During this semester, you have already seen the word “const” many times in the lecture. Now let’s see more examples and have an idea of some different kinds of const declarations.

The basic definition of “const” is stable value which can’t be changed. Remember the const pointer.

int a = 10;

const int \*p = &a;

This declaration means you can’t use the pointer to change the value which stored in address of a (&a)

So this is not allowed:

\*p = 5; // you can’t change the value which the pointer p points to.

But if you want to change the value by the variable a itself, it is allowed because you didn’t declare int a as a const integer.

a = 5; // It is allowed

Another declaration is:

int \*const p = &a;

Which means the pointer p can only points to address of a (&a). You can’t change p to point to another address.

int a = 10;

int \*const p = &a;

int h = 3;

p = &h;

p = &h is not allowed because you set p is a const pointer.

There are actually way more different const declaration. You can find some information on the links below. Take a look than you will fill more comfortable to see “const” in the future.

* <http://www.cprogramming.com/tutorial/const_correctness.html>
* <http://stackoverflow.com/questions/16449889/why-using-the-const-keyword-before-and-after-method-or-function-name>