User Manual

Programming: Principles and Practices using C++

The graphics/GUI support library

Qt Account

First things first: create a Qt account at https://login.qt.io/register.

Register yourself as an open-source user.

Installer download

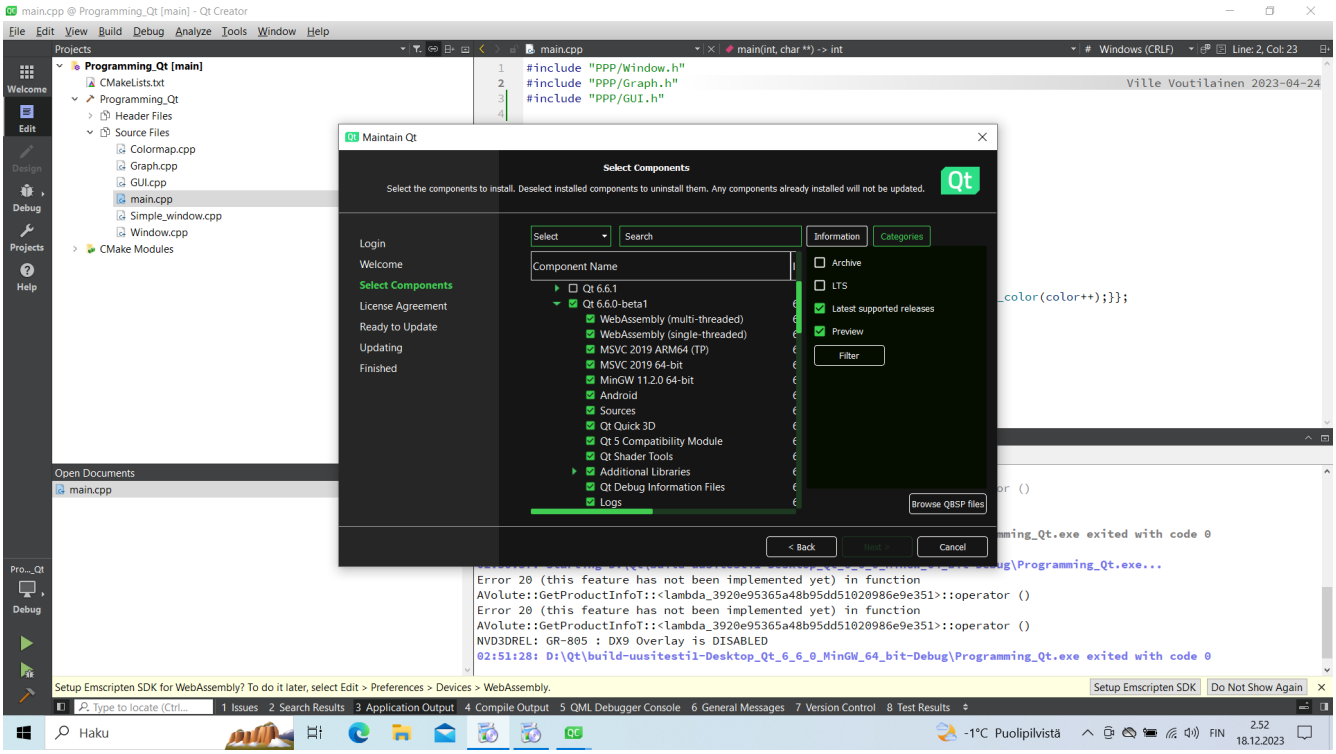
Get the installer from https://www.qt.io/download-qt-installer-oss?utm_referrer=https%3A%2F%2Fwww.qt.io%2Fdownload-open-source and launch it after downloading.

Installation

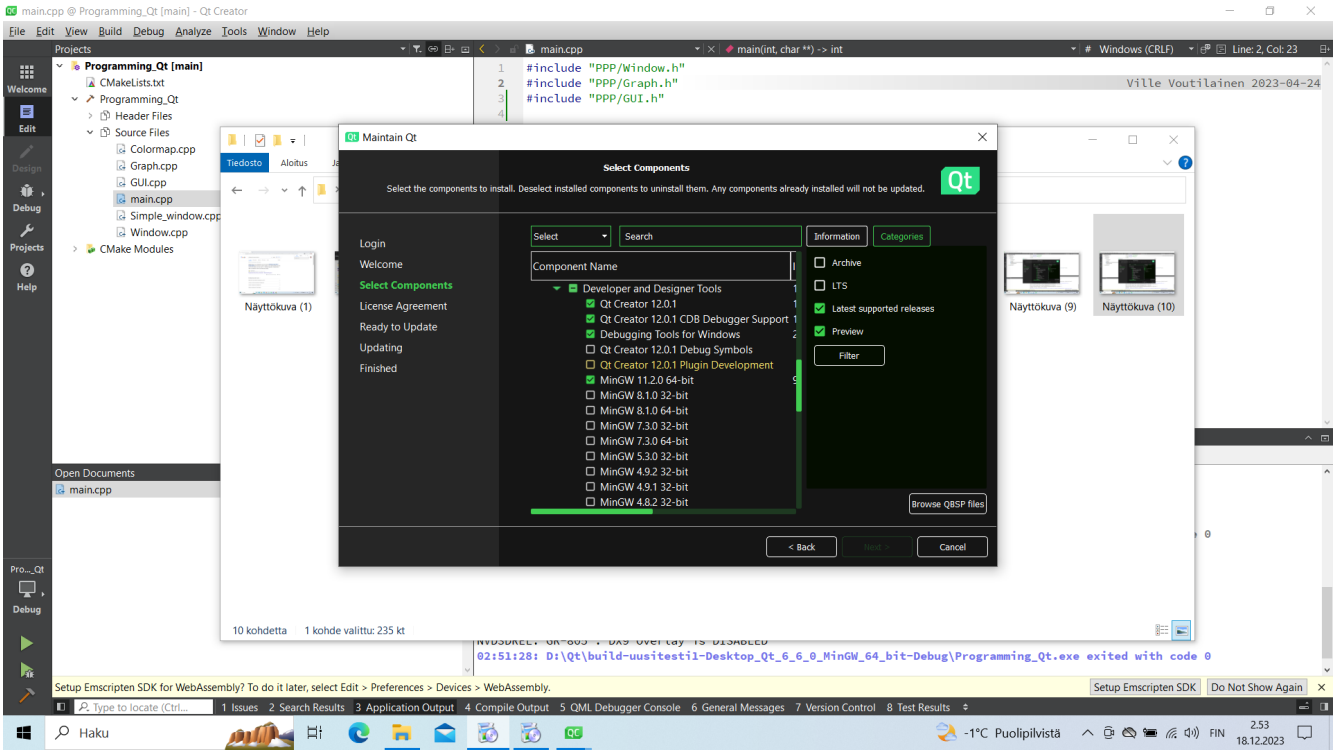
The default installation will work on Linux and Mac OS.

