Why are rvalues `const`?

Andreas Weis

BMW AG

MUC++, December 19, 2017

Spot the bug!

Spot the bug!

```
struct X {
    int foo;
};

void printX(X& x) {
    std::cout << x.foo << '\n';
}

X createX() { /* ... */ }

int main()
{
    printX(createX());
}
```

Spot the bug!

```
struct X {
    int foo;
};

void printX(X const& x) {
    std::cout << x.foo << '\n';
}

X createX() { /* ... */ }

int main()
{
    printX(createX());
}
```

But in order to understand...

But in order to understand...

Operator overloading

```
int ix = 1;
int iy = 2;
int iz = ix + iy;
```

Operator overloading

```
int ix = 1;
int iy = 2;
int iz = ix + iy;

complex cx(1, 0);
complex cy(2, 0);
complex cz = cx + cy;
```

Operator overloading (by-value)

```
complex operator+(complex x, complex y)
{
    return complex(x.real() + y.real(),
                   x.imag() + y.imag());
}

complex cz = cx + cy;
```

Operator overloading (by-pointer)

```
complex operator+(complex* x, complex* y)
{
    return complex(x->real() + y->real(),
                   x->imag() + y->imag());
}

complex cz = &cx + &cy;      // ???
```

Operator overloading (by-pointer)

```
auto        foo = &cx - &cy;          // ??????
```

Operator overloading (by-pointer)

```
ptrdiff_t foo = &cx - &cy;          // !
```

Operator overloading

Operator overloading

```
complex operator+(complex& x, complex& y)
{
    return complex(x.real() + y.real(),
                   x.imag() + y.imag());
}

complex cz = cx + cy;
```

References - syntactic ambiguities

```
complex* px = &cx;
complex* py = &cy;

px  = py;
*px = *py;
```

References - syntactic ambiguities

```
complex* px = &cx;
complex* py = &cy;

px  = py;
*px = *py;

complex& rx = cx;
complex& ry = cy;

rx = ry;          // ???
```

References

References cannot be re-bound.

A `T&` behaves like a `T* const`.

The end.

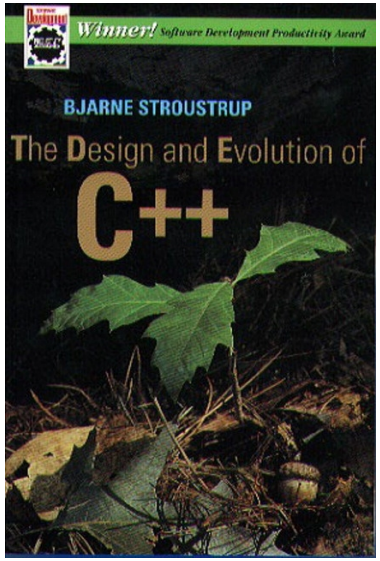
The end.

Just one more thing...

```
void incr(int& ri) { ri++; }

void g()
{
    long l = 1;
    incr(l);
}
```

If you want to know more...



- The Design and Evolution of C++ (Addison-Wesley, 1994)
- A History of C++: 1979 - 1991 (HOPL-II, 1993)
- Evolving a language in and for the real world: C++ 1991-2006 (HOPL-III, 2007)

Thanks for your attention.