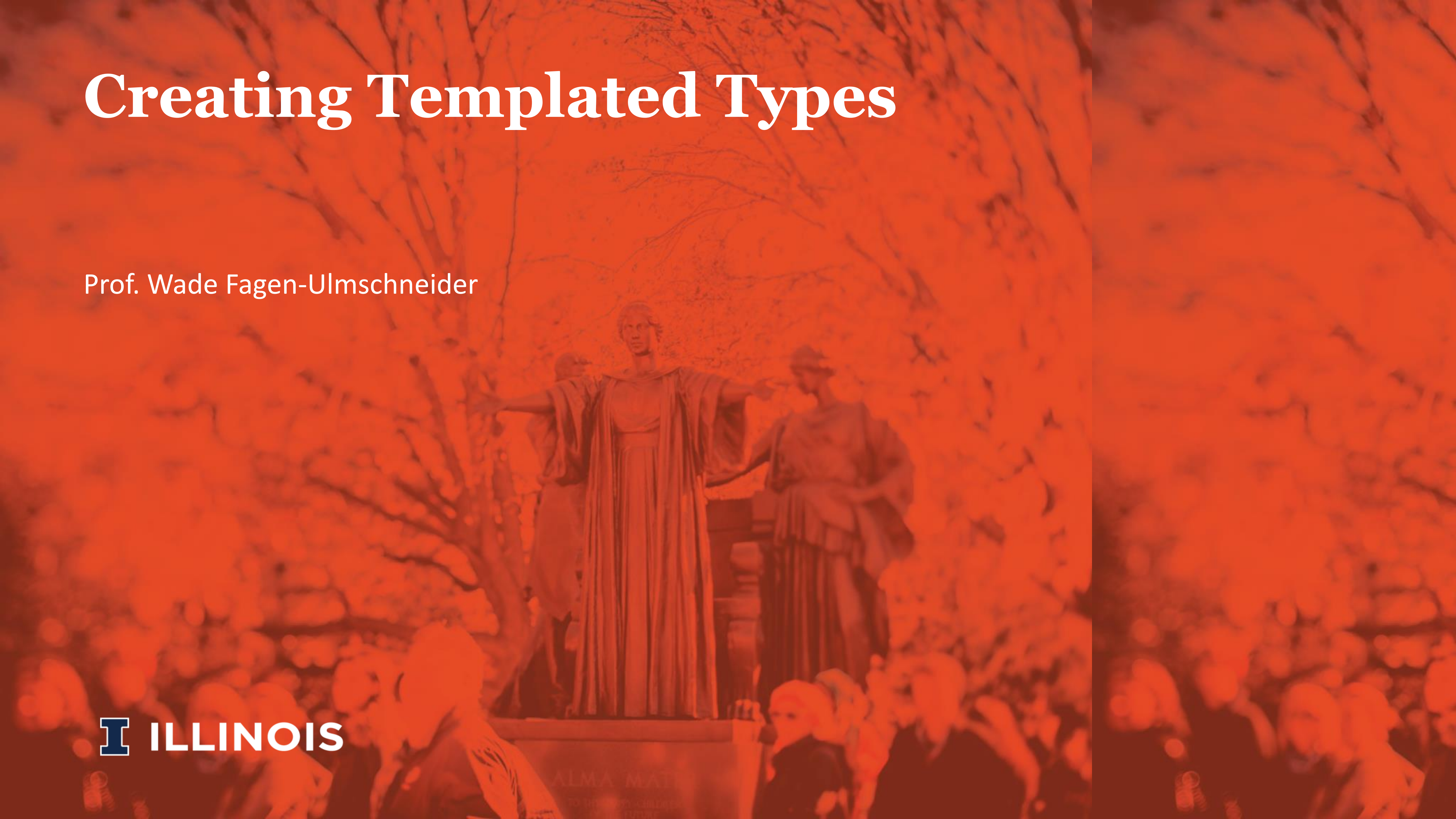
Creating Templated Types

Prof. Wade Fagen-Ulmschneider



ILLINOIS

C++ allows for us to use the power of templates in building our own classes.

Templated Functions

A template variable is defined by declaring it before the beginning of a class or function:

```
template <typename T>
class List {
   …
   private:
      T data_;
};
```

```
template <typename T>
int max(T a, T b) {
  if (a > b) { return a; }
  return b;
}
```

Compile-Time Binding

Templated variables are checked at compile time, which allows for errors to be caught before running the program:

```
template <typename T>
T max(T a, T b) {
  if (a > b) { return a; }
  return b;
}
```

```
max( 4, 7 );  // OK



max( Cube(3), Cube(6) );
// !! Error
```

cpp-templates/main.cpp

```cpp
template <typename T>
T max(T a, T b) {
  if (a > b) { return a; }
  return b;
}

int main() {
  cout << "max(3, 5): " << max(3, 5) << endl;
  cout << "max('a', 'd'): " << max('a', 'd') << endl;
  cout << "max(\"Hello\", \"World\"): " << max("Hello", "World")
                                                               << endl;
  cout << "max( Cube(3), Cube(6) )" << max( Cube(3), Cube(6) ) << endl;

  return 0;
}
```