For Linux, you need to separately install GCC's C++ compiler, which is available in a package manager under the name "g++".

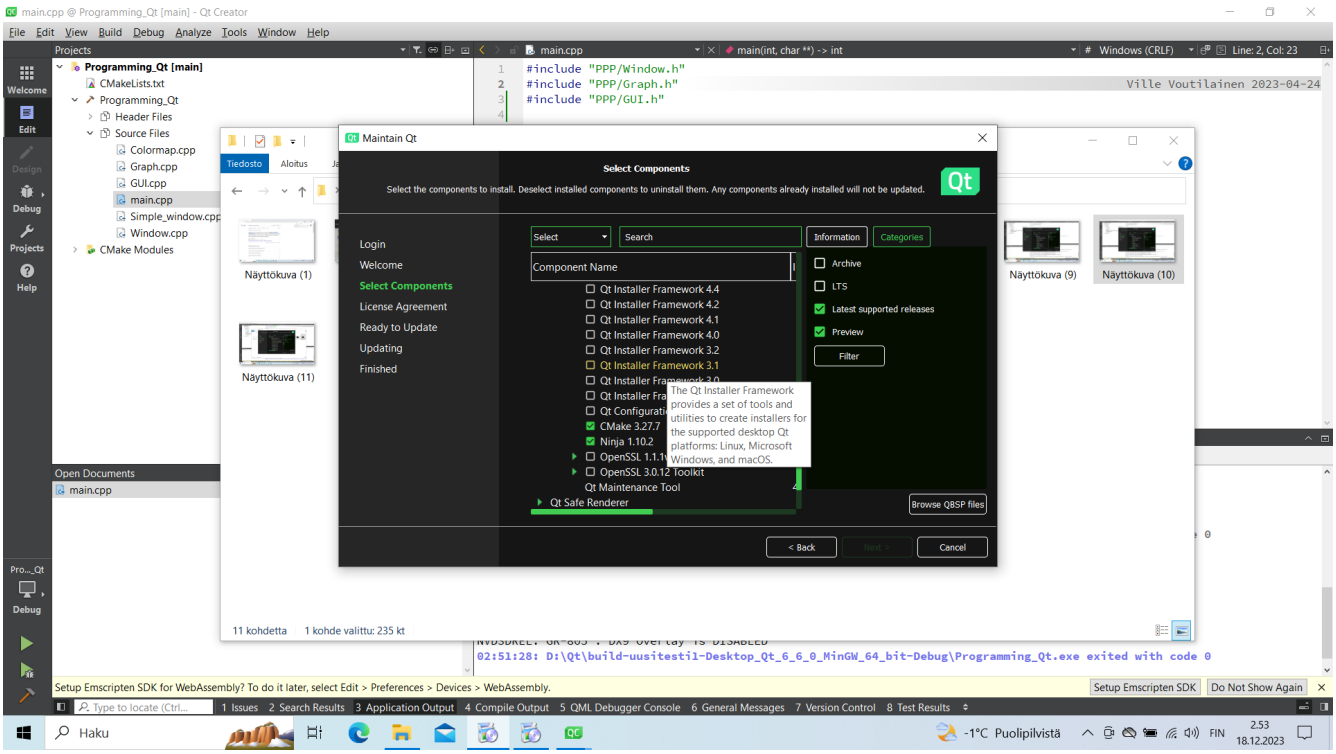
For Windows, if you want to use Qt Creator, choose MinGW as your target, after selecting a custom installation, under the Qt version to be installed:



In that screenshot, I have both MinGW and Visual Studio selected. We'll get to the details of a Visual Studio installation later, but for use with Qt Creator, you can just select MinGW. In addition to selecting a MinGW version of Qt, also install MinGW itself:



In general, it's good to verify that you install both CMake and Ninja, which are the build tools used for building applications using Qt:

