C++ Training

Move Semantics

Olivier Parcollet

Center for Computational Quantum Physics (CCQ)

Flatiron Institute, Simons Foundation

New York

Temporaries

- Another kind of references…
- Mostly a library details,
  but you need to understand the basic since they appear in documentations.

```
struct A {
  A(std::vector<int> && v);
};
```

```
template<typename T>
void f(T && x);
```

- What is the meaning of && ? From user's point of view only

Rvalue reference

- A reference to a temporary

```
class A {
public:
  A(std::vector<int> const & v); // 1
  A(std::vector<int> && v);      // 2
};
```

- The class can take ownership of the data of v

```
auto v = std::vector<int>{1,2,3};

auto a = A(v);                          // 1 : copy data

auto a = A(std::vector<int>{1,2,3}); // 2 : steal data

auto a = A(std::move(v));               // 2 : steal data

// DO NOT USE v afterwards (sanitizers)
```

Forwarding reference

- Special case when using template.
- User's point of view :
  bind to anything. Reference, temporary, const, not const …

```
template<typename T>
void f(T && x);
```