

CST8219 – C++ Programming

Lab 1

Introduction:

The goal of this lab is to ensure that you can install Visual Studio, create a CMake project, and compile it properly.

Reference:

<https://medium.com/breaktheloop/why-using-namespace-std-is-used-after-including-iostream-dc5ae45db652>

<https://www.learncpp.com/cpp-tutorial/2-9-naming-collisions-and-an-introduction-to-namespaces/>

<https://www.softwaretestinghelp.com/preprocessor-directives-in-cpp/>

Steps:

1. Use the project you started with by following the Powerpoint slides for week 1. Modify it to look like this code:

```
1 // Week 1.cpp : Defines the entry point for the application.
2 //
3
4 #include "Week 1.h"
5
6 #define usingNamespaces 0
7
8 using namespace std;
9
10 int main()
11 {
12     cout << "Hello world!" << endl;
13     return 0;
14 }
15
```

2. Use the compiler directives `#if`, `#elif`, `#endif` to create two different versions of the `main()` function. They should check the value of the compiler variable `usingNamespaces` to decide with lines of the file are compiled or not.

One version of the `main()` function should have the **using namespace std;** command:

```
using namespace std;
int main()
{
    cout << "Hello world!" << endl;
    return 0;
}
```

The other version of the main() function should not have the **using namespace std;** command:

```
int main()
{
    std::cout << "Hello world!" << std::endl;
    return 0;
}
```

3. The challenge of this lab is to correctly place the #if, #elif, and #endif commands so that only one of these two versions gets compiled based on what the variable usingNamespaces is set to.
4. Once that is working, add a #pragma message() command to each main() function so that a message shows up in the compiler output window. The message strings are either:

```
#pragma message("Using namespaces")
```

Or:

```
#pragma message("Not using namespaces");
```

Only one of these messages should appear when you compile your project.

5. The final part of this lab is to set the required version of CMake to be **3.2** and change the name of the executable file that you are creating to be your name followed by _Lab1. For example, the professor's executable file should be Eric_Lab1:

```
add_executable (Eric_Lab1 "Week 1.cpp" "Week 1.h")
```

6. On Brightspace, submit the **Week1.cpp** file, and **CMakeLists.txt** under the Week1 assignment
7. Don't forget to complete Quiz 1 on Brightspace.

Marks:	(total of 6)
The version of CMake is 3.2	+1
The executable file name has your name in it.	+1
You have put #if, #elif, #endif in the right locations	+3
Your pragma messages are in the right locations	+1