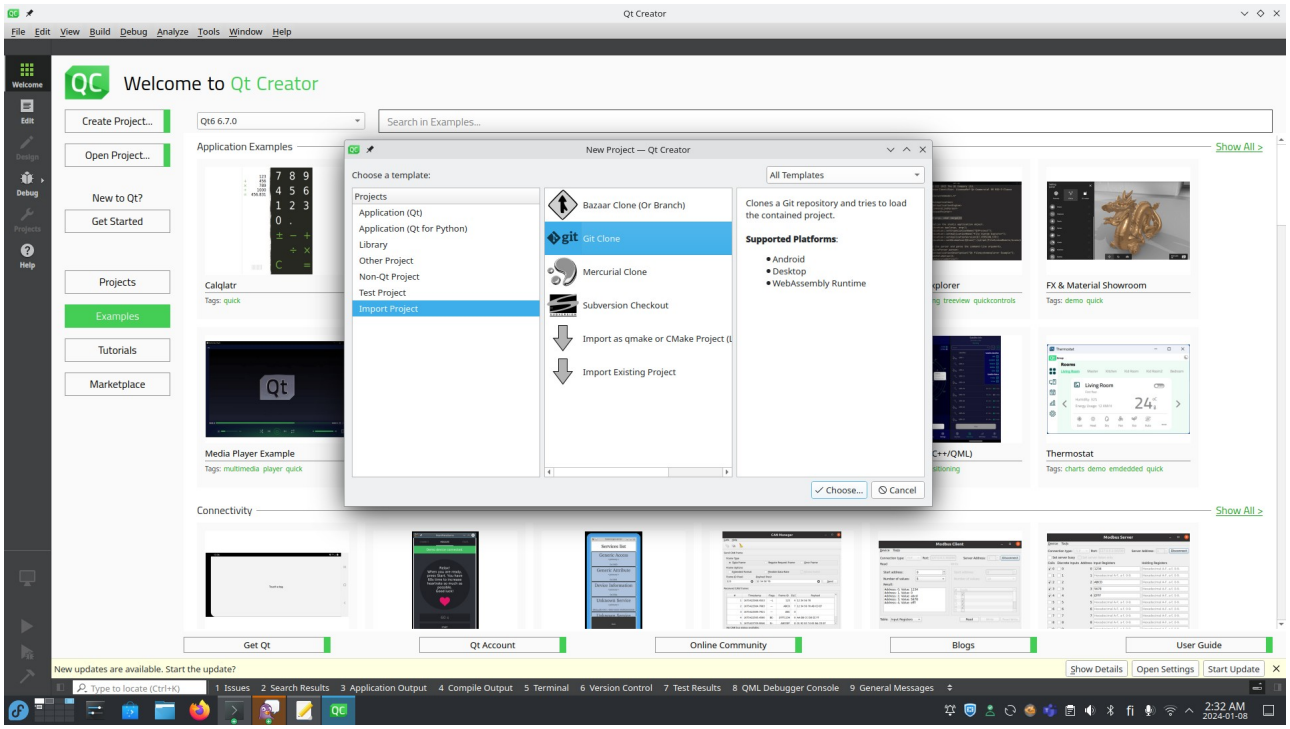


Use the search box to find these components if need be.

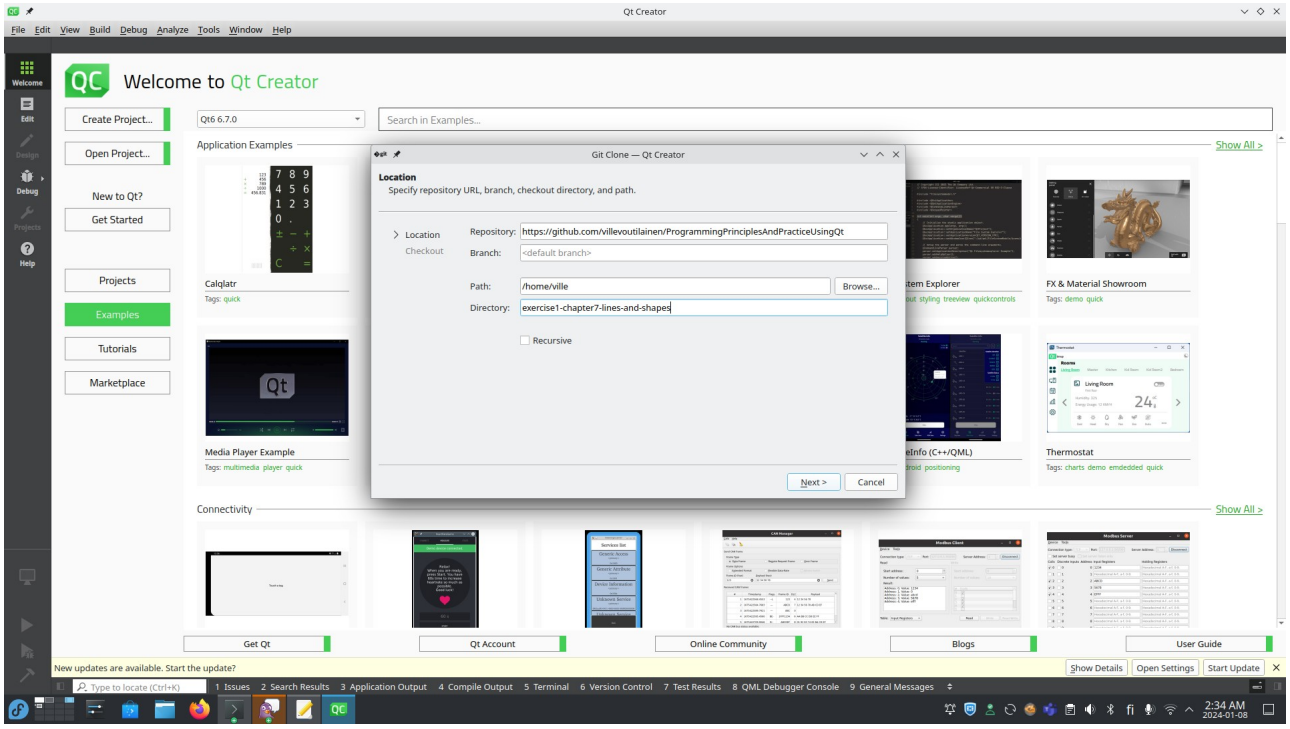
Creating a new project (a new exercise)

The easiest way is to create a new project by cloning our github repository with Qt Creator:

