***Lab 1 OOP***- Ex 5 -

Onofrei Alexandru Gabriel , CEN 2.2B

**Header Files (.h)**: Used to declare functions and classes. They do not contain actual code but serve as interfaces between different files.

**Static Libraries (.lib)**: A collection of object files linked into the program at compile time. The program contains the library's code after linking.

**Dynamic Libraries (.dll)**: Code is stored in separate files that are linked at runtime, allowing multiple programs to share the same library without recompiling.

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

// 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)**

// 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)

// 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;

}