

# EE208 Laboratory Session 1

Bryan Hennelly

30th January 2018

## 1 Lab objectives

In this lab you will use a simple online compiler and a basic text editor like notepad to create your cpp files. You will then work through a number of introductory level C++ programming problems. The aim of these problems is to reinforce your understanding of the concepts discussed in the lectures and in the C++ tutorial. You have approximately one week to submit all of your solutions. **The due date is Monday 5th Feb 5pm.** This lab is worth 2% of your overall grade. Don't forget to comment your code for full marks. The demonstrators are here to help so don't be afraid to shout out if you're having difficulties.

### Learning Outcomes

By the end of this lab sheet you should be able to:

1. Write, compile, and execute basic C programs using a simple online compiler.
2. Write simple C++ programs that make use of (i) primitive data types, (ii) the facilities of the iostream library, (iii) branching and looping control structures, and (iv) functions.

## 2 Compiling using a simple online compiler

For the first few labs we will keep life really simple for you; you will do all of your work on an online compiler. The reason for this is that most people find using Integrated Development Environments (IDEs) like Microsoft Visual Studio, or Eclipse, a little bit of a headache at the start, and it tends to put novice programmers off what should be fun. So, for the first few weeks you can use this really simple online compiler. By doing this you are effectively connecting to a server somewhere in the world that is running the most up-to-date GCC compiler. The GNU Compiler Collection (GCC) is a compiler system produced by the GNU Project supporting various programming languages. GCC is a key component of the GNU toolchain. The Free Software Foundation (FSF) distributes GCC under the GNU General Public License (GNU GPL). So in summary this is a free compiler that you could download and run on your own machine if you wanted. I should note that using an online compiler like this does have some disadvantages; mainly, you can't use a debugger to analyse each line of your code to look for bugs, or add breakpoints. The other problem is

that you can link two files together. We'll do lots of that later once we move onto a proper IDE. Regardless, the online compiler is simple to use and is the perfect place to start and is a really useful study companion when going through your notes.

Once you have written your program and you're happy with it you can copy and paste your code to a simple text editor like Notepad++. A text editor is used to edit plain text files. You should then save your file with the \*.cpp extension and store it somewhere on your own shared drive (Z directory). Once you save as cpp you will have syntax highlighting; syntax highlighting is a very useful feature. It means that the editor will highlight certain words or types or syntax specific to a language. For example, if you have C++ highlighting turned on, the editor might make all C++ control flow keywords appear green. This makes it much easier to follow the flow of your program. As another example, the editor might have all quoted text show up as light blue. This way, if you forget to include an opening or closing quotation mark, you will quickly realize it because of the color of the text on your screen. A text editor might also indicate mismatched parentheses or brackets by turning them red; if you have a closing brace with no corresponding opening one, the color will tell you that you made a syntax error somewhere. Once you have done all of the exercises you should upload them to moodle directly - **DO NOT ZIP**.

### 3 Beginning C++ Programming

For each of the problems given below write a C++ program that provides a solution. Each box provides a filename to use (or in certain cases multiple filenames). Please ensure that you use those filenames when uploading your solutions to moodle.

**Remember:**

- **Comment your code.**
- **Use proper indentation for function and control structure.**

**Exercise 1.1:** Write a program that inputs three integers from the keyboard, and prints the sum, average, product, smallest, and largest of these numbers. You should save the source in a file called `exercise_1.1.cpp`.

**Exercise 1.2:** Write a program that reads in two integers and determines and prints if the first is a multiple of the second. Hint: you should use the modulus operator `%`. If you're not sure how it works you can have a quick look at this useful resource. You should save the source in a file called `exercise_1.2.cpp`.

**Exercise 1.3:** Write a program that inputs a five-digit number, separates the number into its individual digits and prints the digits separated from one another by three spaces each. Hint: you could make use of the modulus operator again, or you could try casting like we learned in the most recent lecture. You should save the source in a file called `exercise_1_3.cpp`.

**Exercise 1.4:** Develop a C++ program that will determine if a bank customer has exceeded the overdraft limit on their bank account when they try to make a withdrawal request. For the customer, the following information is available:

1. account number (an integer);
2. current balance;
3. maximum overdraft amount;

You should set arbitrary values for the above variables for a single user (its a very small bank). When a customer tries to withdraw money they need to enter their account number and the requested amount. If the account number matches the only one on record, the balance should be changed accordingly and a courteous message printed on screen. The account number and new balance should also be printed on the screen. If the requested amount will make the balance exceed the overdraft, print a message on screen telling them so. You should save the source in a file called `exercise_1_4.cpp`.

**Exercise 1.5: Difficult!** An integer is said to be *prime* if it is divisible only by the two distinct factors 1 and itself. Write a program that determines if a number is prime. Now extend the program to print all the prime numbers between 1 and 5000. Hint: this one will require you to use a while loop. You should save the source in a file called `exercise_1_5.cpp`.