Subject: Re: Wonders of C++14
Posted by dolik.rce on Wed, 07 Dec 2016 08:14:50 GMT

View Forum Message <> Reply to Message

Hi,

This is possible since C++11. In C++14, it actually works for any function, not only lambdas:

```
auto st() {
  struct Foo { int a, b; };
  Foo x {10, 20};
  return x;
}
int main() {
 auto v = st();
 return v.a;
}
```

koldo wrote on Wed, 07 December 2016 08:14struct Foo is declared inside st, but after that is assigned to v and used in DUMP().
How is it possible?

The magic is in the "auto", thanks to automatic return type deduction. The details of type Foo are known to the compiler, the only problem is that it is not declared outside the function, so you can't specify it as a return type. Unless you use auto, in which case compiler uses its knowledge to fill all the necessary data. It works kind of similarly to anonymous structs:

```
struct {
  int a = 10;
  int b;
} x;

int main() {
  return x.a;
}
```

You can use the values in x, but you can't simply use it's type (e.g. to create second variable of the same type), since it has no name. The auto just exposes the "anonymous type" returned by the function for the programmer to use.

Best regards,
Honza