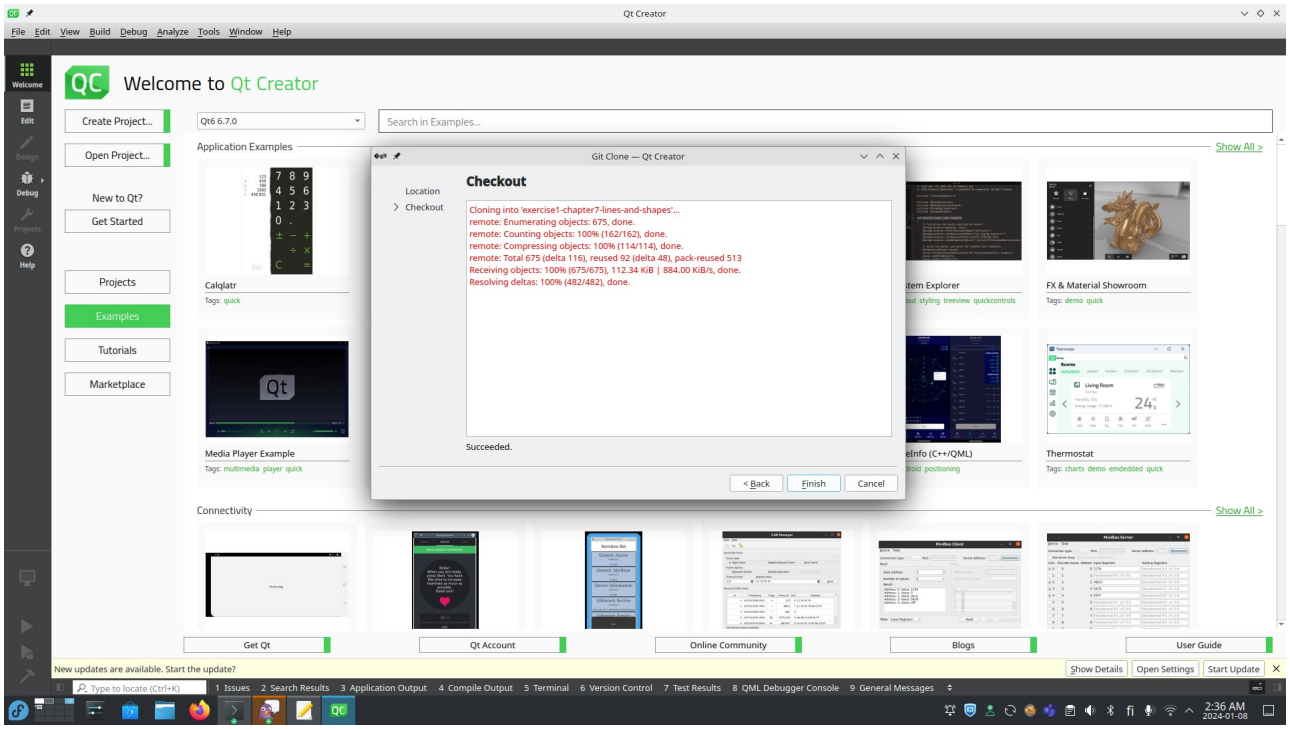
1. Open the File menu, select New Project..
2. Select Import Project, then Git Clone:



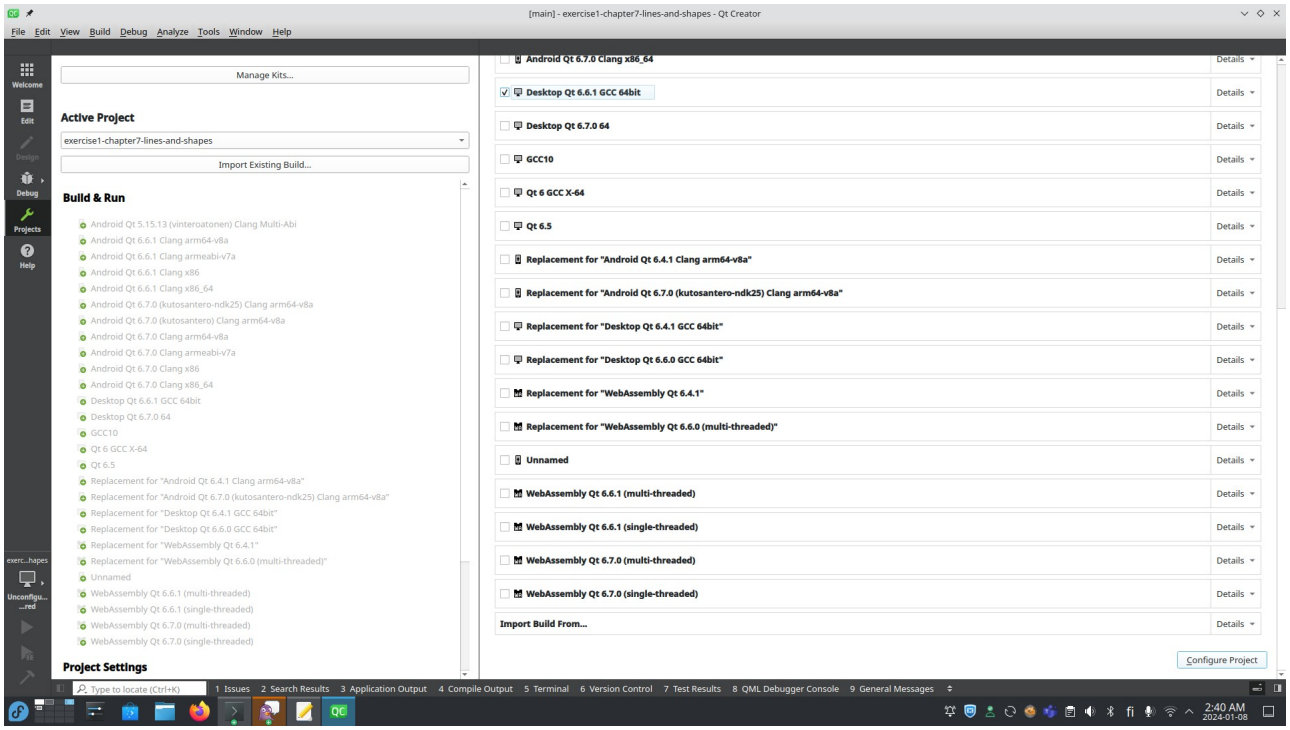
3. Enter https://github.com/villevoutilainen/ProgrammingPrinciplesAndPracticeUsingQt as the repository, choose your desired location for the project:



