Exercise 5d - Multiple files

Beskrivning

The source code for most programs is very big and separated into different folders and files. We shall now go through how to split a program into multiple files.

When we used one file, we had a constructor at the beginning of the file to help the compiler to find the functions. When we split it to multiple files, we add those- to the header file instead.

We here have a function named adder that takes 2 arguments and return the result of the 2 arguments. We place this function in the file named *add.cpp*

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

We create a new header file and name it *add.h* with the following content

```
#ifndef ADD_H_INCLUDED
#define ADD_H_INCLUDED
int add(int a, int b);
#endif // ADD_H_INCLUDED
```

The reason why we have #ifdef and #define is to make sure we only include this file once. In this case this will not appear since we will only include the file from the main.cpp file but to be sure it is always recommended to have a #ifdef and a #define.

The main.cpp file then needs to include the "*main.h*" file

```
#include <iostream>
#include "add.h"
int main()
{
   std::cout << add(1,2);
   return 0;
}
```

If we compile the files from an IDE (Codeblocs for example) we can now compile the project as we have done before.

If we want to compile it from the command prompt we need to compile both files. For example if we use G++ (from Gcc) we compile it as:

```
G++ main.cpp add.cpp -o adder.exe
```

Information

Publicerades den den 13 januari 2022 av Psajd, alexander

HEMUPPGIFT

Den här uppgiften är inte obligatorisk

Exercise 5d

- Create the files as described in this exercise
- Add a new source file named *multiply.cpp*
- Add a new header file named *multiply.h*
- Create a function named *multiply* that have 2 integer arguments and return the product of the 2 arguments
- Use the multiply function in the *main.cpp* file and add header file