June, 6th 2013

1. WinCE: optimized for devices that have minimal storage, are often configured without disk storage, and may be configured as a “closed” system that does not allow for end-user extension. A distinctive feature is that large parts of it are offered in source code form.
2. .NET Framework: library+language interoperability. Programs written for the .NET Framwork execute in a software environment, known as CLR(Common Language Runtime), an application virtual machine that provides services such as security, memory management, and exception handling. Class library+CLR=.NET Framework.
3. 在类外部另外一个文件中写构造函数有参数的时候，使用：

类名：：构造函数名称(参数名以及类型)

：参数名(初始化参数值),

参数名(),

Etc….

参数名()

{如果不能直接赋值的内容可以写在这里大括号括起来，最后一个大括号之前的函数不用逗号了}

1. Creating and Using DLL:

<http://msdn.microsoft.com/en-us/library/ms235636(v=vs.90).aspx>

For Example：

File Something.h

Namespace X1

{

Class X2

{

Public:

Static \_declspec(dllexport) F1…

}

}

File Something.cpp

#include Something.h

Namespace X1

{

Functions….

Double X2::F1…

}

File FileWhichWillUseDLL.cpp

#include Something.h

When you are using the functions in the DLL, write like this:

X1::X2::F1….

1. Catch (…) the ellipsis (…) means that the handler will catch any exception no matter what the type of the throw exception is. This can be used as a default handler that catches all exceptions not caught by other handlers if it is specified at last.
2. \_declspec: specifies that an instance of a given type is to be stored with a Microsoft-specific storage-class attribute listed below, such as dllexport, dllimport, etc.
3. extern "C" \_\_declspec( dllimport )

extern means the entity is visible outside its translation unit(C/CPP file). A corresponding symbol will be placed in the object file, and it will hence also be visible if this object file is made part of a static library.

\_\_declspec( dllexport ) means the symbol should be exported to a DLL. It is used when compiling the code that goes into the DLL.

\_\_declspec( dllimport ) means the symbol will be imported from a DLL. It is used when compiling the code that uses the DLL.

Since the same header file is usually used both when compiling the DLL itself as well as the client code that will use the DLL, it is customary to define a macro that resolves to export when compiling the DLL and import when compiling its client:

#if COMPILING\_THE\_DLL

#define DLLEXTERN \_\_declspec(dllexport)

#else

#define DLLEXTERN \_\_declspec(dllimport)

#endif

1. delete: calls the destructor of the given argument and returns memory allocated by new back to the heap.