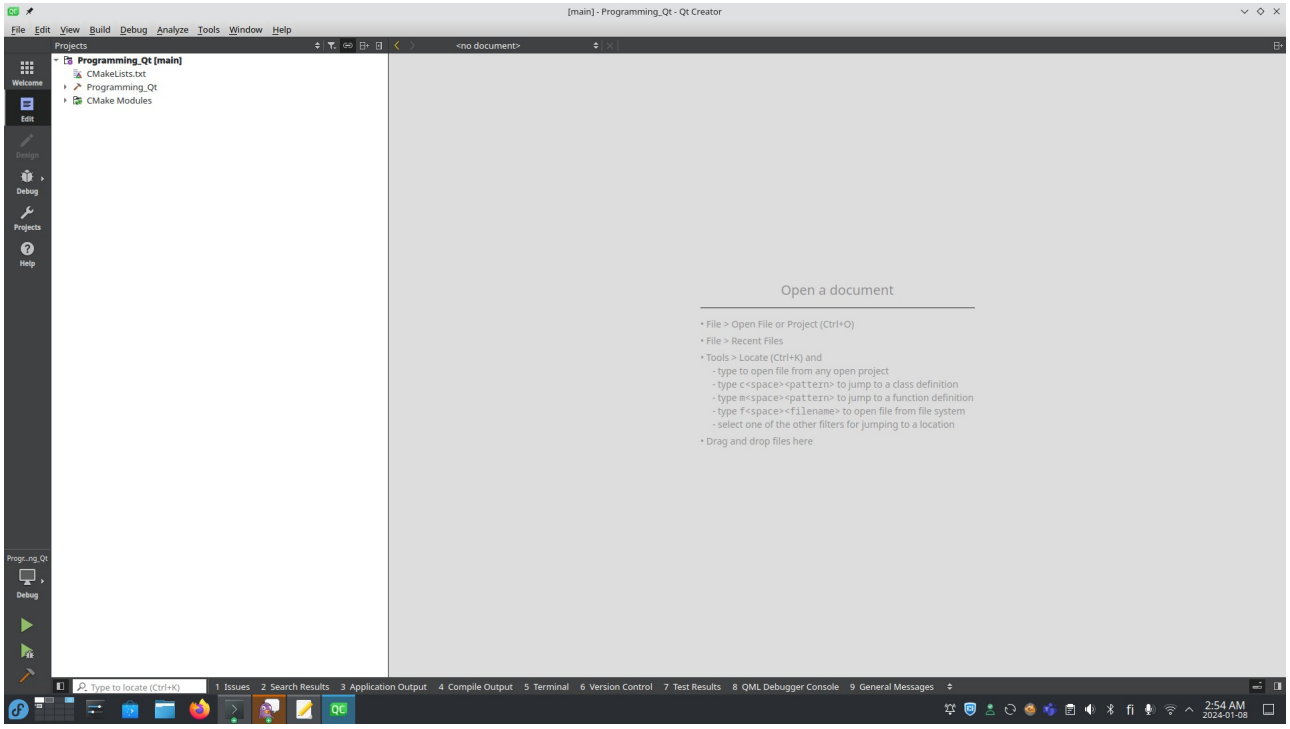
4. Click Next, Creator will clone the project:



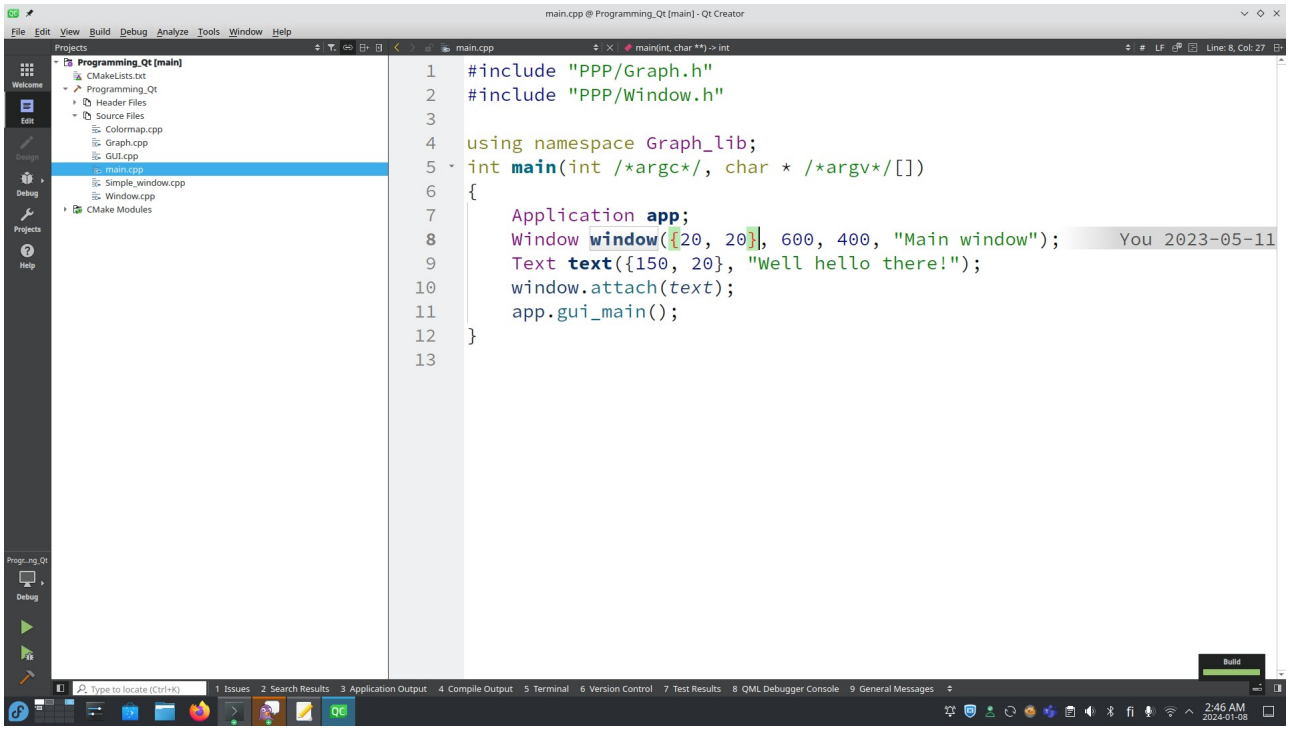
5. Click Finish to complete the project creation.
6. Select a desktop build (don't worry, you won't have as many alternatives as I do, it's going to be less cluttered), and click Configure Project:



