Course
ELEC-A7151
Course materials
Your points
Code
Code Vault
Course Pages
MyCourses
Teams Channel

This course has already ended.
The latest instance of the course can be found at: Object oriented programming with C++: 2023 Autumn

« 1 Development environment Course materials 3 Common problems »
ELEC-A7151 / Getting Started / 2 Exercises

2 Exercises¶

Contents

After setting up your development environment, you can now learn how to start doing the exercises.

In this course, the exercises are defined as a part of each module and their requirements are described inside their own frames in the A+ system. The exercises require you to edit/write some source files provided to you, and then submit your development to A+ system. In the following you can find detailed instructions and descriptions related with the exercises.

2.1 Downloading exercise templates¶

The exercise templates are distributed as `zip` files, and are required to be extracted inside the folder you have created in the Configuring Visual Studio Code section.

When you open a module page, you will see a header section containing a link to the module `zip` file. The header section looks like this.

- **You can download the template for the programming tasks of the module as a `zip` file from `this link`.**

You can download the templates using your browser and File Manager of your operating system, or using command line tools.

Important

- It is not recommended to use browser download if you have selected and prepared for `Remote connection` option above.
- It is not recommended to use command line option if you have Windows computer and you do not have WSL.

Command Line

1. Create a new folder with name `<module-name>`.

   ```
   mkdir <module-name>
   cd <module-name>
   ```

2. You can use `wget` tool to download the template `zip` file.

   ```
   wget --no-check-certificate <download url>
   ```

   `<download url>` can be copied by right clicking the download link and selecting `copy link`.

1. After downloading the template, you need to unzip its content using `unzip` tool.

   ```
   unzip <module-name>.zip
   ```

2. Find the folder that contains `<module-name>.code-workspace`. It should have a name `Module-<X>` where `X` is the module number.

   Hint

   You can use *list* command `ls` to see the content of the current directory.

3. Change to that folder, and then start Visual Studio Code.

   ```
   cd Module-<X>
   code <module-name>.code-workspace
   ```

Graphical User Interface

1. Download the `zip` file.
2. Extract its content into the course folder you have created.
3. Start Visual Studio Code.
4. In Visual Studio Code window, click `File - Open Workspace...`.
5. Navigate to the module template folder you just extracted, and then into the folder that is just extracted.
6. Select `<module-name>.code-workspace` file.

2.2 Doing the exercise tasks¶

After opening the module template in Visual Studio Code, which you have installed in Installing Visual Studio Code and tools and configured in Configuring Visual Studio Code, now it is time to start developing the code for the exercise task.

1. The template should have all the necessary files to allow you start writing your code. The content of the workspace can be seen on `Explorer` windows of Visual Studio Code. Explorer window can be opened by any of the following methods:
   1. Press `CTRL + ⇧ + p` (`(⇧ + ⌘ + p)` in MacOS) and write `explorer` and select `Focus on folders view`
   2. Press `CTRL + ⇧ + e` (`(⇧ + ⌘ + e)` in MacOS)
   3. Look for `Explorer` on the (Left) side bar.
2. Write your code in the provided files.
3. Once you feel that you can start testing your code, you can build the workspace using Visual Studio Code by clicking `Terminal - Run Task...` and selecting the pre-configured `Run Tasks`. The following tasks are provided.
   - **Build**: builds the source file `main.c` and creates executable `main.out`.
   - **Clean**: deletes the executable file `main.out`.
   - **Valgrind**: runs `valgrind` with `main.out`.

   `Build` **and** `Clean` **tasks can be used for building and cleaning the project.**

   `Valgrind` **task can be used for identifying/validating dynamic memory related problems.**

   Supported platforms

   Valgrind is only available in Linux platforms. Thus, this task can only be used in
   - WSL under Windows 10
   - Linux computers
   - Remote connection to Aalto servers
4. You can also single step debug your code using Visual Studio Code Debugger interface.

   You can access the debugger window by any one of the following methods:
   1. Press `CTRL + ⇧ + p` (`(⇧ + ⌘ + p)` in MacOS) and write `debug` and select `Debug: Start Debugging`
   2. Press `CTRL + ⇧ + d` (`(⇧ + ⌘ + d)` in MacOS)
   3. Look for `Run and Debug` on the (Left) side bar.

