In the C and C++ programming languages, **#pragma once** is a non-standard but widely supported preprocessor directive designed to cause the current source file to be included only once in a single compilation to avoid the reentrant issue during compilation process. Thus, #pragma once serves the same purpose as #include guards, but with several advantages, including: less code, avoiding name clashes, and improved compile speed.

|  |  |
| --- | --- |
| **Header1.h** | |
| **Line#** | **Code** |
| 1  2  3  4 | #pragma once  /\*  The header file contents are here....  \*/ |

Since #pragma once is non-standard, it might not support by all compilers. If it is not supported, the following #include guards can be used:

|  |  |
| --- | --- |
| **Header2.h** | |
| **Line#** | **Code** |
| 1  2  3  4  5  6 | #ifndef \_\_MY\_HDR\_\_  #define \_\_MY\_HDR\_\_  /\*  The header file contents are here....  \*/  #endif |

Let’s consider the following code:

|  |  |
| --- | --- |
| **OtherHeaderFile.h** | |
| **Line#** | **Code** |
| 1  2  3  4  5  6  7  8  9 | #include "Header1.h"  #include "Header2.h"  /\*  Assume that this header file is depends on Header1.h and Header2.h  To ensure the compilation success, this header file will indirectly include them  \*/ |

|  |  |
| --- | --- |
| **EX08-05.cpp:** *Control Reentrant* | |
| **Line#** | **Code** |
| 1  2  3  4  5  6  7  8  9  10 | #include "Header1.h"  #include "Header2.h"  #include "OtherHeaderFile.h"  /\*Up to this point, The contents of Header1.h and Header2.h will being included once only!\*/  int main()  {  std::cout << "Hello World!\n";  } |