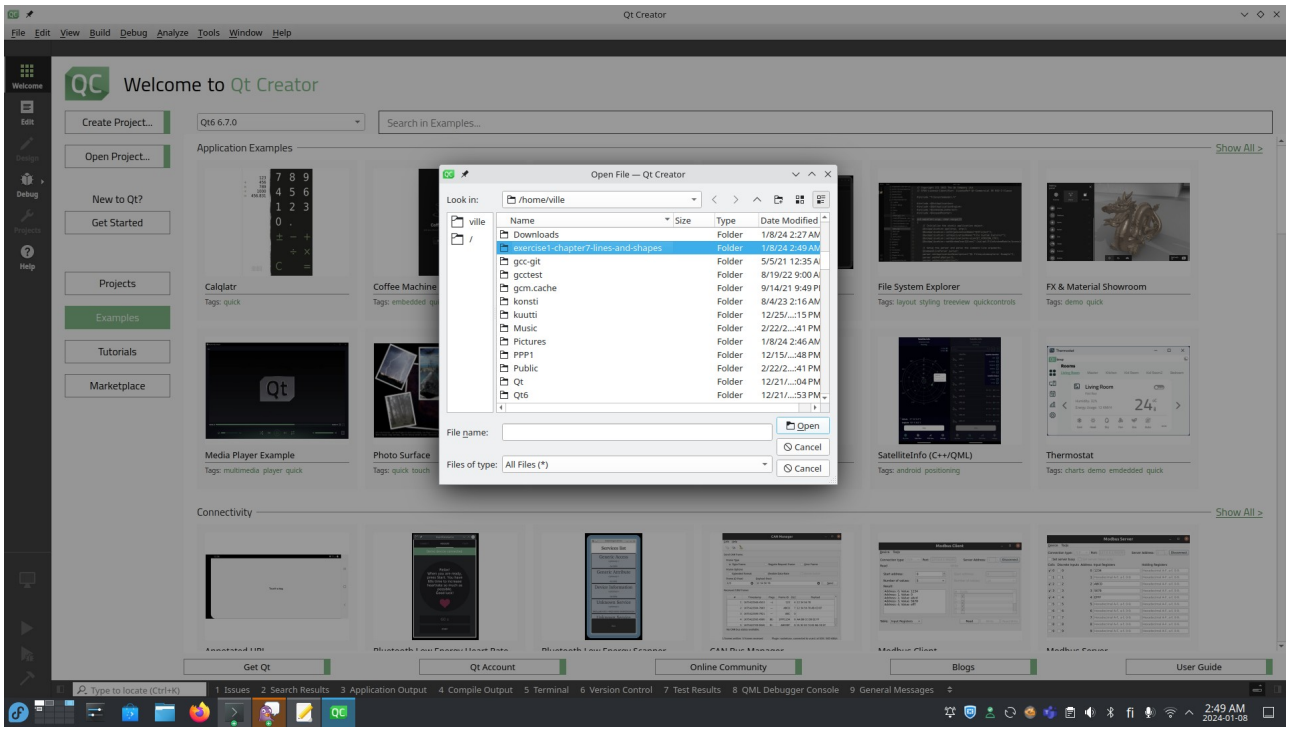
7. Our project will run the build configuration, and is then ready to build:



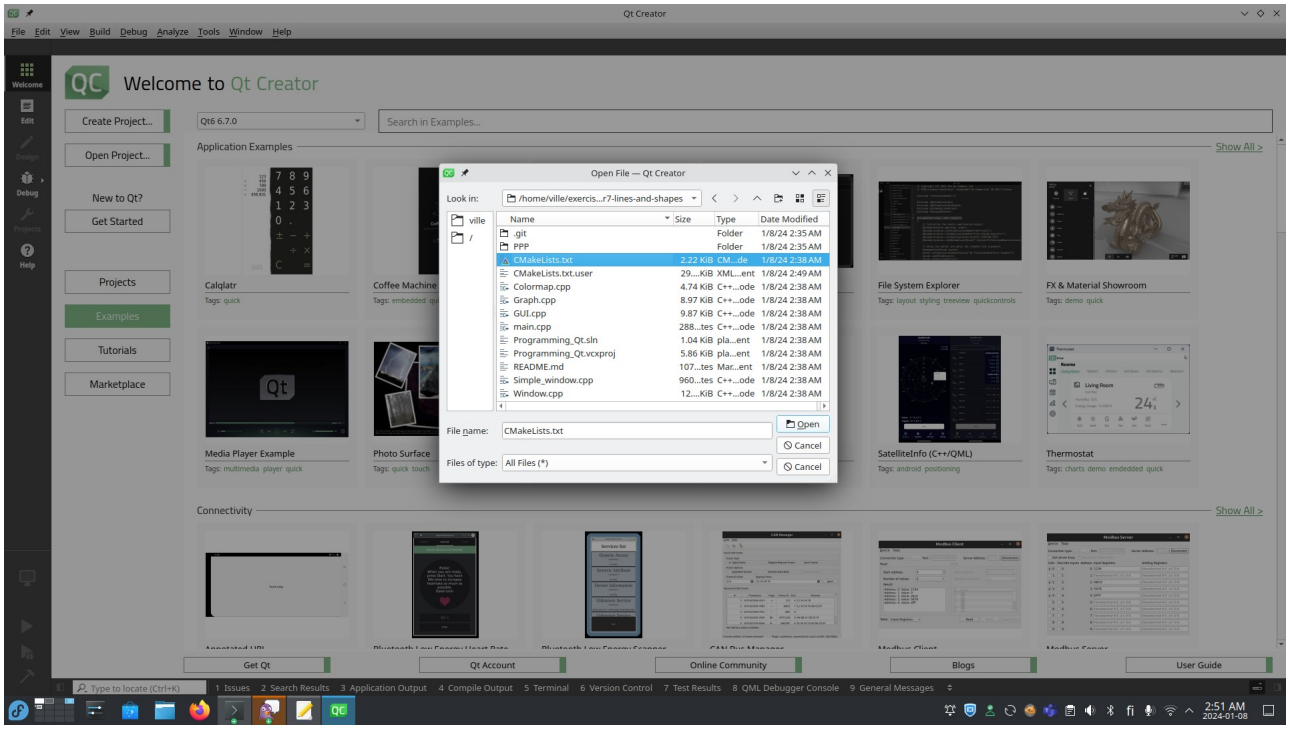
8. Press Ctrl+B to build it. Or on Mac, ⌘+B. Or select Build from the menu, select Build Project "Programming_Qt". Then, under the project tree, open main.cpp:



