**C++ Interview questions**

1. C++ Core concepts
   1. Constructor and destructors
   2. Operator overloading and function overloading
   3. Virtual class and friend class
   4. Polymorphism
   5. Virtual functions
   6. Virtual destructors
   7. Abstract classes
   8. File operations
   9. Constant, mutable
   10. Templates
   11. RTTI
2. STL
   1. Vector
   2. List
   3. Map
   4. Hash table
   5. unordered\_map, unordered\_set
3. Class diagram for your project
4. Design Patterns, Provide use cases
   1. Creational
      1. Singleton design pattern
      2. Abstract factory
   2. Structural
      1. Proxy and Adaptor
      2. Composite
   3. Behavioral
      1. Observer
      2. visitor
5. Multi-threading
   1. Semaphore
   2. Mutex

**Mutex**: Mutual exclusion, for synchronization of two processes, any of this two process can have KEY (MUTEX) to proceed there work, implemented as locking mechanism.

**Semaphore**: Generalization of mutex, implemented as signaling mechanism

**Binary semaphore**:

**Main Difference between mutex and semaphore**

Mutex is Locking Mechanism.

Semaphore is Signaling mechanism.

**Recursive and Non recursive mutex:**

**Shared Mutex**

**Read/Write mutex**

**Spin locks**

**C++ Mutex types**

1. **std::mutex**
2. **std::timed\_mutex**
3. **std::recursive\_mutex**
4. **std::recursive\_timed\_mutex**
5. **std::shared\_timed\_mutex**

**C++ Lock types**

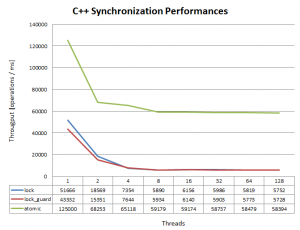
1. **std::lock\_guard<>**
2. **std::unique\_lock<>**
3. **std::shared\_lock<>**

**Performance comparisons**

A single std::mutex with calls to lock() and unlock()

A single std::mutex locked with std::lock\_guard

An atomic reference on the integer



Ref : <http://baptiste-wicht.com/posts/2012/04/c11-concurrency-tutorial-advanced-locking-and-condition-variables.html>

C++ /OOPs interview Questions

1. What is OOP?

* Philosophy of design and implementation, which is based on concept of object and classes.
* Programming language Model, organized around object
* Historically programming has been viewed as logical procedure
* It is not procedural programming.

1. What are building blocks of OOP?
   1. Classes and objects
   2. Encapsulation
   3. Data Abstraction
   4. Inheritance
   5. Polymorphism
2. Explain classes and objects in details.
   1. Concept

* It is just a blueprint
* Defines basic characteristics and behaviors of similar kind of data.
  1. Give one/two theoretical example (simple bookish example)
  2. Give one/two classical example (Practical / real example)
  3. Give one/two example from your project (Actual example on which you worked)

1. Explain Encapsulation in details.
   1. Concept

* Data hiding
  1. Give one/two theoretical example (simple bookish example)
  2. Give one/two classical example (Practical / real example)
  3. Give one/two example from your project (Actual example on which you worked)

1. Explain abstraction in details.
   1. Concept

* Exposing only essential details and hiding other irrelevant details.
* Reduces code complexity
  1. Give one/two theoretical example (simple bookish example)
  2. Give one/two classical example (Practical / real example)
  3. Give one/two example from your project (Actual example on which you worked)

1. Explain Inheritance in details.
   1. Concept

* Derive new type of objects from existing objects.
  1. Give one/two theoretical example (simple bookish example)
  2. Give one/two classical example (Practical / real example)
  3. Give one/two example from your project (Actual example on which you worked)

