

Structured Binding

`std::tuple`, `struct`'s, or C-arrays can directly bound to variables.

```
std::tuple<T1, T2, T3, T4, T5> getValues(){ ... }  
auto [a, b, c, d, e] = getValues(); // types are: T1, T2, T3, T4, T5
```

```
struct Values{  
    std::string f{"test"};  
    int g{5};  
};  
Values val;  
auto [f, g] = val;
```

➡ `a` to `g` are automatically created.

Structured Binding

Copy Semantics

```
auto pair = std::make_pair(2011, 2011);  
auto [fir, sec] = pair;
```

```
sec = 2017;
```

```
std::cout << fir << ", " << sec;  
std::cout << std::get<0>(pair) << ", "  
           << std::get<1>(pair);
```

➡ 2011, 2017
 2011, **2011**

structuredBinding.cpp

Reference Semantics

```
auto pair = std::make_pair(2011, 2011);  
auto& [fir, sec] = pair;
```

```
sec = 2017;
```

```
std::cout << fir << ", " << sec;  
std::cout << std::get<0>(pair) << ", "  
           << std::get<1>(pair);
```

➡ 2011, 2017
 2011, **2017**