

# ELEC 278: Fundamentals of Information Structures

## Lab 1: Getting Started with Data Structures

Fall 2023-Instructors: Ni & Mertin

September 5, 2023

## 1 Lab Overview

### 1.1 Goals for Labs

The labs in ELEC 278 are designed to

- Give learners an opportunity to explore the data structures discussed in class in more detail.
- See how the data structures are used to solve different kinds of problems.
- Give learners some examples of coding problems that may be appropriate to base quiz and exam questions on.
- Provide an opportunity to practice and improve the learner's programming and coding skills, including debugging.

### 1.2 Schedule

There are 6 labs in this course, each demanding you to implement programs using the C programming language in response to specific lab instructions. Within the allocated lab sessions, you are tasked with both coding and presenting the results to the graduate TAs. These TAs will evaluate the results in alignment with the prescribed marking criteria. You are expected to successfully complete a minimum of 5 labs. Notably, if you completes all 6 labs and receives evaluations from the TAs, only the highest 5 grades of 6 labs will contribute to the final grade. The lab portion of this course will be worth 20% of the final grade, with each lab contributing 4% towards the total. Each lab will have its own marking scheme. The marking scheme for a lab exercise will be provided with each set of Lab Instructions. Lab marks are recorded in OnQ based on the lab's total mark. OnQ does the conversion and will show you your total lab marks accumulated throughout the semester.

Some lab questions are complex, it is normal that you cannot complete the lab in your lab section (within 2 hours). Usually, the lab questions will be posted several days before lab sections, so you could start to work on the lab early. If you cannot complete the lab in your scheduled lab section, you can attend other's sections if you are available. For example, you are in section 005, if you cannot complete the lab in the scheduled time of section 005, you can attend the section 010 and ask the TAs to mark your results. However, please pay attention that the deadline of your lab is day 7 of your scheduled lab day. No accommodation will be provided. Please make sure that your results will be marked by the TA before your deadline. For example, the deadline of lab 1 for section 007 (September 14 (Thursday) 9:30AM - 11:30AM) is September 21, 11:30AM. For the lab 6, the deadline for all students is the last day of the lab, that is, December 5 (Tuesday) 2:30PM. The submission of lab after the deadline will be rejected.

Here is the schedule of lab 1:

- Section 005: September 7 (Thursday) 9:30AM - 11:30AM; Deadline: September 14, 11:30AM.
- Section 006: September 6 (Wednesday) 11:30AM - 1:30PM; Deadline: September 13, 1:30PM.

- Section 007: September 14 (Thursday) 9:30AM - 11:30AM; Deadline: September 21, 11:30AM.
- Section 008: September 13 (Wednesday) 11:30AM - 1:30PM; Deadline: September 20, 1:30PM.
- Section 009: September 5 (Tuesday) 12:30PM - 2:30PM; Deadline: September 12, 2:30PM.
- Section 0010: September 12 (Tuesday) 12:30PM - 2:30PM. Deadline: September 19, 2:30PM.

## 2 Lab 1: Getting Started with Data Structures and Algorithms

### 2.1 Lab 1 Objectives

The objectives of this lab are to setup the development environments and explore some of the basic C programming concepts that will be useful for all the other labs in this course.

### 2.2 Task 1: Development Environments

There are several freely available development environments which can be used to edit and test C programs. You may use any you wish, but we recommend either CLion or Visual Studio Code (VS Code). CLion is available from the JetBrains website; you will need to sign up for the free for education pack from them using your Queen's student email. VS Code is available from AppsAnywhere or from its website. You may also need to install a C build system; the following sections give some instructions on this depending on your operating system type.

