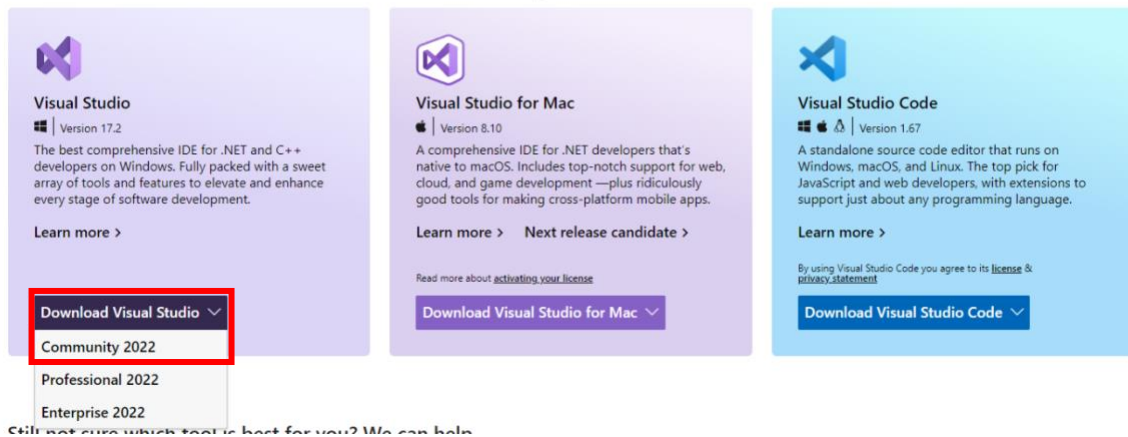


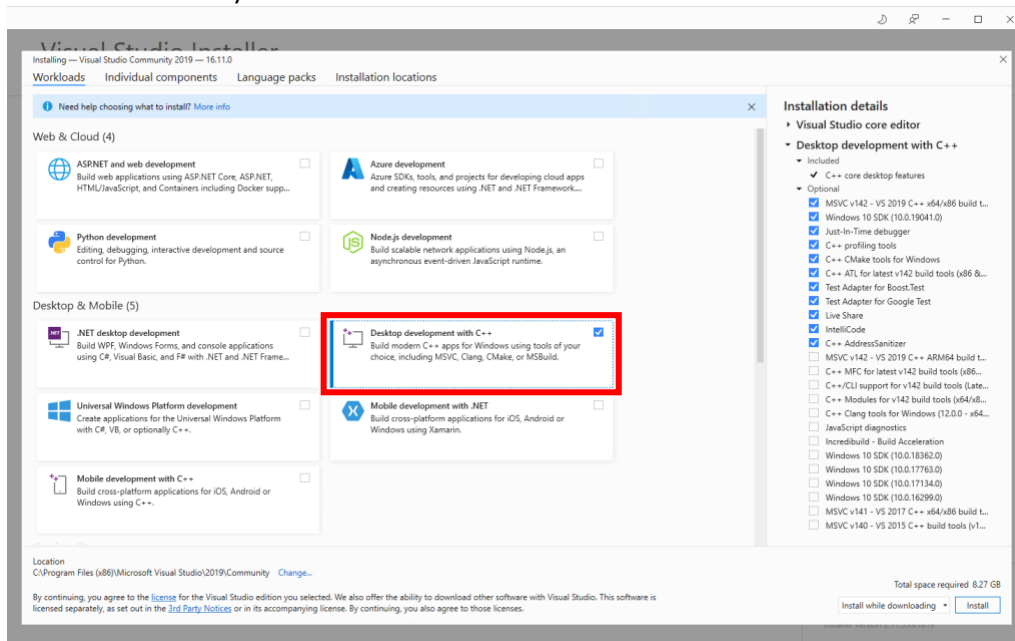
Install Guide

0. Download Microsoft Visual Studio 2022 Community Edition from:
<https://visualstudio.microsoft.com/>

Meet the Visual Studio family

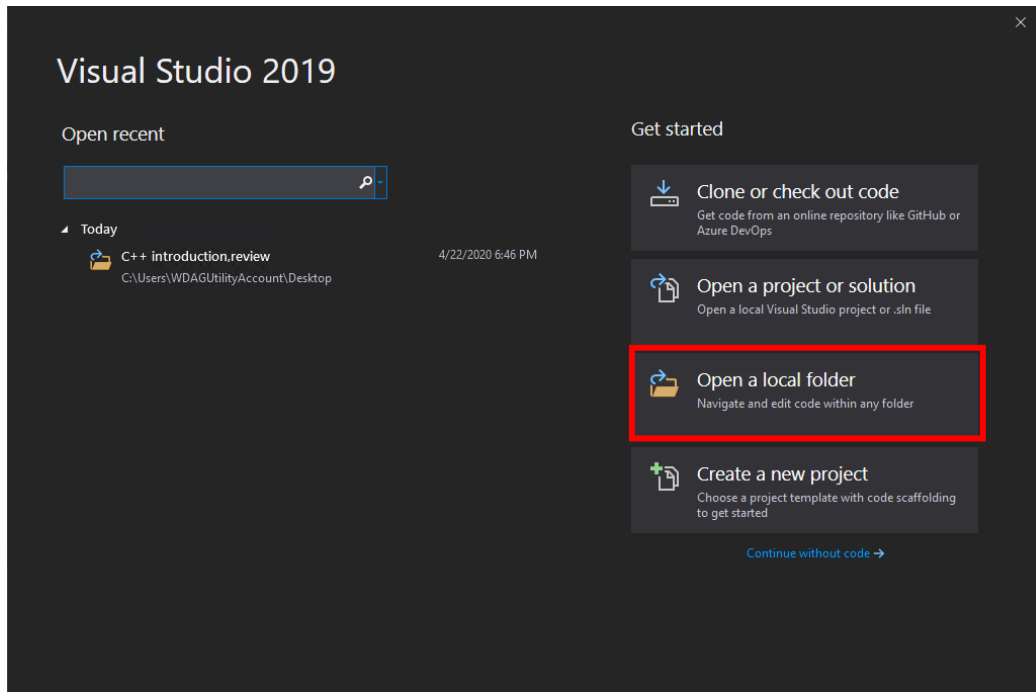


1. In the window asking about workloads, enable **Desktop development with C++** (see illustration below):

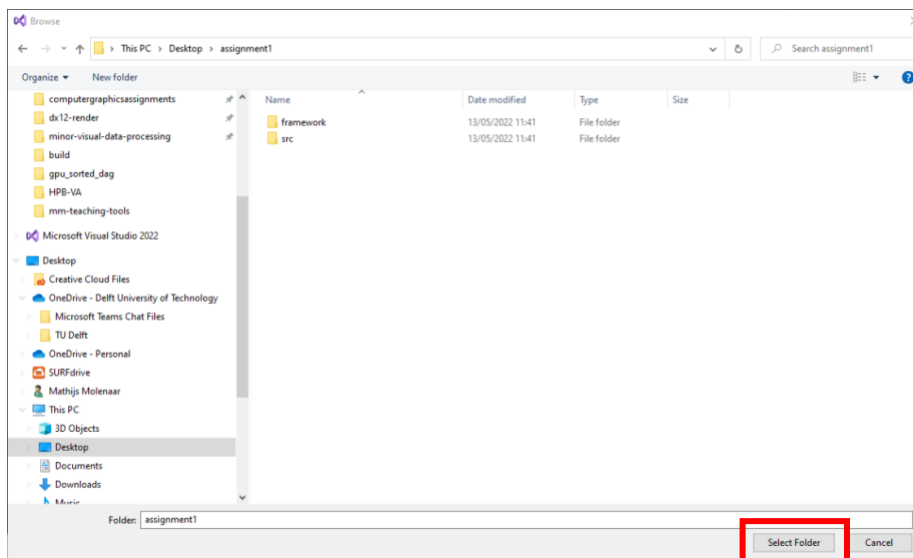


Running the assignments

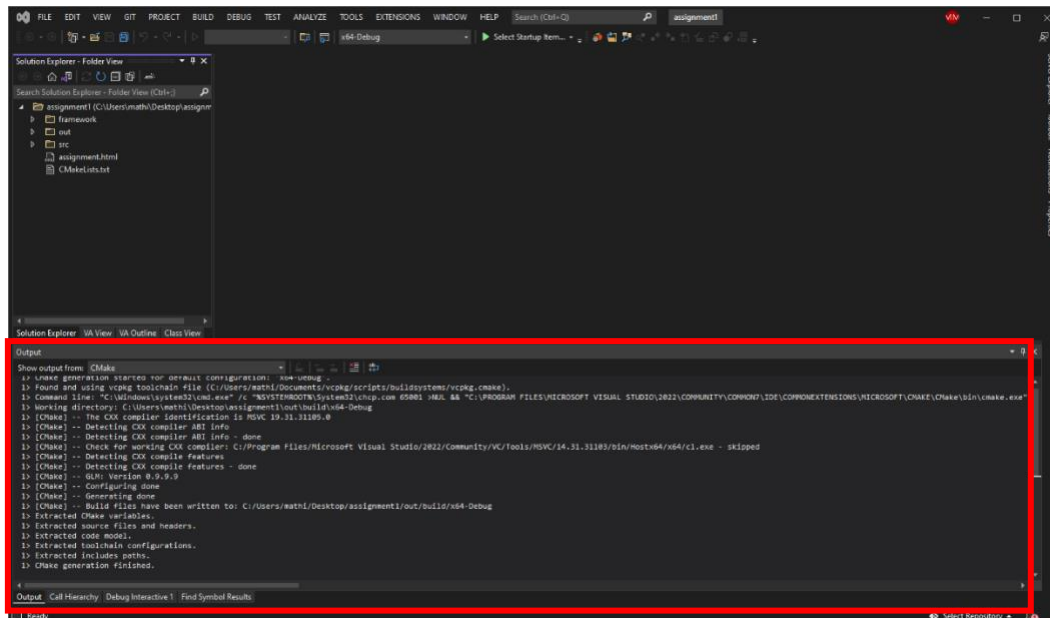
