Re: declaration vs. defintion

Source: http://coding.derkeiler.com/Archive/C_CPP/alt.comp.lang.learn.c−cpp/2004−11/0042.html

---

From: Alwyn (*dt015a1979_at_mac.com.invalid*)
Date: 11/01/04

```
Date: Mon, 01 Nov 2004 12:26:53 +0000
```

In article <2ummi2F2bqr3qU1@uni−berlin.de>, Val \
<chrisval@bigpond.com.au> wrote:
>
*> AFAIU, when we say we have *defined* something, doesn't*
> that imply that storage is associated with the particular
> identifier being defined ?,

You shouldn't think of it in terms of storage being allocated. Just think of a definition as a specification that is sufficiently complete to be used in a program.

Take for instance:

```
const int Five = 5;

inline int timesFive(int n)
{
  return n * Five;
}
```

There is no need for there to be storage associated with either Five or timesFive, yet they have been defined, and they are usable, no?

> therefore meaning that once we
> have a definition of an object, we cannot define it again ?,
> hence the ODR ?

I believe the one−definition rule goes something like this:

A translation unit must not contain more than one definition of any variable, function, class type, enumeration type, or template, and external definitions must not be redefined within a program.

A user−defined type is not an object and is not included in the above list.

Alwyn