

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

The role of a prototype program is to help spot key problems that may arise during the actual programming.

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

Refining stage, Production stage, and Maintenance stage.

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

Information can be hidden. It is easier to debug. Programming is easier and more manageable.

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

at the main function `int main() {}`

5. What is a variable?

A variable is the location in a computer's memory where a value can be stored for use by a program.

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

Variables are usually declared at the very beginning of a C++ program, but can be declared any in the program as long as they appear before they are called upon.

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

NO ANSWER

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?

Variable declared in a function definition's body cannot be used outside of that function. Data members accessible only to member function of that class.

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

a constructor is a function that initializes a class instance, a function performs whatever task it is written for

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

C++ creates a default constructor during compilation if one is not provided.

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

one

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

A function prototype just specifies parameters. A function definition includes parameters and a code body.

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

Header files have reusable source code in a file that a programmer can use.

15. What does a function signature include?

The name of the function and the types of its arguments. This includes the number, type, and order the parameters appear in.

16. What is the scope of global variables?

it is accessible by all functions within a program

17. How does the compiler handle inline functions?

Inline functions are copied in place, instead of being referenced.

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

It is more efficient and it changes the variable not online inside the function but outside so that the new value can be used elsewhere.

19. How are overloaded functions differentiated by the compiler?

Is based on number, types, and order of arguments in the function call.

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

If at each recursive call the problem does not diminish or if it does not diminish to a base case, can cause infinite recursion.

21. What are the similarities between iteration and recursion?

Both will repeat (loop) until a condition is met, and exit the loop. Both are capable of entering an infinite loop.

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

One way is to declare a size of the array using a variable that is equal to an integer. The other way, if the size is omitted in the declaration, the compiler determines the size of the array based on the size of the initializer list.

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

All strings represented by character arrays end with the null character. You declare an object of type string just like any other type, for example: string s;

24. How are arrays passed to functions?

Arrays are passed to functions by reference.

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

a static array can only be referenced not accessed. Also static arrays can be used without creating an object.

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

at least 2, depending on how many arrays are being used.

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

Multi dimensional arrays are accessed using more than 1 index, one for each dimension at least.

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

Searches an array for the largest or the smallest element and is moved out to the front of the array then it will search for the next largest or the smallest element to move out to the front

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

Insertion sort is $O(n)$ and performs $n-1$ operations in the best-case scenario, which is when the list or array is already sorted.

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

The base case for a recursive implementation of merge sort is one.

31. What is a pointer?

Pointers are variables that contain as their values memory addresses of other variables.

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

The address of the where the data is stored: &b; will return the address of b.

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

* returns the synonym for the object its operand points to.

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

The pointer to the array will initially point to the first element of that array. When the pointer is incremented, it will point to the next element, and so on.

35. What does the sizeof operator return?

the size in bytes needed to store the object in memory

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

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

37. What is a function pointer?

Function pointer is a pointer to a function. function pointer contains address of functions

38. What is a linked list?

A list of objects where each object contains a link to the next item in the list

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

Arrays require less memory than linked lists and allow direct access to a specified item. Also, insertion and deletion in arrays does not require a list traversal.

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

Pass-by-reference

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

In a circular linked list, the last node points back to the first node, there is no NULL

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

You can point both forwards and backwards

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

its a little more confusing. The special cases at the end and beginning become more difficult to do.

44. What is a stack?

an object list that stores elements in a particular order, the first object inserted is at the bottom with the last object at the top so the first object in is the last object out.

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

Push and Pop

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

By allocating an array of predetermined size, and an integer to track the top element of the stack. The bottom member of the stack will go in element 0 of the array, and for each element pushed the "top" tracking integer is incremented.

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

the top is the first pointer.

48. How are infix expressions evaluated by computers?

infix expressions are converted to postfix, (i.e. $3+2$ is changed to $32+$)

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

StackPush() StackPop() StackIsEmpty()

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

enqueue , which inserts a specified element at the head of the specified queue, and dequeue , which removes the specified element from the queue.

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

For applications in which a fixed-sized queue does not present a problem, you can use an array to represent a queue.

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

Enqueue to the end of the linked list... dequeue items from the beginning (head) of the list.

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

Not sure what this question means. You can use the stack 'push' operation and the enqueue operation to detect palindromes.

54. What is a tree?

one or more nodes in a hierarchy starting with the root and branching off like a tree to subtrees

55. What is the height of a tree?

The number of nodes on the longest path from the root to the leaf.

56. What is a leaf?

an element in a tree that has no children

57. What is a binary search tree?

A tree data structure where each node has a value, where the total order is defined on these values. the left subtree contains only values less than the parent node, and vice versa for the right subtree.

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

printing a tree in order from least to greatest. This done by going as far left down the tree as possible and printing the parent and then right tree. Then move up the tree.

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

$O(n \log(n))$

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

Parameters and their type, Class name, and return type, and the code to be executed in the class.

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

A C++ class definition may include access-specifiers public, private, and static.

62. How are objects initialized when they are created?

When the object is initialized, its constructor is called (whether one is provided or c++ assigns a default constructor).

63. What is a function signature?

the body of the function that holds all the actual code

64. What is a recursive function?

A function that calls itself, in order to solve a problem.

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

Another alternative way to solve a problem instead of using recursive functions is by using iteration.

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

an array that is declared as static will retain the values stored in its elements between function calls, and will not reinitialize them to default values.

67. 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 you read a string of characters, you also read in the white space. When you read characters declared char you only read the characters, no white space.

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

divided and conquer is the breaking up of a large complex program into small easy to solve components.

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

Continually split array in half, sort each side of the halves, then put them back together once organized.

70. What is a pointer?

a pointer points to a location in memory of a certain data type

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

Opposite of a theoretical assessment of the algorithm to determine runtime, but to run the code first to determine the runtime. This is not recommended because it is a limited test. It does not include all possibilities of the data, nor the hardware used to process the data.

72. Briefly, how does selection sort work?

A selection sort scrolls through the data structure looking for the lowest (or highest) unsorted piece of data, moves it to the held spot, increments that spot by one, and starts the process over.

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

Linked lists are able to grow in size. So elements can be added to the list.

74. What is a queue?

A queue is a data type that operates under a FILO (First In Last Out) method. This means that the first element inserted into the queue is the first element removed.

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

pop and push

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

go to the bottom of the left sub tree and visit the parent and then its children

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

If the node is a leaf, just set its parent's pointer to null and delete it, if it has a single child, set the parent's pointer to the child and delete; if it has two children, set the node to one of the middle children and remove that child from its previous position as a leaf (rightmost child of the left subtree or leftmost child of the right subtree).

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

If the binary search tree is constructed efficiently, best case scenario is $O(\log n)$ time. Where n is the number of items in the tree. If the binary search tree is constructed poorly, with for instance the root of the tree being 1, and progressing downwards and to the right its children are each more than the last: you have a one-way linear linked list. That worse case scenario would be a full traversal at $O(n)$ time. Where n is the number of items in the tree.