Request and Suppress Methods

This lesson highlights the special methods supported by the compiler.

WE'LL COVER THE FOLLOWING

Special methods

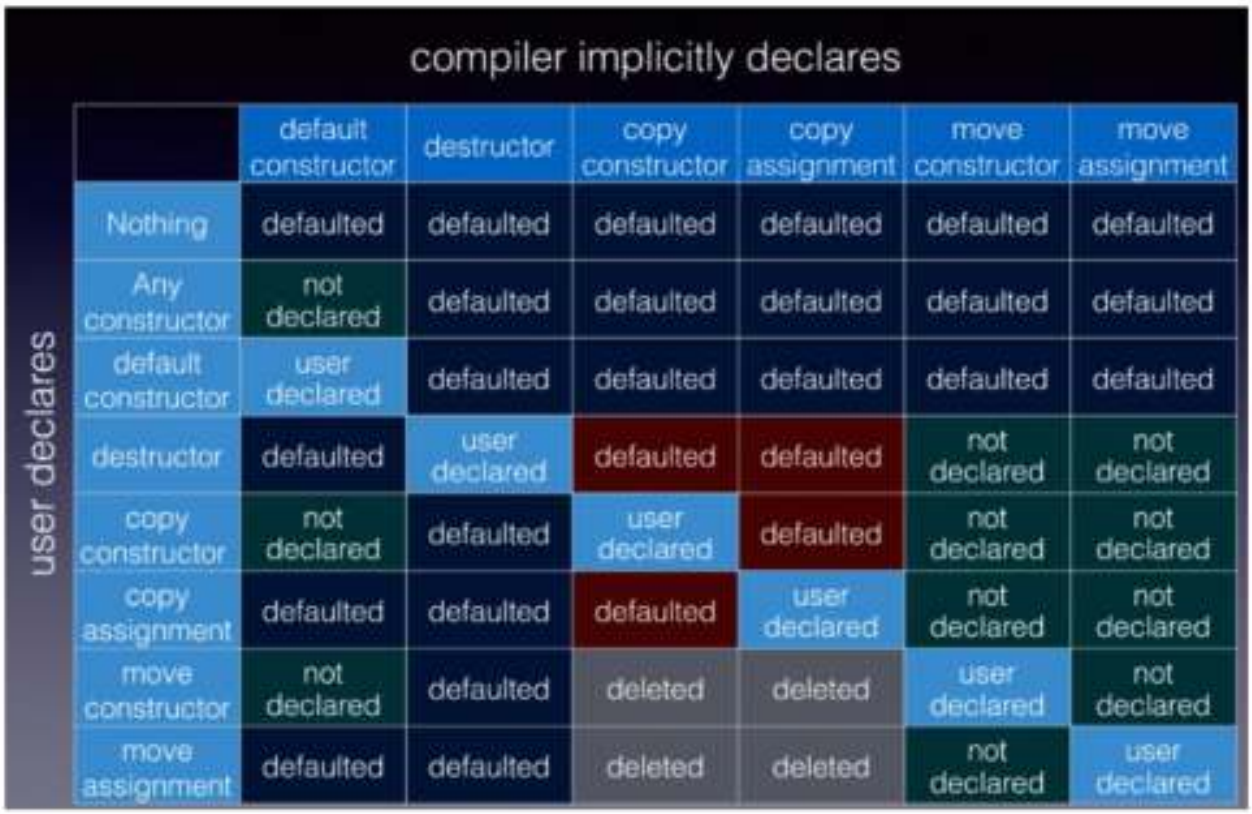
Since C++11, there has been a list of special methods that the compiler can generate implicitly if we have not defined them:

- **Default constructors** and **destructors.**
- **Copy/move** constructors and copy/move assignment operators.
- `new` and `delete` operators for objects and C arrays of objects.
- The `default` and `delete` keywords can be used to guide the creation or suppression of these special methods.
- `default` can only be assigned to special methods that do not have any default arguments. Hence, it wouldn't work with something like an ordinary class method or a parameterized constructor.

Let's suppose we have a parameterized constructor for our `Account` class but no default constructor. The compiler can easily generate it for us. All we need to do is assign `default` to the default constructor.

```
...
Account() = default;
Account (double balance){this->balance = balance;}
...
```

The behavior of the compiler varies based on what special members the user has defined. We can find details in the diagram by Howard Hinnant below:



Request methods: `default`

The compiler generates the request methods when it has the following characteristics:

- `public` access rights and are not virtual.
- The copy constructor and copy assignment operator get constant lvalue references.
- The move constructor and move assignment operator get nonconstant rvalue references.
- The methods are not declared explicit and possess no exception specifications.

Suppress methods: `delete`

- By using `delete`, we can define purely *declaratively* that an automatically generated method from the compiler is not available.

- We can simply tell the compiler **what** to do without explaining **how** to do it.
- By using `delete` in combination with `default`, we can define whether or not a class's objects:
  - can be copied.
  - can only be created on the stack.
  - can only be created on the heap.
- Apart from objects and pointers, `delete` is also applicable to functions.

We will look at concrete examples of `default` and `delete` in the next lesson.