# Input-Output Parameters

**We can use a single parameter for both input and for output. Consider** `toUpperCase()` in Java. It takes a `String` as an argument, and returns a new, uppercase version of the original.

```
String str = "cat";
str = str.toUpperCase();  // CAT
```

This **builder method** does not (indeed, in Java, cannot) change its argument. However, that is a little inefficient, especially when assigned to the same variable, as we've done here.

Since C++ strings **may be modified**, we can write a more efficient version like this:

```
void toUpperCase(string& str)
{
    for (auto& c : str) { c = toupper(c); }
}
```

Here, `str` provides **both** input and output. We call this and **input-output** parameter. Because of that, it is passed by reference, **not** `const` reference. Note also that the loop variable `c` is a reference, not a value, so we can **modify the character it refers to**. Here's how to use it:

```
string str = "cat";
toUpperCase(str);  // Not str = toUpperCase(str)
```