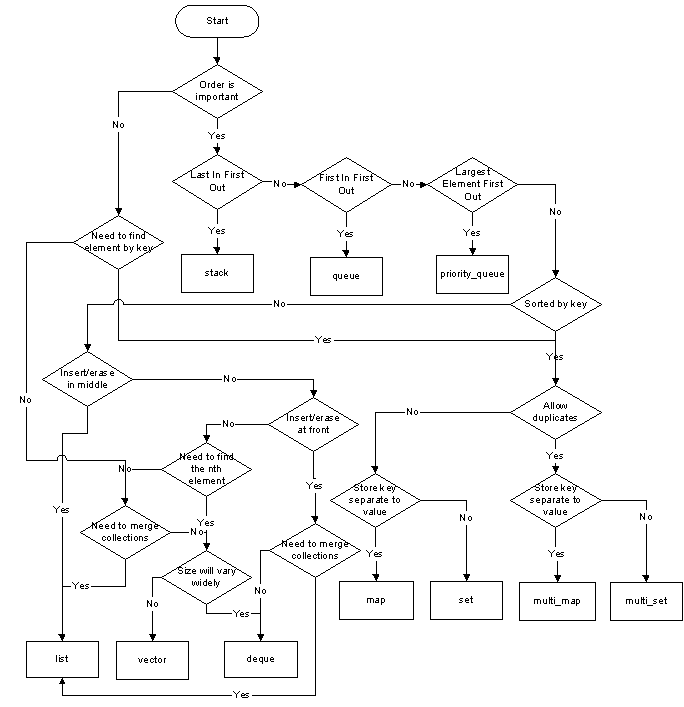
STEP 1:

1. Explain the different types of containers available in the standard library, what are their differences? Explain use cases for each. <https://en.cppreference.com/w/cpp/container> <https://stackoverflow.com/questions/4010097/general-use-cases-for-c-containers>



2. What are some sorting algorithms, list at least 3? What are their differences and give use cases for each.

- c++ std::sort() – Uses the introsort algorithm which is a hybrid of quicksort, heapsort, and mergesort. It begins with [quicksort](https://en.wikipedia.org/wiki/Quicksort), it switches to [heapsort](https://en.wikipedia.org/wiki/Heapsort) when the recursion depth exceeds a level based on (the [logarithm](https://en.wikipedia.org/wiki/Logarithm) of) the number of elements being sorted and it switches to [insertion sort](https://en.wikipedia.org/wiki/Insertion_sort) when the number of elements is below some threshold. This combines the good parts of the three algorithms, with practical performance comparable to quicksort on typical data sets and worst-case [O](https://en.wikipedia.org/wiki/Big-O_notation)(*n* log *n*) runtime due to the heap sort. Since the three algorithms it uses are [comparison sorts](https://en.wikipedia.org/wiki/Comparison_sort), it is also a comparison sort.

3. What is the purpose of virtual destructors? What types of issues can arise if not used correctly.

A virtual destructor on the base class ensures the destructor is called on the derived class when the derived class is contructed polymorphically with the base class, otherwise you may have undefined results.

4. Explain the keyword: static. What does it mean in each context?

Static is a keyword in C++ used to give special characteristics to an element. Static elements are allocated storage only once in a program lifetime in static storage area. And they have a scope till the program lifetime. Static Keyword can be used with following,

1. Static variable in functions - Static variables when used inside [function](https://www.studytonight.com/cpp/functions-in-cpp) are initialized only once, and then they hold there value even through function calls.
2. Static Class Objects - has scope till the program's lifetime, instead of the lifetime of the class object.
3. Static member Variable in class - Static data members of class are those members which are shared by all the objects. Static data member has a single piece of storage, and is not available as separate copy with each object, like other non-static data members.
4. Static Methods in class - These [functions](https://www.studytonight.com/cpp/functions-in-cpp) cannot access ordinary data members and member functions, but only static data members and static member functions.

5. When are static member variables initialized?  
static member variables (data members) are not initialized using constructor, because these are not dependent on object initialization.

Also, it must be initialized explicitly, always outside the class. If not initialized, Linker will give error.

6. What is the difference between R-Values and L-Values?

In C++ an lvalue is something that points to a specific memory location. On the other hand, a rvalue is something that doesn’t point anywhere. In general, rvalues are temporary and short lived, while lvalues live a longer life since they exist as variables.

