

Defaulted Functions

Chapter 1 Safe Features

in overload resolution and, if selected, results in a compilation failure; see Section 1.1. “Deleted Functions” on page 53.

Example 2: Providing just a copy constructor Now, consider a **struct** with a user-provided copy constructor:

```
struct S2
{
    S2(const S2&); // user-provided copy constructor
};
```

A user-provided copy constructor (1) suppresses the generation of the default constructor and both move operations and (2) allows implicit generation of both the copy-assignment operator and the destructor. Similarly, providing just the copy-assignment operator would allow the compiler to implicitly generate both the copy constructor and the destructor, but, in this case, it would also generate the default constructor. Note that — in either of these cases — relying on the compiler’s implicitly generated copy operation is deprecated.

Example 3: Providing just the destructor Finally, consider a **struct** with a user-provided destructor:

```
struct S3
{
    ~S3(); // user-provided destructor
};
```

A user-provided destructor suppresses the generation of move operations but still allows copy operations to be generated. Again, relying on either of these implicitly compiler-generated copy operations is deprecated.

Example 4: Providing more than one special member function When more than one special member function is declared explicitly, the *union* of their respective declaration suppressions and the *intersection* of their respective implicit generations pertain. For example, if just the default constructor and destructor are provided (S1 + S3 in Examples 1 and 3), then the declarations of both move operations are suppressed, and both copy operations are generated implicitly.

Defaulting the first declaration of a special member function explicitly

Using the `=default` syntax with the first declaration of a special member function instructs the compiler to synthesize such a function automatically without treating it as being user provided. The compiler-generated version for a special member function is required to call the corresponding special member functions on every base class in base-class-declaration order and then every data member of the encapsulating type in declaration order, regardless of any access specifiers. Note that the destructor calls will be in exactly the opposite order of the other special-member-function calls.