C++ / OOP

- What is procedural programming?
- What is object based programming
- Concrete or abstract data types
- What is object oriented programming?
- Inheritance
- Polymorphism (static polymorphism, dynamic polymorphism)
- Operator overloading
- Resource acquisition is initialization
- Static and const member data and member functions
- Virtual member functions
- Abstract classes and interfaces
- implicit/explicit constructor calls of base class, member variables
- deciding between inheritance and containment friendship
- Non-public inheritance
- Forward declarations
- Common root class
- Special member functions
- Operator(), conversion operator
- Local classes
- Multiple inheritance
- Virtual inheritance
- Why we can need non-virtual multiple inheritance
- Traditional exception handling mechanisms
- - drawbacks of traditional error handling mechanisms used in C language
- - exception handling in C++ (try, throw, catch)
- Exception type match
- Exception specification
- Concordance of exception specification
- Exceptions during construction and destruction
- Checking for an uncaught exception
- Standard exceptions
- Exception handler hierarchy
- Rethrowing an exception
- Function try block

Data Structures

1. What is a Data Structure?
2. What are linear and non linear data Structures?
3. What are the various operations that can be performed on different Data Structures?
4. What is an Array?
5. What is a Vector?
6. How is Vector different from Linked List?
7. What is Stack and where it can be used?
8. What is a Queue, how it is different from stack and how is it implemented?
9. What is a Dequeue?
10. What is a Linked List and What are its types?
11. How to find the middle of a given linked list?
12. How to check the linked list is circular or not?
13. What is a Set(hash and based on the binary search tree)?
14. What is a Map(hash and based on the binary search tree)?
15. Explain Binary Search Tree.
16. Explain Binary Search.
17. How do you insert a new item in a binary search tree?
18. How do you traverse a given binary tree in preorder(with recursion so)?
19. How do you traverse a given binary tree in inorder(with recursion so)?
20. How are all leaves of a binary search tree printed(level-order)?

Algorithms

- Basic sorting, searching algorithms
- How and why to use binary tree
- Simple algorithms complexity
- Basic sorting, searching, algorithm complexity
- Reasonable knowledge of
    - divide and conquer, dynamic programming algorithms
    - matrix algorithms (e.g. image processing)
    - simple tree algorithms (e.g. binary tree construction, search, etc.)
    - simple graph algorithms (e.g. Dijkstra, low cost assignment)

Systems programming

- Compiler/interpreter differences
- Understands the main phases of compilation process (preprocessing/compiling/assembling/linking).
- What assembly code is and how things work at the hardware level
- The difference between static and dynamic linking
- Used a debugger for troubleshooting/fixing code issues
- The difference between debug and release build, several compiler flags
- Debugger, call stack, breakpoints, debug commands
- Make String to number conversion.

Build process

- Which are the differences of interpretative and compiling languages?
- Which are the main stages of compilation?
- What is the input/output of each stage?
- Which are the differences of static/dynamic libs?

GNU make

- What is GNU Make?
- How many exit statuses does make have? (explain all statuses)
- What is Makefile?
- What name can have Makefiles?
- How does 'make' process a Makefile?
- Enumerate the capabilities of make.
- Explain the variable defining and assignment in Makefiles.
- Enumerate and explain the built-in functions used in Makefiles.
- What does .PHONY mean in a Makefile?
- How to automatically generate the dependencies of C++ files?

Gcc/gdb

- Which are the options to stop at an intermediate stage?
- Several options for each stage.
- How to compile several files?
- Kind of libraries. Disadvantages and advantages of each kind.
- What is GDB?
- How to run program under GDB?
- How to start and stop debugging?
- What is breakpoint/watchpoint/catchpoint?
- What is conditional breakpoint?
- How to set/delete/enable/disable the breakpoint/watchpoint/catchpoint?
- Enumerate and explain commands for debugging.
- What is stack frame?
- Enumerate and explain commands for examining the stack.