test.cxx File Reference

Macros

#define **MORE_EXAMPLES** (0)
Define as 1 for more examples.

Typedefs

typedef int **Int_Function1**(int, int)
Generates correct typedef of a function returning int.

typedef int() **Int_Function2**(int, int)
Generates **WRONG** typedef of a function returning int.

Macro Definition Documentation

◆ MORE_EXAMPLES

#define MORE_EXAMPLES (0)

Define as 1 for more examples.

Typedef Documentation

◆ Int_Function1

typedef int Int_Function1(int, int)

Generates correct typedef of a function returning int.

The code

```
typedef int Int_Function1(int, int);
```

generates correct doxygen output:

```
typedef int Int_Function1(int, int)
```

which is OK and can be compiled.

◆ Int_Function2

typedef int() Int_Function2(int, int)

Generates **WRONG** typedef of a function returning int.

The code

```
typedef int (Int_Function2)(int, int);
```

generates **WRONG** doxygen output:

```
typedef int() Int_Function2(int, int);
```

which does not compile if used literally by a user.

Note

This syntax is used in FLTK: generates incorrect doxygen output with extraneous '()'. https://github.com/fltk/fltk/blob/e44a988c4a0770b7a12193cc023d1a55b5ff89e7/FL/Enumerations.H#L1174 Real example in FLTK (https://www.fltk.org) as of Sep 8, 2023: This link may become invalid when new docs are generated. https://www.fltk.org/doc-1.4/Enumerations_8H.html#a427d30c9bd3a26e024ed6a712702e609

The code

```
typedef Fl_Color (Fl_Contrast_Function)(Fl_Color, Fl_Color, Fl_Fontsize, int);
```

generates this wrong doxygen output:

```
typedef Fl_Color() Fl_Contrast_Function(Fl_Color, Fl_Color, Fl_Fontsize, int)
```

Note 'Fl_Color()' with '()' instead of 'Fl_Color' w/o '()'.