

# MCIS-4130

## Homework Assignment 1

### Purpose:

This assignment will aid the student in exploring the basics of C++ program construction and execution. Basic programs will be built and run, using the student's selected compiler.

This assignment is due for the first class..

### What to hand in:

For the programming exercise, hand in a listing of the program (which must be formatted in a reasonable style), and sample run(s) of the program. Please format code listings and output in a fixed-width font (such as Courier) for ease of reading and review.

### Problems:

#### A. A Compiler To Know And Love:

1. Ensure you have a compiler available, and know its vendor and version.  
Note that any modern C++ compiler is acceptable for the course. If you need a list, consult <http://www.du.edu/~mschwart/cxxcompilers.html> for some information and recommendations.
2. Enter the following simple program into the file `first.cpp`, replace your own name for the "Put Your Name Here" tokens, and replace the correct month, day, and year values, compile it, and run it. Capture the results for output.

```
// MCIS 4130
// Author: Put Your Name Here
#include <iostream>
#include <string>
using namespace std;

void line() {
    cout << "-----"
         << "-----"
         << endl;
}

void printdate() {
    int month = 1;
    int day = 1;
    int year = 2001;
    cout << "Produced on " <<
         month << "/" << day << "/" << year << endl;
}

int main()
{
    string myName("Put Your Name Here");
    cout << "A simple C++ program!" << endl;
    line();
    cout << myName << endl;
    line();
    printdate();
    return 0;
}
```

**Notes:**

In this first problem, the main goal is to ensure you have access to a compiler and understand the basics of compiling and running programs with it. If you get stuck on this, be sure to call the instructor! If you get through this exercise quickly, you may enter additional programs from Chapters 1-4 of the book and see if you can get them to compile as well (don't hand these in, please). Additionally, the ambitious can experiment with their debugger to set breakpoints and examine variables in the program, and step through execution.

The program itself comprises two include statements, a using statement, and three functions.

The include statements refer to interfaces used in the program. The iostream interfaces include something named cout, an object which knows how to print output. The double-less-than operator (<<) tells it what you want to output. The string interfaces are used for the myName variable. The string is one of the most frequently used C++ classes.

The using statement lets the programmer reference all the interfaces in the standard C++ library (called std) without mentioning what library they are in.

The first two functions, line and printdate, each print a message. Note that in the line function, the long double-quoted string constant was simply divided into two parts for ease of formatting.

The third function, main, is special. This is the starting point for all C++ programs. The instructions for this program start with main. When main encounters a function (line, printdate, or operator <<), it executes that function in the indicated sequence. The line and printdate functions are invoked *because* the main function called them. Note that the line function was called more than once.

This code will not be used in future exercises.

Because this exercise is due the first night, and some class participants add the course at the last minute, if the exercise is not handed in, the student's grade will be based solely on the other scores; for other students, this exercise will count if it results in raising the student's overall average.

**Evaluation Criteria:**

Code will be evaluated both for proper style and for correct content.

20	Program was entered into requested file properly
25	Substitution was made for name
25	Substitution was made for date
30	Evidence that program was compiled and run, and output produced