0. Open Visual Studio 2022 from the start menu and in the window that appears, click on “Open a local folder”



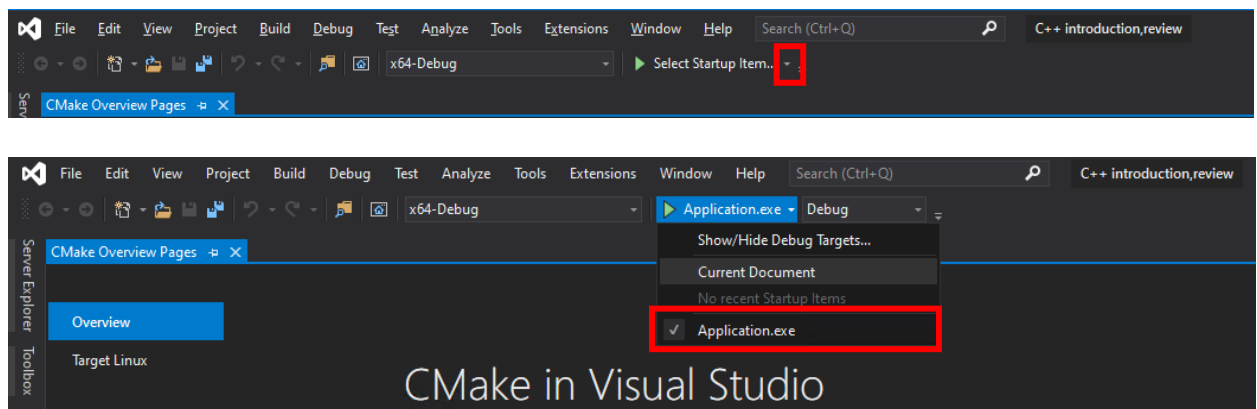
1. Navigate to the root of the project and click “Select Folder”. **Make sure that the path to the folder does not contain any special characters (ASCII only).**



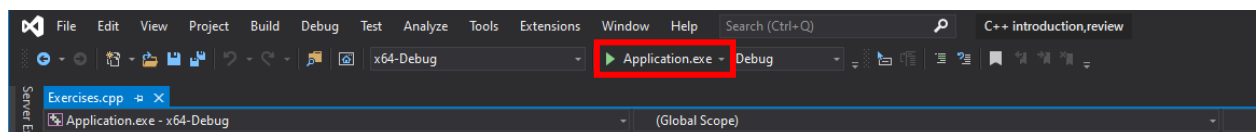
- It will take a couple seconds for Visual Studio to load and configure, after which the output window should show the line “CMake generation finished”.



- Select the executable that you want to compile and run. Click on the little down arrow next to “Select Startup Item...” and click on [NameOfExecutable].exe (for practical one this will be “Practical1.exe”)



- Press the green button to compile & run the program



Debugging

Visual Studio and C++ give you various ways of debugging your code. The simplest and most naïve way is to use print statements. However, we recommend that you use breakpoints instead and only fall back to print statements when necessary.