7. Is this code safe? If so why? If not why?  
std::string foo()  
{  
std::string something = �avalue�;  
return something;  
}

This code is not safe because is variable ‘something’ is local to the function and will be deallocated when function ends. A better way would be to define it as ‘static string something = ‘avalue’;

Bonus: What would most compilers do with this code? Visual Studio 2019 did not complain.

8. Why would you use new rather than malloc when allocating an object?

**Calling Constructors:**new calls constructors, while malloc() does not. In fact primitive data types (char, int, float.. etc) can also be initialized with new

**b. operator vs function:** new is an operator, while malloc() is a function.

**c. return type:** new returns exact data type, while malloc() returns void \*.

**d. Failure Condition:**On failure, malloc() returns NULL where as new throws bad\_alloc exception

**5. Memory:** In case of new, memory is allocated from free store where as in malloc() memory allocation is done from heap.

**6. Size:** Required size of memory is calculated by compiler for new, where as we have to manually calculate size for malloc().

**7. Buffer Size:** malloc() allows to change the size of buffer using realloc() while new doesn’t

Likewise, what�s the difference between free and delete?

In C++, delete operator should only be used either for the pointers pointing to the memory allocated using new operator or for a NULL pointer, and free() should only be used either for the pointers pointing to the memory allocated using malloc() or for a NULL pointer.

The most important reason why free() should not be used for de-allocating memory allocated using NEW is that, it does not call the destructor of that object while delete operator does.

9. Explain the purpose of std::move, and std::forward.

Std::move changes variable to rvalue so it can be deallocated instead of making a copy.

std::forward is more magical. When you do std::forward<T>, T is sometimes a reference (when the original is an lvalue) and sometimes not (when the original is an rvalue). std::forward will therefore cast to an lvalue or rvalue reference as appropriate.

You cannot make this work in the non-template version precisely because you'll have only one type available. You have to specify the template type in std::forward.  *official* name for "universal reference" is now *forwarding reference.*

10. How do you share resources safely between threads?

As we can see from the output, the two threads get the cout resource in a ramdom fashion. To have a deterministic access, the code below is using mutex: lock before accessing cout, and then unlock after using it:

How would you share an integer value vs user defined type?

You can use a std::mutex with std::lock\_guard to synchronize access to the shared data. Or if the shared data fits in an integer, you can use std::atomic<int> without locking.

11. What are the some of the principles of object-oriented programming?  
Inheritance, encapsulation, polymorphism, abstraction, SOLID

12. Explain inheritance vs composition vs aggregation?

Inheritance vs composition - Composition is usually used for wrapping classes and to express relationships between classes that contain one another. Inheritance is used for polymorphism, where you have a base class and you want to extend or change its functionality

Inheritance vs aggregation - The difference is typically expressed as the difference between "is a" and "has a". Inheritance, the "is a" relationship, is summed up nicely in the Liskov Substitution Principle. Aggregation, the "has a" relationship, is just that - it shows that the aggregating object has one of the aggregated objects

Composition vs aggregation - Aggregation implies a relationship where the child can exist independently of the parent. Example: Class (parent) and Student (child). ... Composition implies a relationship where the child cannot exist independent of the parent. Example: House (parent) and Room (child).

13. Should you always initialize variables?

Unlike some programming languages, C/C++ does not initialize most variables to a given value (such as zero) automatically. Thus when a variable is assigned a memory location by the compiler, the default value of that variable is whatever (garbage) value happens to already be in that memory location!

14. Write a program (or multiple) in a known programming language to do the following:  
a. Query for installed windows patches.

run "systeminfo" in a CMD window and it will pull back a load of statistics about your system including what patches are installed.

b. Query for installed Linux Packages

To check whether or not a currently installed package has been patched for a bug or security vulnerabiltiy, zypper can be used to query packages using --bug and --cve flags (this is the preferred method). The "rpm" command with flags "-q --changelog" will also show the patches including security patches

Run command apt list --installed to list all installed packages on Ubuntu. To display a list of packages satisfying certain criteria such as show matching apache2 packages, run apt list apache.

c. Query for installed Mac Packages

pkgutil --pkgs

15. Write a program (or multiple) in a known programming language to do the following:  
a. Query for windows system information.

systeminfo

b. Query for Linux system information

uname -a

sudo lshw

c. Query Mac system information

system\_profiler

16. What concept(s) in C++ would you use to manage resources automatically?

<https://www.modernescpp.com/index.php/c-core-guidelines-rules-to-resource-management>

a. How important are these concepts? It’s the core of modern C++

b. What tools are you familiar with for tracking resource allocations? I don’t know of any On Unix/Solaris c++/98

17. What security concerns have you come across in the past and how have you addressed them?

- Adobe used to allow pdf attachments to have attachments. This exposed a virus threat and when the threat started getting trapped then none of products I supported would download. I had to re-engineer the pdf attachments to solve the problem

18. Name some tools and/or techniques that you personally find to be the most helpful surrounding development.

I use the debugger with my development. This worked very well with 32-bit Solaris/C. Unfortunately, there was no debugger with 64-bit causing development to take 2x longer.

