Integers

C++ comes with a wide variety of built in numeric types. There are signed and unsigned **integers** in five different sizes, as well as three different sizes of **floating-point** (or real) numbers. In addition, the standard library contains a **complex number** class, and it is easy to create your own custom numeric types.

Integers are whole numbers; the name is Latin, meaning "whole". Mathematical integers are infinite, but the C++ varieties are **finite**; each stored in a **fixed region of memory**.

The sizes for C++ integers are: `short`, `int`, `long`, and `long long`. C++ **does not** specify an **exact range** or **representation** for the integers. Both are **implementation dependent**. Here are the rules:

- Size **cannot decrease** as you move from `short` to `int` to `long` to `long long`.
- `int` must use at least `2` bytes (16 bits), `long` must use at least `4` bytes (32 bits), and `long long` must use at least `8` bytes of storage (64 bits).

On our platform, `int` is 32 bits, `long` and `long long` are both 64 bits. Other platforms have different limits. For instance, on the current version of Visual C++, `long` is 32 bits, just like the `int`.

This course content is offered under a CC Attribution Non-Commercial license. Content in this course can be considered under this license unless otherwise noted.