

Structured Bindings

Let's look at that last function call once again:

```
Roots r = quadratic(1, -3, -4);  
cout << "roots->" << r.root1 << ", " << r.root2 << endl;
```

Notice that it is up to the programmer to **"unpack"** the returned structure. C++17 added a new feature to the language that makes it easier to retrieve several returned values from a function. These are called **structured bindings**.

With structured bindings, you can **automatically unpack** the structure into a special form of **auto** declared variables like this:

```
auto [r1, r2] = quadratic(1, -3, -4);  
cout << "roots->" << r1 << ", " << r2 << endl;
```

Note that you do not need to specify the names of the data members in the returned structure variable, nor the types of the local variables **r1** and **r2**. The structure is unpacked and **root1** is assigned to **r1** and **root2** is assigned to **r2**. They are both **automatically** declared as type **double**.



This course content is offered under a CC Attribution Non-Commercial license. Content in this course can be considered under this license unless otherwise noted.