

- Course
- ⏪

🔖

🏠 ELEC-A7100

📖 Course materials

📊 Your points

👥 Microsoft Teams

🔒 Code Vault

This course has already ended.

Doing the exercises

Check the above section to set up your development environment first if you haven’t already.

The exercises involve editing some source files in the *src* folders in the git repository. The descriptions for the exercises can be found here on A+.

Testing your code

VS Code

The exercises contain *launch.json* and *tasks.json* files for compiling and running your code through VS Code. In order to use them, you either have to open the *ModuleX.code-workspace* which will open all the exercises in that module or open a specific exercise folder with VSCode.

From the WSL terminal you can open a workspace by navigating to the folder of a module and then running the command `code Module_X.code-workspace`, where X is the number of the module.

Once one of these is open, you can open the debug tab on the left hand side of the VSCode window and select a configuration from the top of the panel.

- **F5 (Start Debugging)** will run the selected debug configuration. In this case it means compiling and then running the program inside a debugger. \
- **Ctrl+Shift+B (Run Build Task)** allows you to build (compile the sources) and clean (delete some files that were compiled previously) the exercises by picking from a list of different tasks. **This does not run the program, it only compiles/cleans it.** As F5 already does the building for you, you do not usually need to run build tasks explicitly.

This page contains some instructions on how to use the C/C++ debugger in VS Code.

Command line and make

Alternatively to using VS Code, you can compile and run the exercises from the command line. For this you will need to install GNU make tool (if not already installed). You can google “GNU make”, “how to install GNU make” and “how to run make”. For installing on windows, you might want to take a look at [this stackoverflow question](#).

In the exercise directory (e.g. `M01/intro`), you can type:

- `make run-main` – to build and run the code using main.cpp provided in the src directory

The basic make targets you might need are the following:

- Inside the exercise folders
 - **run-main** is equivalent to **run** inside the *src* folder
 - **clean** is equivalent to **clean** inside the *src* folder
- Inside *exercise_name/src* folders
 - **run** to compile and run your program with the main function
 - **clean** to delete old object files and files related to running the program
 - **valgrind-run** to compile and run your program with valgrind

Submitting answers

Go to the submission box in A+, load your submission and submit.

Interpreting the submission results



(The image is a bit outdated, but contains the main points)

The above shows an example output of running a test. The boxes with a ^ character are collapsible/expandable. Each test creates a colored box with the test name and points in the header, as you can see in 7-10. The numbered parts are as follows:

6. Compilation output. Shows compilation and linking warnings and errors.

7. Run normally contains nothing relevant. Standard I/O shows the output that wasn’t used by the tests. You can use this e.g. for debugging.

8. Green box means that the test gave full points.
9. Red box means that the test gave zero points.
10. Orange box means that the test gave partial points.
11. Red highlighting shows where the answers differ.