Defining Your Own Type Names Using typedef

Let's make our own data type using the typedef functionality.

In C you can assign an alternate name to a data type, any name you want. The `typedef` statement allows you to do this.

For example, we can use typedef to define a type called “Counter” which is an alternate name for an integer, like this:

```
typedef int Counter;
```

Now we can declare variables to be of type “Counter”:

```
typedef int Counter;
Counter i, j, k;
```

Typedef isn’t used particularly often in most basic C code, but you may come across it in applications requiring a high degree of portability. New types may be defined for basic variables and typedef may be used in header files to tailor the program to the target machine.

One place you may see typedef used more often is while simplifying the declaration of compound types such as the `struct` type (which we will see later).

Next, we’ll take a look at two powerful operators used in C code.