variable?

a block of memory that holds a specific type of data

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

Globally for an entire program, and locally for individual functions (including FOR statements)

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

while statements will evaluate the statement BEFORE executing a cycle of the loop, including the first cycle, meaning that a while loop will not always execute. A do...while will run evaluate the statement AFTER running through a cycle of the loop, ensuring that the loop always executes at least one time.

8. What is typically included in a class definition?

member functions

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

data members are permanent attributes of the class, local variables inside a member function are lost when the scope of the member function ends.

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

A constructor does not return any data and has no return type. Functions can return data or not return data can have a return type.

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

when there is not one already for a specific class

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

1

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

A function prototype is a declaration of a function that tells the compiler the function's name, its return type and the types of its parameters.

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

No

15. What does a function signature include?

The function signature includes the name of the function and the types of its arguments.

16. What is the scope of global variables?

Any functions that follows their deceleration that are in the source code file are allowed to have access the variables. Also global scope variables retrain their value throughout the execution of the program.

17. What is the scope of global variables?

global variables have program scope (accessible anywhere in program)

18. How does the compiler handle inline functions?

Expands the function into the program

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

It gives access and modify the caller's argument data directly. It eliminate the pass by value overhead of copying large amounts of data

20. How are overloaded functions differentiated by the compiler?

The compiler differentiates overloaded functions by their signature.

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

omitting the base case or writing the recursive call so that it does not converge upon the base case.

22. What are the similarities between iteration and recursion?

based on a control statement, involve repetition, involve a termination test.

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

Implicit, namely when you give it values, like {1,2,3,4,5,6,7,8,9}, at which the compiler will automatically give this a size. Static, during the declaration.

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

strings with type string are just that. Strings. They are not part of an array list at all where as one declared by an array is actually an array of characters able to be pointed and detected.

25. How are arrays passed to functions?

You pass the name of the array to the function. So for: int bin [10]; you pass it to an array like so: jingle(bin); but jingle must be declared correctly in order to receive an array, either with prototype void jingle (int []); or with the definition in its initialization: void jingle (int array[]); It's also worth noting that arrays are passed to function by reference, so as to not copy large arrays over and over and burn memory. This gives the function complete control over the array however.

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

A static member prevents naming conflicts (not put into global namespace) while allowing for information hiding (private, public).

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

the size of the first dimension does not need to be specified, but all other dimensions do.

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

have a marker and sort everything to the left of the value for every new value when moving to the right.

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

The main idea behind selection sort is to take the least variable from the right and put it directly to the left.

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

In the best case scenario of an already sorted list, insertion sort will progress once through the list. So the number of operations is directly proportional to the number of elements in the list. Also noted as $O(n)$ in big o notation.

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

where you only must merge sort once

32. What is a pointer?

variable that points to the memory address of another variable and it is type specific.

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

returns the address number of the specified variable

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

a reference to a = variable

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

multi-dimensional array

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

#NAME?

37. What is a function pointer?

when it is dereferenced a function pointer calls its function with zero or more arguments.

38. What is a linked list?

one of the fundamental data structures, and can be used to implement other data structures. It consists of a sequence of nodes, each containing arbitrary data fields and one or two references ("links") pointing to the next and/or previous nodes.

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

Linked lists have constant time insertion and deletion

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

direct access to elements, and speed, linked lists have increased search times to access elements even if the list is sorted. (like getting the last element without a pointer directly to the end of the list.)

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

the head pointer is passed to a function as a reference argument.

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

In circular linked list each node has successor, no node contains NULL, while in basic linked list last node contain NULL.

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

a doubly linked list allows you to back up from a node rather than having to go through the entire list again

44. What is a stack?

A data structure for storing items which are to be accessed in last-in first-out order that can be implemented in three ways.

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

Push and pop.

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

Use a pointer that always points to the end of the array list for push/pop modifications.

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

You dynamically create space as the stack grows. The stack can be as long as you need.

48. How are infix expressions evaluated by computers?

They are first converted into an equivalent postfix expression so it can be read by the computer.

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

pop and push

50. What is a queue?

A queue is a first in first out data structure.

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

enqueue (Data), Dequeue ()

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

add items into a fixed size array, pull the items out by calling on them with the starting index, increase starting index so when it calls again its now the next index element in the array.

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

linked list checking for front and rear. Keep enqueueing elements until front equal to rear -1.

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

Push corresponds with enqueue, both very similarly place new elements into their stack or queue

55. What is a tree?

A finite number of nodes such that there is one root node, each node has a certain number of children nodes, and each child is also a tree

56. What is a leaf?

A tree node with no children.

57. What is a binary tree?

a binary tree is a tree data structure in which each node has at most two children.

58. What is a binary search tree?

a binary tree where the children are ordered such that the right side is greater than the current node, and the left is less than or equal the current node.

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

Inorder traversal visits the left branch of a node (all the way to it's deepest left-most leaf) before visiting the node itself. After visiting the entire left branch and the node, it visits the node's right branch following the same pattern of left child, node, right child.

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

2 way key comparison

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

class variables, class function prototypes

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

public and private

63. How are objects initialized when they are created?

By declaring the class first, the name of the variable, and then any variables that need to be put in the constructor. Ex. Class myClass(x, y)

64. What is a function signature?

function name and parameters

65. What is a recursive function?

It invoke by itself again and again, unless it reach base case

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

Any problem solved recursively could be solved with an iterative function. Iterative replacements for recursive functions may be more difficult to program, but often lead to more efficient solutions to a problem.

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

the data of a static array cannot be changed

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[]?

When the string is passed into the array char, individual characters are stored and can be manipulated. Although when a word is stored into a string variable, it is stored as one word and not as individual characters.

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

a divide-and-conquer paradigm. takes some data, divides it into two parts, and works on each part individually until the item is found.

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

merge sort breaks the array down in halves until it is comparing two values, sorts those two values, then merges that back with the other broken down parts that it sorted, each level merging more sets together till you return back to your whole array in order.

71. What is a pointer?

A pointer is an element that references a memory location.

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

the experimental approach measures actual running time in t. t= seconds.

73. Briefly, how does selection sort work?

selection sort searches for the smallest element of the remaining data organization and places it at the largest point of the new data organization

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

The primary advantage of linked lists over arrays is their ability to become larger or smaller as data is added and removed from the list, whereas arrays may waste memory for small data sets, and run out of space for large data sets.

75. What is a queue?

Queue is a buffer, it follows the rule of First In and First out. Queue is very useful in computer science for scheduling.

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

pop and push to remove an element and to insert an element

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

You recursively visit every node on the list. You visit the node from the left, the bottom, and from the right.

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

to delete the node, you would have to link the children nodes that are connected to the node to be deleted to the remaining nodes of the tree in such a way that nodes on the right of the parent node are larger than the parent and nodes on the left of the parent node are smaller