

Structure Arguments

Structures may be **passed** as arguments to functions; they may also be **returned from** functions. Specify the structure type as the parameter or return type.

We can use this to get around the inconvenience of the missing structure aggregate operations. Although we **cannot** compare two structures with `==` or `!=`, we can write a function to supply the necessary operation, like this:

```
bool equals(Date lhs, Date rhs)
{
    return lhs.month == rhs.month &&
           lhs.day == rhs.day && lhs.year == rhs.year;
}
```

The function `equals()` takes two `Date` arguments and returns `true` if they are equal and `false`, otherwise. Use the function like this:

```
if (equals(d1, d2)) cout << "equal" << endl;
```

*The parameter names **lhs** and **rhs** are shorthand for left-hand-side and right-hand-side, and are commonly use with functions that mimic the built-in operators.*



This course content is offered under a [CC Attribution Non-Commercial](https://creativecommons.org/licenses/by-nc/4.0/) license. Content in this course can be considered under this license unless otherwise noted.