9. Edit to your liking to implement an exercise, build again, figure out whatever problems you may have, and once done, submit your exercise to your professor. ☺
10. Repeat the process for the next exercise.
11. Note, you do <u>not</u> want to repeat all this to continue working on an exercise if you have closed Qt Creator. To reopen a project, navigate to it via File → Open Project.. and choose its directory:



12. Click Open, select CMakeLists.txt, and click Open again:



13. Now your previous project is ready to be edited.

Instructions for Mac OS

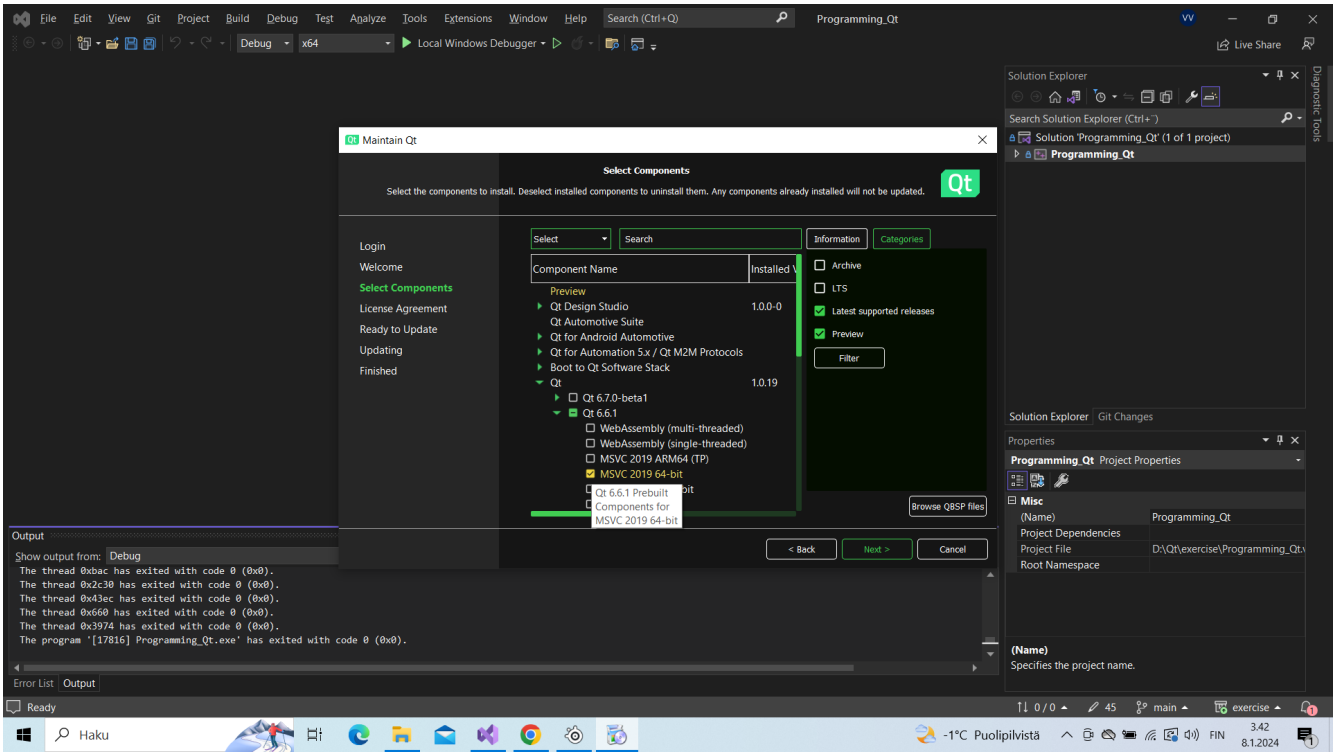
Install Qt as explained before, but before doing so, install XCode from the app store.

Once done, install as explained, clone a new project from github as explained, build and edit as explained.

