Variables in C++

Alfred Mikhael

August 29, 2021

1 What are variables?

Variables are what we name the data that we manipulate when programming. Variables are stored in memory, so that we can access them later. In most (if not all) languages, variables have *types*. C++ is a *statically typed* language, which means that these types are checked at compile time.

2 Primitive Types in C++

A primitive type is a variable type that is built into C++, not user defined. These types are the building blocks of all other types in C++. All types are essentially just 0s and 1s, so the main difference between them is the size that they occupy and the behaviour that programmers assume of it.

2.1 Integer types

The following are the primitive types used to store integers and their corresponding sizes.

- char, 1 byte
- short, 2 bytes
- int, 4 bytes
- long, 4 bytes
- long long, 8 bytes

2.2 Other primitive types

There are a few other primitive types that are not uesd to store integers.

- bool, stores true/false values, 1 byte
- float, stores floating point numbers, 4 bytes
- double, stores floating point numbers, 8 bytes

3 Usage of data types

The `char` type may be confusing; if it is just storing an integer, why is it treated differently?

As programmers, we make assumptions about the usage of types so that we can create appropriate behaviour for them. This is better shown by an example.

```
#include <iostream>

int main() {
    char a = 65;
    short b = 65;

    std::cout << a << std::endl;
    std::cout << b << std::endl;
    cin.get()
}
```

The output for this code will be:

```
A
65
```

This is because `std::cout` defines different behaviour for `char` types, although at the level of the bytes, both variables look the same.

4 Sizeof operator

The sizes of variable types may depend on the specific compiler that you are using. You can access this information by using the `Sizeof` operator followed by whichever type you want to check. This may be useful for less common types like `long double`.