

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

Defined in the Specification phase a prototype stimulates the behavior of portions of the desired software product. Meaning, the role of a prototype is a temporary solution until the program itself is refined to be used extensively in problem solving.

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

In RUP the stages in the software life cycle are influenced by the testing stage is: Elaboration phase- refined project vision, iterative development of core system, development of system requirements, more accurate time and cost estimates. Construction phase- iterative development of remaining system. Transition phase- testing and deployment of the system.

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

The advantages is that OOP allows us to build classes of objects. Three principles that make up OOP are: Encapsulation- Objects combine data and operations. Inheritance- Classes can inherit properties from other classes. Polymorphism- Objects can determine appropriate operations at execution time.

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

A C++ program will begin to execute at the main() function.

## **5. What is a variable?**

Variable is a location in the computer's memory, in which a value can be stored and later can retrieve that value.

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

Variables are declared where ever it is defined either inside a block locally or outside a block globally.

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

While loop used to execute a block of code as long as some condition is true. A do...while loop used to execute a block of code as long as some condition is satisfied. The difference between while and do...while is while loop tests its condition before the execution of its contents and the do...while loop tests its condition after the execution of its contents at least once.

## **8. What is typically included in a class definition?**

Since class is user defined, the body of the declaration can contain members, that can be either data or function declarations, and optionally access specifiers. Example for specifiers would be public, private, and protected. There is the keyword Class and the Class body is enclosed with brackets ({}).

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

Data members are declared inside the class, but outside of any member functions. Unlike with local variables, each object of the class keeps its own copy of the data members stored in memory.

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

all constructors are functions but not all functions are constructors. Also functions can have nearly infinite purposes, while constructors always create an instance of whichever class you are using. You can change how a constructor does something but not what it does. With functions you can change not only the how but also the what.

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

The compiler creates a default constructor when we do not define one in our class file

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

at the very beginning when you run a compiler, matter a fact in C++ it generates its own default constructor for the class provided if no other constructors have been provided by user

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

as many as you want, as long as they each have a unique argument list

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

function prototype describe the class's public interface without revealing the class's member function implementations, function definitions show what implementations are being done

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

To promote function reusability

## **16. What does a function signature include?**

The portion of a function prototype that includes the name of the function and the types of its arguments

## **17. What is the scope of global variables?**

If a variable is declared global, any function or class can use that variable.

## **18. How does the compiler handle inline functions?**

Multiple copies of the function code are inserted into the program, making it bigger.

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

gives function ability to access and modify the caller's argument data directly

## **20. How are overloaded functions differentiated by the compiler?**

their function signature.

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

If the recursion function never reaches or successfully defines the base case it will recurse forever. This happens many ways, such as the function doesn't progress towards the base case, or the function is coded poorly and doesn't even contain a base case.

## **22. What are the similarities between iteration and recursion?**

anything you can do recursively you can do iteratively

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

statically, using {} and dynamically, using []

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

string declared used in an array of characters contains each character in the array and a special string-termination character called the null character versus the type string.

## **25. How are arrays passed to functions?**

In two parts, the array itself, and it's size, so that the function it's being passed to can avoid out of bounds errors.

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

If an array is declared as static, it is not created each time the array is called.

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

just the first one at least.

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

insertion sort is were after k iterations the first k items in the array are sorted it take the k+1 item and inserts it into the correct position in the already sorted k elements.

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

Find the smallest and put it in the current position till you get to the end.

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

The best case scenario for an insertion sort, where all the elements of the array have been already sorted, will cause the for loop to iterate "x" times ("x" being the number of elements in the array) and the while loop to never have to iterate, resulting in the insertion sort to require "x" operations.

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

$T(n) = 0$ , if  $n < 2$ .

### **32. What is a pointer?**

It is like a variable however instead of holding data, it holds a location in memory.

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

The & operator returns the memory address of the variable it precedes.

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

the value of the variable the pointer points to.

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

by referencing a pointer and referring to other pointers relative to the first pointer(pointer+1, pointer +2, etc)

### **36. What does the sizeof operator return?**

the total number of bytes of an object.

