CS50 Data Types

Overview

Unlike many modern programming languages, C is a **statically-typed** language; it requires that every time you declare a variable, that you specify the data type of that variable. Many modern languages are **dynamically-typed**: at runtime, the program figures out the type of all the variables in the program. There are several different primitive (or basic) data types that are built in to C, and several more that are offered by the CS50 Library.

Key Terms

- statically-typed
- dynamically-typed
- native
- qualifiers

Native Types

C's **native** data types are the data types built into the programming language. An `int` is a data type which represents an integer: its value could be a positive or negative whole number, or zero. Numbers like 5, 28, -3, and 0 can be represented as `int`s, but numbers like 2.8, 5.124, and -8.6 cannot. When an `int` is declared, the computer allocates 4 bytes worth of space for it. Since 4 bytes is 32 bits, this means that there are 2^{32} (more than 4 billion) possible integers that can be represented as an `int`: in the range from -2^{31} to $(2^{31} - 1)$.

What if you need to store an integer outside of this range? C also includes **qualifiers**, which are keywords that can be added in front of type names to cause changes to the type. One such qualifier is the `unsigned` qualifier, which designates a type to be not negative. As a result, an `unsigned int`, while still 4 bytes in size, doesn't need to include the negative numbers in its range of possible values. An `unsigned int` can therefore take on a value in the range 0 to $2^{32} - 1$.

Another qualifier is the `long` qualifier, which allocates more bytes to the variable, allowing it to store more values. The long long integer (denoted by the type `long long`) is an integer which uses 8 bytes of storage instead of 4, allowing numbers in the range from -2^{63} to $(2^{63} - 1)$.

In addition to `int`s, C also has several other native data types. A `char` is a data type which represents a character of text. A `char` in C is surrounded by single quotation marks. Examples of possible `char` values include lowercase letters like 'a', uppercase letters like 'Z', symbols like the exclamation point '!', or even the newline character `'\n'`, which counts as a single character.

To store numbers that isn't a whole number, C has a type called `float` (short for floating-point), which uses 4 bytes to store a decimal value like 2.8 or 3.14. C also has a native type called `double`, which also stores decimal values but does so using 8 bytes instead of 4.

CS50 Library Types

The CS50 Library makes other types available to you, so long as you remember to type `#include <cs50.h>` at the start of your program. The `bool` type (short for Boolean) stores one of only two values: `true` or `false`.

The CS50 Library also defines a type called `string`, which stores text.

C doesn't limit users to only using the data types built into the programming language. It also offers additional features which allow the programmer to define their own custom types to use in programs.

Data Type	Native?	Sample Values	Size
int	Yes	5, 28, -3, 0	4 bytes
char	Yes	'a', 'Z', '?', '\n'	1 byte
float	Yes	3.14, 0.0, -28.56	4 bytes
double	Yes	3.14, 0.0, -28.56	8 bytes
long long	Yes	5, 28, -3, 0	8 bytes
bool	No	true, false	1 byte
string	No	"Hi", "This is CS50"	4 or 8 bytes