#### 2.2.1 Windows

1. **(If you are using VS Code).** Install CMake; you most likely want the “Windows x64 Installer”. Download and run the installer and follow the on-screen prompts. Make sure to select that you want it available on the PATH when it asks (either for just you or for all users, it does not matter). (CMake is also available on AppsAnywhere, but I ran into issues trying to use that version).
2. Download `mingw-13.1.0.zip` from OnQ and extract it somewhere on your computer.
3. **(If you are using VS Code).** Install the “CMake Tools” extension (the extension ID is `ms-vscode.cmake-tools`) from the “Extensions” tab on the left (you can also get here by pressing `Ctrl+Shift+X`).

Once this is installed, press the “Reload” button if prompted. Then, go to `File → Preferences → Settings (Ctrl+,)`, select the “User” tab, and then `Extensions → CMake Tools`. Under “Cmake: Additional Compiler Search Dirs”, press “Add Item”, then paste the location you extracted the ZIP file to, followed by `\mingw64\bin`; for example, on my computer, this reads:

```
C:\Users\nmert\Downloads\mingw64\bin
```

**(If you are using CLion).** Open CLion and go to `Settings (Ctrl+Alt+S)`, then navigate to `Build, Execution, Deployment → Toolchains`. On the right, press the dropdown for “Toolset” and choose “Custom MinGW installation”. Then, paste the location you extracted the ZIP file to, followed by `\mingw64`; for example, on my computer, this reads:

```
C:\Users\nmert\Downloads\mingw64
```

#### 2.2.2 macOS

Disclaimer: I was not able to confirm these instructions myself as I do not have a Mac. Please follow the tutorial to configure CLion on macOS (<https://www.jetbrains.com/help/clion/quick-tutorial-on-configuring-clion-on-macos.html>). I will be at the first lab session to help sort out any potential issues or confusion.

1. **(If you are using VS Code).** Install CMake; choose the correct `.dmg` image based on the version of macOS you have. Download and open the file and drag the CMake app to your **Applications** folder

to install it. Then, open the app, and select Tools → Install For Command Line Use. Press the button “Install Command Line Links”.

You may need to run CMake with administrative privileges to accomplish this. You can do this by opening Terminal and running the command:

```
sudo /Applications/CMake.app/Contents/bin/cmake-gui
```

2. By following the instructions in the “Installing the command line tools” section on this page, make sure you have Apple’s developer tools installed. You should be able to run the following command and see a version number:

```
gcc --version
```

### 2.2.3 Linux

You need to install CMake and a C toolchain for your system. If you are using Linux, please follow the steps to setup CLion through snap or tarball (<https://www.linuxfordevices.com/tutorials/linux/install-clion-on-linux>).

### 2.2.4 Instructions

Once you have a development environment installed, download and extract the file `lab1.zip` from OnQ and open the extracted folder `lab1` in VS Code. When prompted, choose “yes” to configure the CMake project automatically.

Press the triangular “play” button in the bottom bar to run and build the program. It should print `Hello, World!` in the terminal. If there are any errors, you may (particularly on Windows) need to make sure that a valid “kit” is selected; this is done by pressing the “change active kit” button in the bottom bar, left of the “Build” button. On Windows, you need to select the kit with the path that you entered earlier; if you don’t see it, press “Scan for kits” and then try again. Feel free to call a TA over for help.

## 2.3 Task 2: Array Operations

Once you are able to successfully run the code, please modify the file `main.c` to calculate the average value and find the maximum element in an array:

- In `main()`, define an array and hard code the elements of an array (you can set the elements by yourself), print the average value and the maximum element of the array.
- In `array_average()`, calculate and return the average value of the input array.
- Define a new function, which finds and returns the maximum element of the input array.
- Present the results to the TA.

## 3 Marking Criteria

**After completing task 1 and task 2, call the graduate TA to mark the lab.** Lab 1 has 4 marks in total:

- Do you have a development environment set up so you can build and run C code? (1 mark)
- Do you correctly write the `array_average()` that calculates and returns the average value of the input array? (1 mark)
- Do you correctly create a new function that finds and returns the maximum value of the input array? (1 mark)
- Do you correctly write `main()` that hard code an array and prints the average value and the maximum element of the array? (1 mark)