C++'s Standard Library (std)

Prof. Wade Fagen-Ulmschneider

The **C++ standard library** (std) provides a set of commonly used functionality and data structures to build upon.

Standard Library Organization

The C++ standard library is organized into many separate sub-libraries that can be #include'd in any C++ program.

iostream

The iostream header includes operations for reading/writing to files and the console itself, including `std::cout`.

cpp-std/cout.cpp

```cpp
#include <iostream>

int main() {
  std::cout << "Hello, world!" << std::endl;
  return 0;
}
```

Standard Library Organization

All functionality used from the standard library will be part of the `std` **namespace**.

- Namespaces allow us to avoid name conflicts for commonly used names.

If a feature from a namespace is used often, it can be imported into the global space with `using`:

```
using std::cout;
```

cpp-std/cout2.cpp

```cpp
#include <iostream>

using std::cout;
using std::endl;

int main() {
  cout << "Hello, world!" << endl;
  return 0;
}
```

Adding Our Cube

cpp-std/main.cpp

```cpp
#include <iostream>
#include "Cube.h"

int main() {
  uiuc::Cube c;
  c.setLength(2.4);
  std::cout << "Volume: " << c.getVolume() << std::endl;

  double surfaceArea = c.getSurfaceArea();
  std::cout << "Surface Area: " << surfaceArea << std::endl;

  return 0;
}
```

Using the uiuc Namespace

A “cube” is rather generic – hundreds of cube-based data structures exist!

We will be specific about **our** `Cube` and specify that our `Cube` is within the `uiuc` namespace!

cpp-std/Cube.h

```cpp
#pragma once

namespace uiuc {
  class Cube {
    public:
      double getVolume();
      double getSurfaceArea();
      void setLength(double length);

    private:
      double length_;
  };
}
```

cpp-std/Cube.cpp

```cpp
#include "Cube.h"

namespace uiuc {
  double Cube::getVolume() {
    return length_ * length_ * length_;
  }

  double Cube::getSurfaceArea() {
    return 6 * length_ * length_;
  }

  void Cube::setLength(double length) {
    length_ = length;
  }
}
```