Allocators

Allocators are used by the Standard Template Library to handle the allocation and deallocation of elements stores in containers. All STL containers have a template argument of type **allocator<Type>**, where *Type* represents the type of the container element. For example, the vector class is declared as follows:

```
template <
    class Type,
    class Allocator = allocator<Type>
>
class vector
```

The Standard Template Library provides a default implementation for an allocator. In most cases, this default allocator should be sufficient.

Writing Your Own Allocator

The default allocator uses new and delete to allocate and deallocate memory. If you want to use a different method of memory allocation, such as using shared memory, then you must create your own allocator.
Any allocator used with STL containers must implement the following type definitions:

const_pointer	**rebind**
const_reference	**reference**
difference_type	**size_type**
pointer	**value_type**

In addition, any allocator used with STL containers must implement the following methods:

Constructor	**deallocate**
Copy constructor	**destroy**
Destructor	**max_size**
address	**operator==**

allocate	**operator!=**
construct	