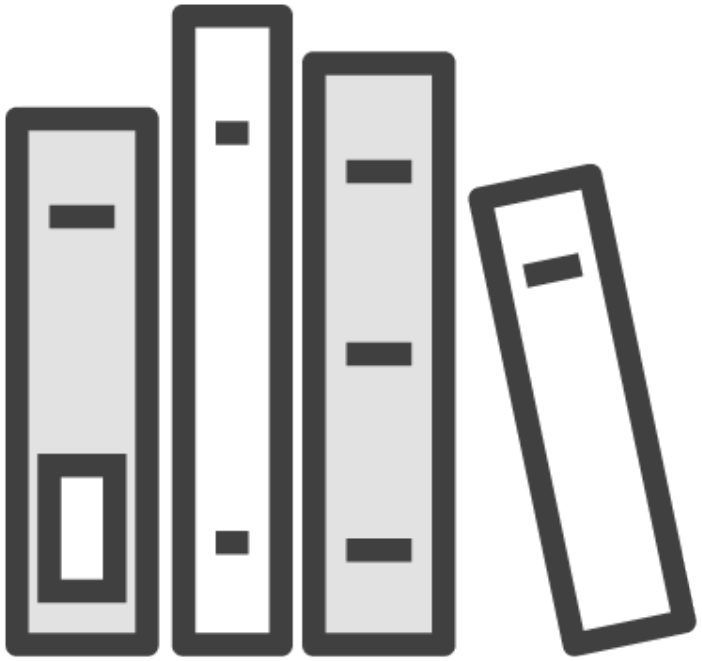
Streams and Locals

Kate Gregory

www.gregcons.com/kateblog @gregcons

Libraries



Some languages provide keywords to perform specific tasks

- Print, eval, call

C++ uses functions and classes for just about everything

- Even places that don't look like function calls

Both functions and classes can be distributed in a library

- Linked into your app along with your object files, or included into your code to be compiled

A library called The Standard Library comes with your tools

- And the tools know where to find it

Stream I/O

Input / Output

Keyboard/screen

Files

Other sources / targets that support streaming

Stream I/O

Contrast to “record based” or “fixed” I/O

- Database
- One line in the middle of a file

C++ supports stream I/O with operators

- << sends something to a stream
- >> reads something from a stream

The library can create objects to use the operators with

- cout is console output
- cin is console input

```
std::cout << "Hello!";

std::cout << "Hello!\n";

std::cout << "Hello!“ << ‘\n’;
```

- To send characters to cout, use the << operator
- For a line break, use ‘\n’
- Can send several separate things at once

Some Syntax Notes

Anything from // to end of the line is a comment even if it looks like code

Double and single quotes are different – look closely

Remember almost every line ends with a semi colon

Exercise

Write your own version of Output that prints out some words and some numbers.

- Use double quotes " not single ' around words and groups of words
- Use as many '\n' as you want
- The numbers can be calculated (2+2) or just type them (42)
- Try some non-integer numbers too

Don't ask for the impossible

- 3/0
- "hello" + 2

Errors for these won't make sense to you yet

Include

Including a file into your application

#include <iostream>

Opens up a world of libraries and capabilities

For some libraries also need to link in the library

Namespaces

```
std::cout << 2+2 << '\n' << "Hello!";
```

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

- Good libraries are in a namespace
- To use something that's in a namespace, use the full name
- Prevents conflicts if other libraries use the same names
- To save typing, can use a namespace or (usually better) each piece of it

Local Variables

Variables in C++ have a type

- string, number, date, Employee, etc
- Some types are built into the language
- Some are User Defined
  - Some “users” are actually library writers

Variables must be declared before they are used

- int limit;
- float rate;

Local Variables

Built in types are not initialized for you

- User defined might be
- Best practice: declare and initialize at once
  - int limit=100;
  - float rate=0.23;
- Can ask the compiler to deduce type when initializing
  - auto x = 7; // x is int

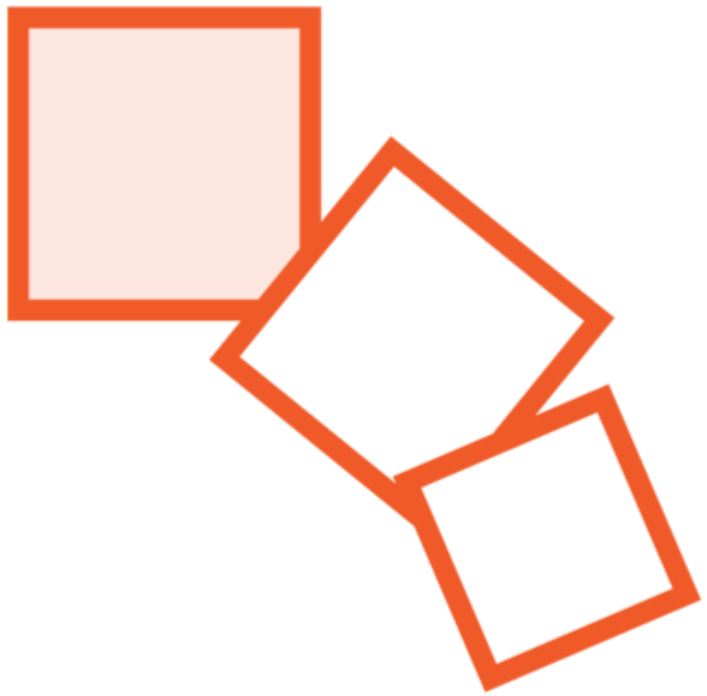
Type Safety

C++ enforces type

Variables have type

Expressions have type

Type Safety



It is ok to “promote”

- Put an integer (eg 3) into a float

You will be warned if you “demote”

- Put a floating point number (eg 4.9) into an integer

Type Safety

Some combinations are just not allowed

Put a string into an integer

Multiply a string and a float

Keyboard Input

You can store keyboard input into a local variable

```
cin >> i;
```

The library takes care of parsing

Type rules still apply

Summary

The Standard Library provides all you need to read from the keyboard or write to the screen

In C++ all variables have a type that does not change

All expressions also have a type

The compiler enforces a number of rules related to the type system

Users (you, or a library writer) can define new types