   The launch configuration is used for running compiled (Build) executable. The provided workspace has a launch configuration pre-configured so that you can use the debugger right away. In order to ease required effort, the launch configuration has `Build` task as prerequisite, and runs it if it has not been done.

   Hint
   - You can use `CTRL + ⇧ + b` to `Run Build Task` which shows the configured tasks.
   - You can use `F5` to `Start Debugging`.

   Caution

   The debugger does not stop execution unless you put a `breakpoint`. Before you start debugging, place a `breakpoint` by pressing `F9`.

   This page contains some instructions on how to use the C/C++ debugger in Visual Studio Code, and the following video shows how to debug your code using Visual Studio Code.

Command line and `make`

Alternative to Visual Studio 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`.

In the exercise directory (e.g. `Module_1/first_touch/src`), you can type:

```
make all
```

This command will build and run the code using `main.cpp` provided in the exercise directory. The provided make targets are the following:

- `all` builds `main.out` and runs `main.out`
- `main` build the exercise and creates executable `main.out`.
- `clean` removes the generated executable file `main.out` and intermediate object files
- `valgrind-run` to build and run executable with Valgrind

2.3 Submitting your solution¶

You can submit your submission directly to the A+ system, where it will be evaluated and graded. For all exercises, the submission box looks like as shown below.

Submission Box

- You can see that, in the header area your points for the exercise are shown in `points/<exercise points>` format where `<exercise points>` is the maximum number of points you can get for the exercise.
- On the right of the points, you can see your submission history and number of submissions.
- You can also see the deadline for the exercise.
- The file that must be submitted is shown just above the `Choose File` button.

When you are convinced that you can submit your solution to the system for grading, follow the instructions below.

1. You need to access to your code in order to load it to A+ system.
   - In Visual Studio Code, in the `Explorer` window, find your file.
   - Right click it.
   - Click `Download`, and select a suitable directory to save it.
2. Click `Choose File` to select your file.
3. Press `Submit` button.

2.4 Interpreting the submission results¶

After submitting to A+ system, you need to wait grader to complete its operation, which might take while. Once your submission is evaluate, you will see a window that has a content similar to the image shown below*.

./_images/grader-output.png

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. These sections does not provide any useful information.
8. Full point tests†.
9. Zero point tests‡, and the associated test cases are expanded.
10. Partial point tests§, and the associated test cases are expanded..
11. Red highlighting shows where the answers differ.

*
The image is a bit outdated, but contains the main points.

†
Green box means that the test gave full points.

‡
Red box means that the test gave zero points.

§
Orange box means that the test gave partial points.

2.5 Getting help¶

If you encounter a problem, and you are sure that you cannot solve it alone, you can ask for help in Exercise Channel of Course Teams.

Warning

After receiving your task, programming *usually* is a personal effort. Therefore, we strongly recommend you to try to identify the source of the problem alone, and research the possible solutions. However, there are occasions that an experienced third eye can identify the problems and help you in understanding what you have missed.

The purpose of Exercise Channel of Course Teams is to guide you through the problems when you get stuck.

Its purpose is not to provide you the solution!

In case you need to share a code snippet, please use Code Vault for this purpose. The procedure as follows:

1. Find and click `Code Vault` on the Left Menu of the course A+ page.
2. Press `New` on the `Home` tab.
3. Give a **Title** to the problem. For example, *stdout does not print my class*.
4. Copy the content you want to share to the **Content**.
5. Press `Save` button.
6. Copy the address of the page from your browser's address bar.
7. Paste it to Exercise Channel of Course Teams, after explaining your help request.

Important

Before you ask for help, make sure that you have submitted your code to the A+ system. After submitting your code, you should include the following information in your request:

1. **Module:** <Name of the module of the exercise>
2. **Programming Task:** <Name of the programming task>. It is the first line (in bold) of the exercise description. You could also mention <Section>.<Subsection> numbers.
3. **Problem Description:** A brief description of the problem you need help with.
4. **Code Vault link (Optional):** A link to a code snippet that can be used for describing your problem.

Hint

You can access the Exercise Channel of Course Teams by clicking `Teams Channel` on the Left Menu of the course A+ page.