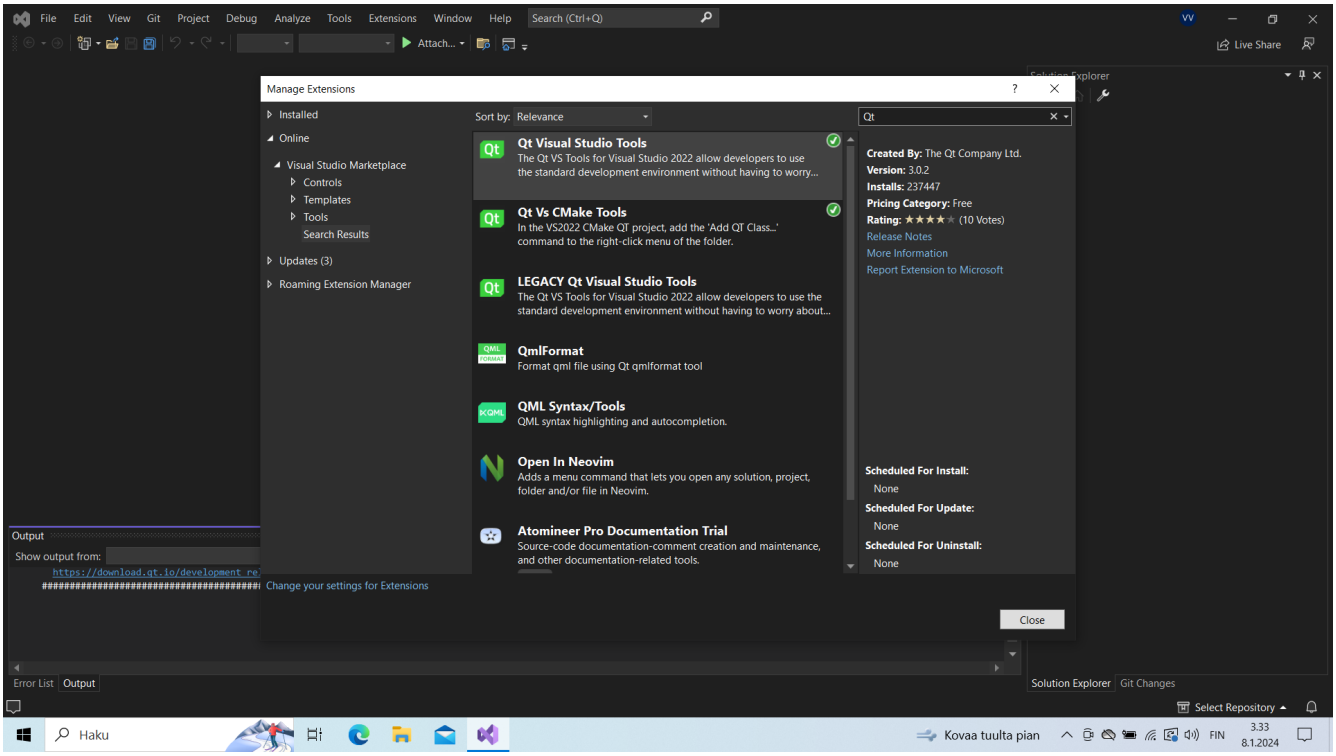
Instructions for using Visual Studio on Windows

Download and install Visual Studio Community from https://visualstudio.microsoft.com/free-developer-offers/

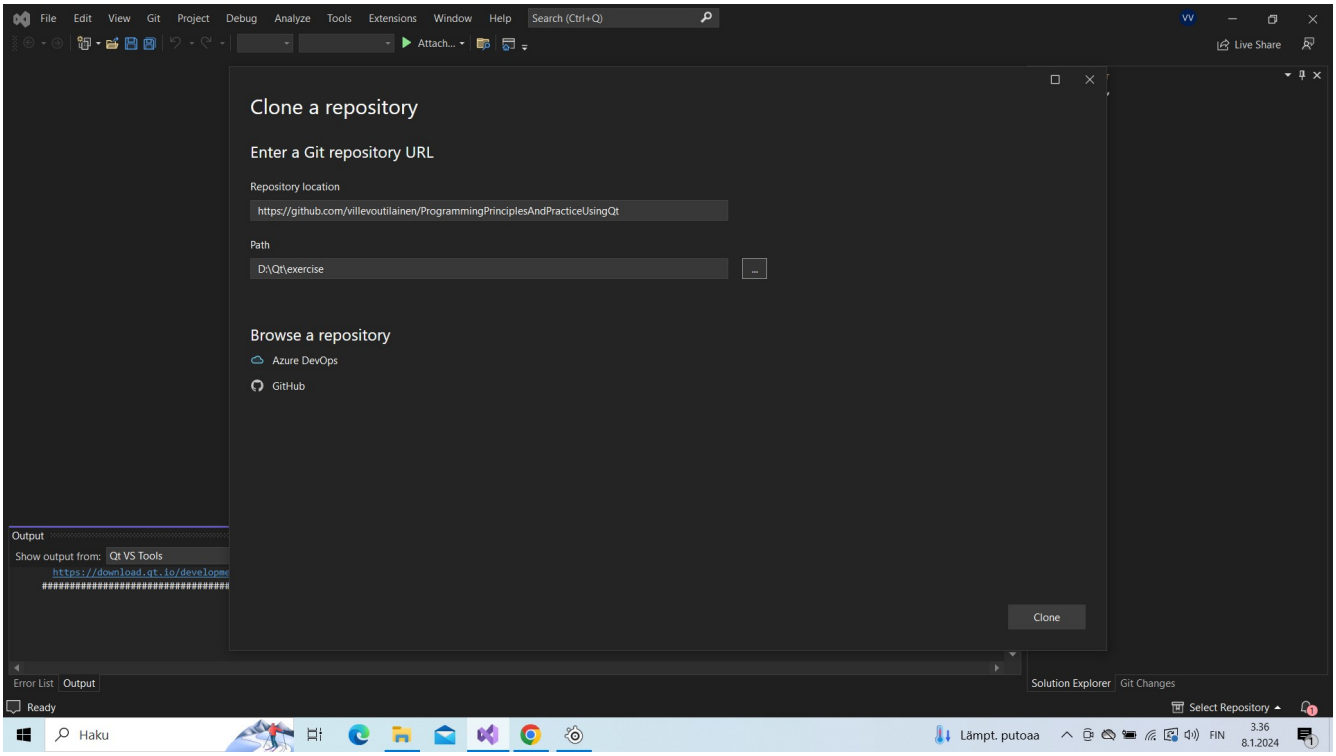
After this, do the Qt installation. Make sure to select the Qt MSVC target:



Once done, open Visual Studio, and if need be, tell it to “Continue without code”. Go to the extensions menu and install Qt Visual Studio Tools:



After the installation, you can use the Git menu of Visual Studio to create a new exercise project:



It will automatically download the project, and open the .sln file in it. That file is what you will look for when reopening an existing exercise, its name will be “Programming_Qt.sln”.