1. Explain polymorphism in details.
   1. Concept
   2. Give one/two theoretical example (simple bookish example)
   3. Give one/two classical example (Practical / real example)
   4. Give one/two example from your project (Actual example on which you worked)
2. What is mean by Generalization?
   1. Concept
   2. Give one/two theoretical example (simple bookish example)
   3. Give one/two classical example (Practical / real example)
   4. Give one/two example from your project (Actual example on which you worked)
3. What is mean by Specialization?
   1. Concept
   2. Give one/two theoretical example (simple bookish example)
   3. Give one/two classical example (Practical / real example)
   4. Give one/two example from your project (Actual example on which you worked)
4. What is mean by Composition?
   1. Concept
   2. Give one/two theoretical example (simple bookish example)
   3. Give one/two classical example (Practical / real example)
   4. Give one/two example from your project (Actual example on which you worked)
5. What is mean by Association?
   1. Concept
   2. Give one/two theoretical example (simple bookish example)
   3. Give one/two classical example (Practical / real example)
   4. Give one/two example from your project (Actual example on which you worked)
6. What is mean by Aggregation?
   1. Concept
   2. Give one/two theoretical example (simple bookish example)
   3. Give one/two classical example (Practical / real example)
   4. Give one/two example from your project (Actual example on which you worked)
7. What are types of inheritance? Give practical example of each type.
8. What are types of polymorphism? Give practical example of each type.
9. What is copy constructor and copy assignment operator?
10. What is mean by shallow copy and deep copy? When to use which copy?
11. What are destructors? Why we need to use virtual destructor?
    1. Case when you have multiple inheritance
    2. Case when you multilevel inheritance
    3. Case when you have hybrid inheritance
    4. So each all above cases, which call destructors, you will define virtual?
12. What are pointers, what are references?
13. Explain following concepts along with example.
    1. Smart pointers
    2. Unique pointers
    3. Shared pointers
    4. Auto pointers
    5. Dangling pointers
    6. Void pointer in C vs void pointer in C++
14. Difference between malloc and new, free and delete, with example.
15. Allocate memory for 1d array, 2d array using new operator.
16. What is stack memory and heap memory?
17. What is stack unwinding?
18. Operator overloading examples
    1. Pre increment and post increment overloading
    2. New/delete operator overloading
    3. Operator overloading with help of friend functions.
    4. Conversion operator
    5. Conversion constructor
19. What are namespaces? What is koening lookup? How it is useful?
20. What are exception? How to handle exceptions? Have you used them any of your programs? Please explain with examples
21. What are templates?
    1. What are function templates
    2. What are class templates
    3. Write a program for class template
    4. Write a program for function template.
22. What are all string operations?
    1. Strrev
    2. Strcpy
    3. Strstr
    4. Substr
23. What is RTII?
24. Explain each cast.
25. What is RAII?

Multithreading and Multi Processing and OS Concepts.

1. What is process?
2. What is thread?
3. What is process block? What are elements of process block?
4. What is thread block? What are elements of thread block?
5. What are different process states?
6. What is difference between process and thread?
7. How to create Process in C program?
8. How to create thread in C program?
9. What is Zombie process? How to create zombie process? How to control it?
10. What are different segment?
11. BSS segment?
12. In which segment Global variable stored? And In which segment dynamically allocated memory is stored?
13. What is Re-entrant code?
14. What is difference between logical and physical address space?
15. What is Deadlock?
16. What are deadlock detection algorithms?
17. What is starvation?
18. How to resolve Deadlock?
19. How to prevent Deadlock?
20. What is Daemon?
21. How you will decide, whether to use multithreading or multiprocessing for a program?
22. Provide advantages and disadvantages of multi-threading and multiprocessing.
23. What are different inter process communication mechanisms?
24. What are different inter thread communication mechanisms?
25. What are Volatile variables?
26. What are different thread attributes?
27. What is concurrency?
28. What is parallelism?
29. What is semaphore?
30. What is mutex?
31. How to achieve synchronization in following cases?
    1. Single read thread and single write thread.
    2. Single write thread and multiple read threads.
    3. Multiple write threads and multiple read threads.
32. What are different synchronization mechanisms? Explain each with detail example.
33. What are different types of semaphores?
34. What are different types of mutex?
35. What is difference between semaphore, mutex and binary semaphore?
36. How to know how much memory used by process?
37. What is paging?
38. What is page thrashing?
39. What is fragmentation?
40. What is external fragmentation?
41. What is internal fragmentation?
42. What are interrupts?
43. What is difference between compiler and interpreter?
44. What is marshaling?
45. What is priority inversion?
46. What is context switching?
47. What is process spawning?
48. What is difference between socket and pipe?
49. What are types of scheduling?
50. Explain short, long and medium term scheduling.
51. What is busy waiting?
52. What are popular multi-processor thread-scheduling strategies? Explain following strategies.
53. Load sharing
54. Gang scheduling
55. Dedicated processor assignments
56. Dynamic scheduling
57. What are reasons for process suspensions?

Ans : swapping, interactive user request, timing, parent process request.

1. What is process migration?
2. Explain memory partitions, paging and segmentation.
3. What is monitor?
4. What is difference between semaphore and monitor?