STL

Container	Implementation	Insert	Remove	Index	Find
vector	dynamic array	O(n)	O(n)	O(1)	O(log n)
list	double link list	O(1)	O(1)	-	O(n)
map	red-black b tree	O(log n)	O(log n)	O(1)	O(log n)
hashmap	hash table	O(1)	O(1)	O(1)	O(1)

containers speed

C++ syntax notes

***static* - all the different meanings?**

1. at *file scope*: signifies “internal-linkage” i.e. not shared between translation units
2. at *function scope*: variable retains value between function calls
3. at *class scope*: signifies independence of class instance

const* and *mutable Const member function doesn’t alter the data it operates on; except the one marked as *mutable*

default *copy constructor* and *assignment operator* - when to override? Rule of 3 in C++03 / Rule of 5 in C++11 When the class needs to be copy/move assignable - that is when it needs to be *cloneable* i.e. has non-shareable data

***volatile* keyword** Depends on language and compiler. Usually marks *atomicity* for data (but not guarantees it): reads from threads are guaranteed to have latest; marks that variable can be modified “externally”

***restrict* keyword** http://stackoverflow.com/questions/776283/what-does-the-restrict-keyword-mean-in-c

Optimisation hint to limit pointer aliasing and aid caching - it means a particular data is accessed only thru that pointer thus making optimisations like storing the ptr value in a registry for subsequent access

***in place* new** Allows to explicitly specify the memory management of individual objects, i.e. their “placement” in memory.

new (expression) [(arguments)]; for example:

```
char buffer[] = new char[256];
string *str = new (buffer) string("Hello world");
```

there is no placement delete syntax (but both *new* and *delete* functions can be overrided to specify the in-place)

C++ virtual

What is a *virtual* function? A virtual function allows derived classes to replace the implementation provided by the base class.

When you refer to a derived class object using a pointer or a reference to the base class, you can call a virtual function for that object and execute the derived class’s version of the function. Virtual functions ensure that the correct function is called for an object, regardless of the expression used to make the function call

***pure* virtual function** A virtual function that is required to be implemented by a derived class.

Classes containing pure virtual methods are termed “abstract”; they cannot be instantiated directly.

virtual *destructor* - when/why? At the root of a class hierarchy to insure proper cleanup

virtual call in *assembly*

```
mov eax, dword ptr [this]
mov edx, dword ptr [eax]
mov eax, dword ptr [edx+4]
mov ecx, dword ptr [this]
call eax
```
