

#pragma Directive in C/C++

The preprocessor directive `#pragma` is used to provide the additional information to the compiler in C/C++ language. This is used by the compiler to provide some special features.

Here is the syntax of `#pragma` directive in C/C++ language,

```
#pragma token_name
```

The table of some of `#pragma` directives in C/C++ language is given as follows,

Sr.No.	#pragma Directives & Description
1	#pragma startup Before the execution of <code>main()</code> , the function specified in <code>pragma</code> is needed to run.
2	#pragma exit Before the end of program, the function specified in <code>pragma</code> is needed to run.
3	#pragma warn Used to hide the warning messages.
4	#pragma GCC dependency Checks the dates of current and other file. If other file is recent, it shows a warning message.
5	#pragma GCC system_header It treats the code of current file as if it came from system header.
6	#pragma GCC poison Used to block an identifier from the program.

Here is an example of `#pragma` directive in C language,

Example

```
#include<stdio.h>
int display();

#pragma startup display
#pragma exit display

int main() {
    printf("\nI am in main function");
    return 0;
}

int display() {
    printf("\nI am in display function");
    return 0;
}
```