Print Statements

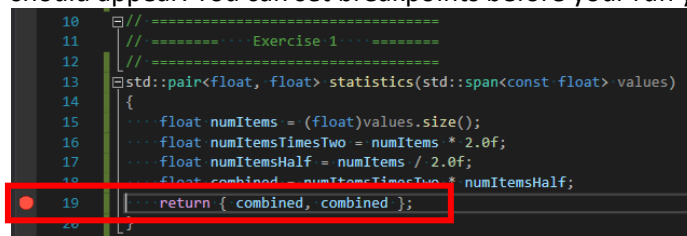
There are various ways to print in C++:

1. Input/Output streams <https://godbolt.org/z/67To8E3fP>
2. fmtlib print <https://godbolt.org/z/WKEPabbM7>
3. C style printf <https://godbolt.org/z/8hWjxYvo1>

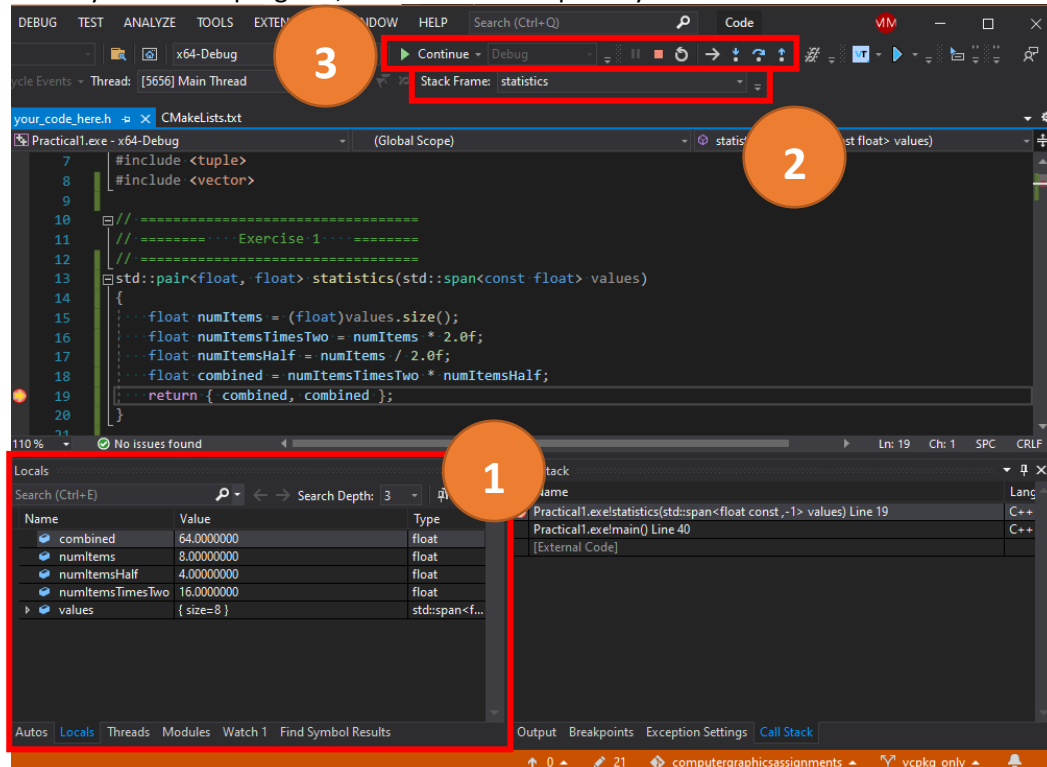
Don't forget to add the relevant `#include <>` at the top of your file

Breakpoints

You can set a breakpoint by clicking on the gray vertical bar to the left of the line numbers, a red bubble should appear. You can set breakpoints before you run your program but also while it is running.



When you run the program, execution will stop every time the selected line is reached.



1. Watch Window

When the watch window in the lower left corner is set to Locals, it will show the values of (most of) the variables in the current scope. This is of great value when debugging.

2. Stack Frame

You can use the stack frame dropdown to look at the calling function(s) while the program is still paused. When you switch to another function in the stack frame the watch window will update with variables that are local to that function.

3. Continue / Step tools

You can continue execution by hitting the Continue button. Use the buttons on the right to step into a function that you are calling, step over to the next line, or step out of the current function respectively. The step over button (F10 on the keyboard) is a great way to go through a buggy piece of code. Work out the computation by hand; Does the code match your pen-and-paper solution?