Exercise 5e - Library

Beskrivning

In previously exercise we went through how we could split functions to different files. We will now instead hide the implementation of the add and the multiply function.

To be able to let other use our *add* and *multiply* functions without sharing the source code we can create a library. This can be done in both windows, Linux Mac etc. Here we will create a library for windows (DLL) using codeblocks IDE.

We start by creating a “Dynamic Link Library” in codeblocks and name it “dll-test” (File->New->Dynamic Link Library)

We will now add the 2 functions below the example function named *SomeFunction* to the *main.cpp* file.

```
int DLL_EXPORT add(int a, int b)
{
   return a+b;
}

int DLL_EXPORT multiply(int a, int b)
{
   return a*b;
}
```

You will notice that we added an extra “DLL_EXPORT” prefix before the functions. This is all we need to do to create the library. Now we add out 2 function prototype to the main.h file after the example function *SomeFunction*.

```
int DLL_EXPORT add(int a, int b);
int DLL_EXPORT multiply(int a, int b);
```

We can now compile it for both Release and Debug. The created dll files will then appear in the projects bin folder, and you will have one small dll in the release folder and one bigger variant of it in the debug folder. The reason for this is that the debug variant contains debug symbol.

Now we have created our dll, now we continue and use it.

We start by creating a console application project in codeblocks as we have done before in this course, we can name if *console*.

Then we add a new header file to our project named *main.h*.

Then we copy the complete content of the *main.h* file from out dll project we created before and paste it into the console *main.h* file.

Now we replace the content in our main.cpp file with the following:

```
#include <iostream>
```

Information

Publicerades den den 13 januari 2022 av Psajd, alexander

HEMUPPGIFT

Den här uppgiften är inte obligatorisk

```
#include "main.h"


int main()
{
   std::cout << add(12,12) << "  " << multiply(12,12);
   return 0;
}
```

Select *Build options…* on the project and add the dll under the *Linker settings* tab.

The best way is to use release dll on release project and debug dll on debug target.

We are now done and can compile out project.

When we try to run the project, it will complain about that it misses the DLL file. To solve this, we can select *properties* on out console project and then on *Build targets* tab we can change the *Execution working dir:* to point the DLL project folder where the DLL are located.

The best way is to use release dll on release project and debug dll on debug target.

Exercise 5e

- Do the process described above
- Create a new *subtract* function that have 2 int as argument and return the difference of the 2 arguments as an integer
- Modify the code to also support this function in the DLL
- Also modify the main.cpp file to use it too