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

there are only two ways to pass something, by value and by reference. Use of dot or pointer operators within the receiving function and use of '&' operator inside the passing function

### **38. What is a linked list?**

A linked list is a chain of structs or records called nodes and you have the ability to add, delete, or retrieve items.

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

The principal benefit of a linked list over a conventional array is that the order of the linked items may be different from the order that the data items are stored in memory or on disk, allowing the list of items to be traversed in a different order. A linked list is a self-referential datatype because it contains a pointer or link to another datum of the same type. Linked lists permit insertion and removal of nodes at any point in the list in constant time

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

Linked list has an increase complexity of the implementation. The linked list has an array that keeps growing which leads to a increase in memory.

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

passes a pointer to a linked list object

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

how to determine the end of the list, in basic linked lists the last element links to a null pointer, while circular linked lists link to the head element at the end.

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

each node points to both its predecessor and its successor.

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

Double linked lists require more space per node, and their elementary operation are more expensive

#### **45. What is a stack?**

A data structure that puts elements in a list and only allows the user access to the last element.

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

push - adds an item to the top of the stack pop - removes the top item from the stack

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

You can add an element to the end of the linked list and update a pointer called top each time you modify the list.

#### **48. How are infix expressions evaluated by computers?**

Computers cannot evaluate infix expressions, it must first convert infix to postfix expression then it can evaluate.

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

Pop each item off the top until you find what you are looking for. If the stack ends up empty and you never found the item, then it does not exist in the stack. It may also be necessary to save the popped values in case you want to recreate the stack.

## **50. What is a queue?**

a queue is a data structure that stores elements in a First in First out order.

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

The two main functions of a queue are enqueue and dequeue.

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

make an array of a size, and add on to the front and delete from the back, keep track of the two so that you know when it is full and where to add or subtract from

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

throw in data at the head and take it out at the tail

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

that would be the push operation, if it put the item at the end of the list.

## **55. What is a tree?**

A finite set of nodes that starts with the root and terminates with leaves.

## **56. What is the height of a tree?**

the number of distinct levels of separation

## **57. What is a binary tree?**

A set of nodes that is either empty or partitioned into a root node and 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?**

Binary search trees are similar to binary trees, but have been implemented to organize data in a specific way for later searching.

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

it goes from lowest to highest inside of the tree

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

it takes as many comparisons as the number of levels of the binary search tree. it only compares one number at each level

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

class name data of class definition of functions and methods

**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?**

initialized using the same name as the class

**64. What is a function signature?**

its the scope, type, and parameters of that function that give it its properties and defines it against other functions.

**65. What is a recursive function?**

A recursive function only knows how to solve base cases A recursive function calls itself directly or indirectly until a base case is reached.

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

using an iterative calls

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

one declared as static is one that is already defined. The program knows the length and the array from the start, whereas non-static arrays are declared or assigned later.

**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[]?**

The variable of type string has a terminating character \0 at the end of it.

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

The divide and conquer paradigm, splits a large problem into simpler problems, at which point it solves the simpler problems and merges the simple solutions together to answer the large problem.

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

A merge sort recursively divides the array into half until only one element remains, then it sorts the data on its way out of the recursive call by merging the cells.

**71. What is a pointer?**

A pointer is an alias to an object in memory.

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

You run a program with different data sizes, like  $10^x$ . As you increase  $x$  and measure the completion speeds for the program, you can find patterns and attempt to measure the running time. It's very important to keep the same software and hardware however, which makes experimental testing inferior to theoretical in the eyes of most.

## **73. Briefly, how does selection sort work?**

it runs through the list and finds the smallest (or largest) value and puts it at the appropriate position

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

Linked lists have no predefined length, and thus you can have virtually unlimited items in that list.

## **75. What is a queue?**

A linked list with a First In, Out structure dequeue at the head of the list enqueue at the end of the list

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

push and pop. push puts objects into the stack and pop takes them out following the first in last out rule.

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

The Euler traversal is when you go through a tree looking for nodes to put in the order of left, root, right. Like inOrder traversal.

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

You must first traverse the tree to find the appropriate value. Then you must make sure that the node is a leaf node. If it is, then you can delete the pointer to that specific node.