***Investigate other libraries in Windows programming: header / .lib / .dll What are the differences ? Basic example for each one.***

In Windows programming, libraries can come in different forms: **header files, static libraries (.lib), and dynamic-link libraries (.dll).** Here we have examples for each:

**1. Header Files (.h)**

Description:

* Header files typically contain declarations of functions, classes, constants, and macros. They are not compiled on their own but are included in other source files to provide necessary declarations.

// func\_opt3.h:

#ifndef FUNC\_OPT3\_H

#define FUNC\_OPT3\_H

void sayHello();

#endif

//main.cpp :

#include <iostream>

#include "func\_opt3.h"

void sayHello()

{std::cout << "Hello from header!" << std::endl;}

int main()

{

sayHello();

return 0;

}

**2. Static Libraries (.lib)**

Description:

* A static library is a collection of object files that are linked into the executable at compile time. Once linked, the library becomes part of the executable, which increases its size but improves performance as there are no runtime dependencies.

Create a static library:

// static\_opt3.cpp

#include <iostream>

void sayHello()

{

std::cout << "Hello from static library!" << std::endl;

}

Use the static library:

// main.cpp:

#include <iostream>

#include "my\_functions.h"

extern "C" void sayHello(); // Declaration of the function from the static library

int main() {

sayHello();

return 0;

}

### 3. Dynamic-Link Libraries (.dll)

**Description:**

* A DLL is a dynamic library that is loaded at runtime. This allows for smaller executable sizes and the ability to share code among multiple applications. However, it introduces runtime dependencies, meaning the application must be able to locate the DLL when it runs.

Create a DLL:

// dll\_opt3.cpp

#include <iostream>

#include <windows.h>

extern "C" \_\_declspec(dllexport) void sayHello()

{

std::cout << "Hello from DLL!" << std::endl;

}

// main.cpp:

#include <iostream>

#include <windows.h>

typedef void (\*HelloFunc)();

int main() {

HINSTANCE hinstLib = LoadLibrary(TEXT("my\_dll.dll"));

if (hinstLib != NULL) {

HelloFunc sayHello = (HelloFunc)GetProcAddress(hinstLib, "sayHello");

if (sayHello != NULL) {

sayHello(); // Call the function from the DLL

}

FreeLibrary(hinstLib);

} else {

std::cout << "Failed to load DLL!" << std::endl;

}

return 0;

}

### Summary

* **Header Files (.h)**:
  + Contain declarations, no compiled code.
  + Must be included in source files.
* **Static Libraries (.lib)**:
  + Contain compiled code that is linked at compile time.
  + No runtime dependencies.
  + Increases the size of the executable.
* **Dynamic-Link Libraries (.dll)**:
  + Contain compiled code that is linked at runtime.
  + Smaller executables, can share libraries across applications.
  + Must be available at runtime.