EE273 - Engineering Design for Software Development 2

Week 1 & 2 - Semester 1

Learning Outcomes:

- Familiarisation with the compilation process
- Use text-based compiler
- Read compiler output
- Familiarisation with the Integrated Development Environment (IDE) - Visual Studio
- Create Project from existing source codes.
- Construct a directory structure for the duration of the course
- Start a formal logbook-based record

Preface

- The following exercises do not involve writing or developing any C or C++ code but rather are intended to give students experience in using the IDE on the EEE PCs and allow familiarisation with the compilation process.
- All students are required to come to the lab with a hard-backed A4 logbook in which they will record lab-work as it progresses. Students are expected to have their logbooks with them at all lab sessions and maintain the logbook as an active record of their work within the module.
- The lab exercises in this sheet are required to be completed by the end of week 2 laboratory sessions. Additionally, students are able to work on the lab exercises as self-study in interval within the week 1 and week 2 laboratory sessions - either in EEE or University Labs when rooms are available OR using tools on their own personal devices. Create a directory structure (on your H drive) to hold your code throughout this course. You should create 1 folder per lab that contains 1 folder per lab exercise. Your directory structure should look like:

- EE272
    - Lab01
        - Ex1
        - Ex2
        - .
        - .
          .
- Lab02
    - Ex1
    - Ex2
    - .
    - .
- .Lab X
    - .
    - .
    - .
    - .
- Lab 12

Note: while files should be stored/retained on the “H: Drive” it may be necessary to run projects locally from the C drive. In this case, you must remember to regularly copy across (back-up) your code files to your H: Drive. A local backup to a USB storage device would also be a prudent thing to do.

Part A

Exercise 1

Use your favourite text editor to create a simple program that writes the message “Hello World” to the console. (See end of sheet).

1) Compile the program with GCC and run it in the console
2) Compile the program with Visual Studio and run it both from the console and from within the IDE – any difference to the output? What changes do you need to make?
3) Write down the directories and name of files that you see. Can you identify and explain what each of them are?

Exercise 2

Download the “Exercise 2” source code from MyPlace.

4) Compile the program with GCC and run it in the console
5) Create a project, compile and run the program with Visual Studio and run it both from the console and from within the IDE
6) Correct any mistakes that may exist.
7) Write down the directories and name of files that you see. Can you identify and explain what each of them are?

Exercise 3

Download the “Exercise 3” source code from MyPlace. Note: the project is now split into multiple files.

8) Compile the program with GCC and run it in the console. Hint: to create object files use the –c flag on g++ compiler. On the linking step you need to pass the object files.
9) Create a project in Visual Studio and then compile and run the program; run it both from the console and from within the IDE
10) Write down the directories and name of files that you see. Can you identify what each of them are?

Exercise 4

Download the “Exercise 4” source code from MyPlace and create a project in Visual Studio. Correct any mistakes that may exist. Describe the program operation.

Exercise 5

Download the “Exercise 5” source code from MyPlace and create a project in Visual Studio. Correct any mistakes that may exist.

- Describe the difference between this code snippet and that of Exercise 4.

Exercise 6

Combine and modify the source code from Exercises 1 and 5 to ask the user for their name and greet them by name. For example, if the user is called Jane Grey it should greet them with “Hello Jane Grey”

Part B

The code for the last two parts is found on myplace within the part b directory.

Exercise 7

Download the “Ex 1” source code from MyPlace (Lab0b directory). Note: the project is split into multiple files.

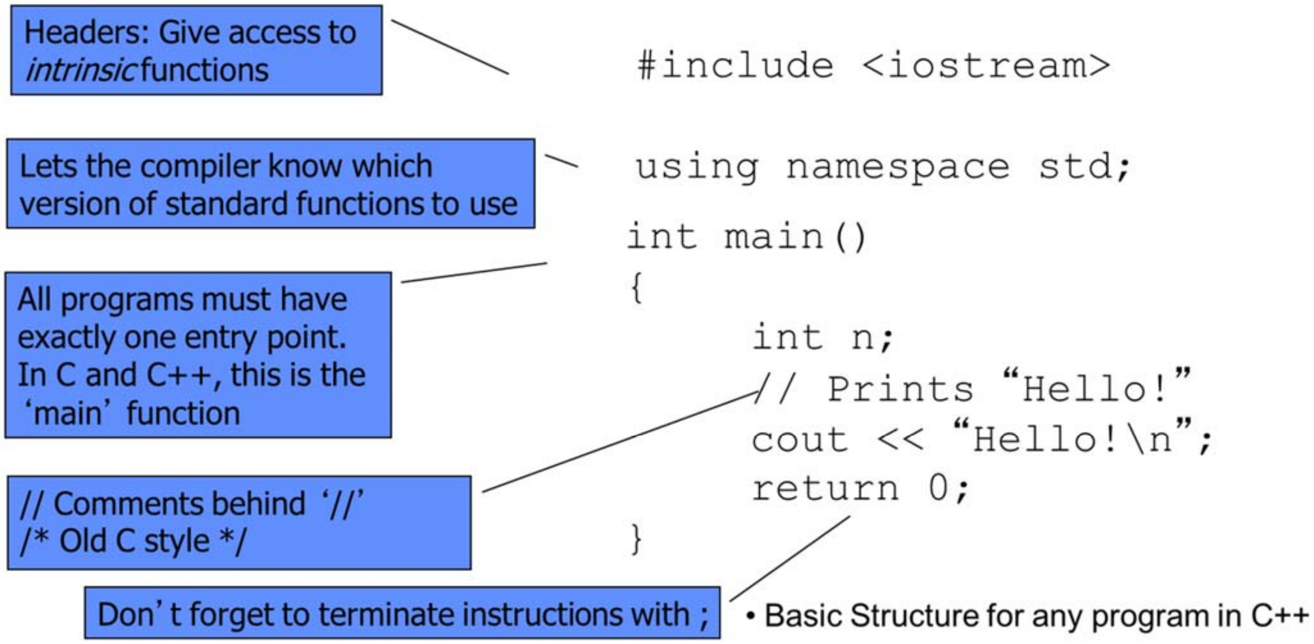
11) Compile the program with GCC and run it in the console. Hint: to create object files use the –c flag on g++ compiler. On the linking step, you need to pass the object files.
12) Run the program from the CygWin Prompt
13) Create a project, compile and run the program with Visual Studio and run it both from the console and from within the IDE
14) Write down the directories and name of files that you see. Can you identify what each of them are?
15) Run the program from the Windows Command Prompt.

Exercise 8

Download the “Ex 2” source code from MyPlace (Lab0b directory). Compile and run the program using compiler of your choice.

- Describe what does the program do?
- Take notes on your logbook from an example run. Comments?
- Input the following values 32500 and 5000. What is the output? Can you explain the output?
- Fix the code to do the computation correctly for the above inputs?

Code for exercise #1



```
#include <iostream>

using namespace std;
int main()
{
        int n;
        // Prints “Hello!”
        cout << “Hello!\n”;
        return 0;
}
```

Headers: Give access to *intrinsic* functions

Lets the compiler know which version of standard functions to use

All programs must have exactly one entry point. In C and C++, this is the ‘main’ function

// Comments behind ‘//’
/* Old C style */

Don’t forget to terminate instructions with ;

- Basic Structure for any program in C++