73. Throw by value, catch by reference.

Summary

Learn to **catch** properly: Throw exceptions by value (not pointer) and catch them by reference (usually to **const**). This is the combination that meshes best with exception semantics. When rethrowing the same exception, prefer just **throw;** to **throw e;**.

Discussion

When throwing an exception, throw an object by value. Avoid throwing a pointer, because if you throw a pointer, you need to deal with memory management issues: You can't throw a pointer to a stack-allocated value because the stack will be unwound before the pointer reaches the call site. You could throw a pointer to dynamically allocated memory (if the error you're reporting isn't "out of memory" to begin with), but you've put the burden on the catch site to deallocate the memory. If you feel you really must throw a pointer, consider throwing a value-like smart pointer such as a **shared_ptr<T>** instead of a plain **T***.

Throwing by value enjoys the best of all worlds because the compiler itself takes responsibility for the intricate process of managing memory for the exception object. All you need to take care of is ensuring that you implement a non-throwing copy constructor for your exception classes (see Item 32).

Unless you are throwing a smart pointer, which already adds an indirection that preserves polymorphism, catching by reference is the only good way to go. Catching a plain value by value results in slicing at the catch site (see Item 54), which violently strips away the normally-vital polymorphic qualities of the exception object. Catching by reference preserves the polymorphism of the exception object.

When rethrowing an exception **e**, prefer writing just **throw;** instead of **throw e;** because the first form always preserves polymorphism of the rethrown object.

Examples

Example: Rethrowing a modified exception. Prefer to rethrow using **throw;**:

```
catch( MyException& e ) {                    // catch by reference to non-const
  e.AppendContext("Passed through here");    // modify
  throw;                                     // rethrow modified object
}
```

References

[Dewhurst03] §64-65 • [Meyers96] §13 • [Stroustrup00] §14.3 • [Vandevoorde03] §20