

## Re: What this mean? Compiler output

---

*Source:* [http://coding.derkeiler.com/Archive/C\\_CPP/comp.lang.c/2006-05/msg03747.html](http://coding.derkeiler.com/Archive/C_CPP/comp.lang.c/2006-05/msg03747.html)

---

- *From:* Roberto Waltman <[usenet@xxxxxxxxxxxxxx](mailto:usenet@xxxxxxxxxxxxxx)>
  - *Date:* Wed, 24 May 2006 10:34:29 -0400
- 

"Vladimir Oka" <[novine@xxxxxxxxxxxxxx](mailto:novine@xxxxxxxxxxxxxx)> wrote:

Olaf "El Blanco" wrote:  
You should have tried to post some code as well...

19 ..\tipos.h previous declaration of 'ccc' was here

This probably means that you have multiple (same) declaration of the symbol `ccc`, and that the first one was on line 19. You should also have an error message telling you where the duplicate was found.

Olaf, you do not mention if 'ccc' is a type or variable name.  
I will assume it is a type, based on the file name you provide.

A common source for this problem is header files without multiple inclusion guards. For example, if I have the following two header files:

```
file tipos.h:  
...  
typedef short int ccc;  
...
```

```
file a.h:  
...  
#include "tipos.h"  
...
```

And then a C source file includes both, the typedef for ccc will be processed twice:

```
file c.c:  
...  
#include "tipos.h"  
#include ""a.h" /* ← ccc multiple declaration error here */  
... /* when tipos.h is processed a second */  
/* time. */
```

## Re: What this mean? Compiler output

The solution is to block the C preprocessor from using the contents of any single header file more than once, as follows:

```
file tipos.h:
#ifndef TIPOS_H_INCLUDED
#define TIPOS_H_INCLUDED
...
typedef short int ccc;
...
#endif /* TIPOS_H_INCLUDED */
```

```
file a.h:
#ifndef A_H_INCLUDED
#define A_H_INCLUDED
...
#include "tipos.h"
...
#endif /* A_H_INCLUDED */
```

```
file c.c:
...
#include "tipos.h"
#include ""a.h" /* ← ok, the contents of tipo.h are */
... /* ignored the 2nd time */
```

Occasionally you will see the same technique applied to typedefs or variables:

```
...
#ifndef BOOLEAN
#define BOOLEAN
typedef unsigned char boolean;
#define FALSE 0
#define TRUE 1
#endif /* BOOLEAN */
...
```

On the other hand, if `ccc` is a variable, you are probably defining it in a header file. Change the definition to a declaration ("extern ??? `ccc`;) and define it in a single C source file.

.