19. Name some tools and/or techniques that you personally find to be the most helpful surrounding code maintenance.

Pre-github – I used a development folder(s), a staged code folder, and a production folder (sounds like github?). I also made frequent backups that came in handy when the hard drive on the development machine would sometimes fail.

20. Scenario: You are dealing with legacy code containing no test suites nor documented requirements, and are required to make a change. Describe your expected process for how you may approach the solution.

I dealt mostly with legacy code with no test suites nor documented requirements and was required to maintain and add features on a continual basis. I am an expert at reading and understanding other’s code and so became an expert of the 100’s of C files that I need to maintain and made changes to.

1. Consider both a long term and short-term solutions.

Based on the timeline of when the change was due, I would either automate as much possible eliminating redundant code and modules, or make the required changes and make the automation changes during slack periods.

21. What concerns do you had supporting legacy operating systems? (If any)  
- Lack of access to github

- Lack of a debugger

- Different versions of compiler

22. Tell us about a project your worked on that you found to be interesting or unusual.

When I worked at NewsBank in 1987-1994 I did both the ETL and Front-End UI/UX for their Newspaper based CD-ROM product. This product enable the user to search thru 100’s of daily newspapers using any available keyword.

When I joined EDR in 1999 they were only searching spatially, with no keyword capability. I created a No-SQL solution and demonstrated this to the company. I was told that my idea changed the Product Development Calendar for the next five years (2015). Unfortunately, the CTO cancelled my project (and was fired shortly afterwards for unrelated transgressions). The following years I continued developing their Flagship Radius Map Report, and learned how to program in Javascript/React. I was also learning C# in 2021 when my job was eliminated.

JOB DESCRIPTION

As a software engineer at NinjaRMM you will be designing and implementing new automation against our existing native clients. You will be required to understand the existing native clients and build new applications to facilitate their control programmatically from our automation platform. C++ Software Engineers at NinjaRMM enjoy an agile team-based work environment that encourages frequent iteration rapid customer focused results. We are looking for an excellent C++ developer to work closely with our client teams to automate our applications lifecycle during acceptance criteria validation.

**Responsibilities**

* Design/code applications following specifications using appropriate tools
* Maintain and modify existing applications without supervision (as well as under direction from senior team members)
* Perform maintenance programming for existing versions
* Assume responsibility for ongoing data architecture for product
* Analyze consumer usability of the product/service
* Develop information architecture
* Design graphic user interface elements
* Conduct layout adjustments based on user feedback
* Code review
* Write project documentation and its support
* Control the architecture during subsequent iterations of the project
* Identify product requirements and limitations to ensure system function
* Hands-on work in terms of developing prototypes, contributing code, and evaluating technologies
* Monitor system(s) performance
* Evaluate new technology
* Develop and execute software test plans in order to identify software problems and their causes
* Identify and correct existing issues based on field issues reported by our support teams
* Provide solutions and workarounds for customers, including custom overrides
* Provide guidance and expertise for our support teams
* Inspect our production monitoring system for potential issues that may be on the horizon

**Skills & Qualifications**

* C/C++
* Native Win32 API (Windows XP -> Windows 10)
* Visual Studio / XCode
* SQL (Postgres, MSSQL, or similar)
* VMWare / EXSI API Programming
* Javascript (Basic to Intermediate)
* Up-to-date knowledge about state-of-the-art development tools, programming techniques, and computing equipment
* General programming skills
* Problem-solving skills
* Technical capacity
* Communication proficiency
* Time management
* Up-to-date knowledge of design software
* Software proficiency that spans multiple programs
* Organized yet creative thinker
* JSON, Javascript, Java (preferred)
* SQL-Lite integration (preferred)
* Mac OSX (Cocoa, etc..) (preferred)
* Qt (preferred)
* TestCafe (preferred)