

Smart Pointers

Smart Pointer automatically manages the lifetime of their resource.

- Smart Pointer
 - automatically allocate or deallocate its resource in the constructor and destructor, following the RAII-Idiom (**R**esource **A**cquisition **I**s **I**nitialization).
 - supports explicit memory management with reference counting.
 - is C++ answer to garbage collection.
 - release its resource, if the smart pointer goes out of scope.
 - is available in four different flavors.
 - needs the header `<memory>`.

Smart Pointers

Names	C++ Standard	Description
<code>auto_ptr</code>	C++98	<ul style="list-style-type: none">▪ Exclusively owns the resource.▪ <i>Moves</i> the resource silently during copying.▪ Deprecated with C++11 and removed with C++17.
<code>unique_ptr</code>	C++11	<ul style="list-style-type: none">▪ Exclusively own the resource.▪ Can only be moved.▪ Deals with non-copyable objects.
<code>shared_ptr</code>	C++11	<ul style="list-style-type: none">▪ Has a reference count on the shared resource.▪ Automatically manages the reference counter.▪ Deletes the resource immediately if the reference counter becomes 0.
<code>weak_ptr</code>	C++11	<ul style="list-style-type: none">▪ Helps breaking cyclic references.▪ Doesn't modify the reference counter.