What's Const For

Ashwin Menon

December 15, 2024

What good are *const* variables? A good compiler would not allocate storage for them, and an enumeration achieves the same effect. According to Bjarne Stroustrupp in *The C++ Programming Language*,

> const's primary role is to specify immutability in interfaces.

Making an object *const* does not specify how the compiler allocates storage for that object; instead, it places limitations on the usage of that object. Consider the *strcpy* function:

```
1 char *strcpy(char *dest, const char *src);
```

The source is a pointer to a *const char*, which is how the designer of *strcpy* tells the compiler that *strcpy* will not modify the source string. This allows the compiler to make optimizations where possible.

Consider the *push_back* method of *vector*:

```
1 void push_back(const T& value);
```

Not only is the argument passed by reference to avoid unnecessary copying, but the reference is also a *const* which tells the compiler that *push_back* promises not to modify its argument. Accidental argument modifications within *push_back* will therefore be flagged as an error by the compiler.