Installation under Mac OS with Homebrew

The instructions are exactly the same as for the `VegaP` project. Skip the installation part and start with the Section **Install course package**.

Please follow the instructions carefully, line-by-line. This is not the time to be creative!

1. Install Homebrew following link: https://brew.sh/. You just need to copy, paste, and run some script in the macOS terminal.

2. As part of the installation, you will get `CLang` compiler and `LLDB` debugger. To check, type in the terminal:

   ```
   clang --version
   lldb --version
   ```

   If the compiler and the debugger are missing, then install them by typing in the terminal:

   ```
   xcode-select --install
   ```

   Check again that you have `CLang` and `LLDB` properly installed.

3. Install `CMake` and `Visual Studio Code` by typing in the terminal:

   ```
   brew update
   brew upgrade
   brew install cmake
   brew install ninja
   brew install visual-studio-code
   brew install doxygen
   brew install gsl
   brew install graphviz
   brew install pkg-config
   ```

   To check that everything is correctly installed, type in the terminal:

   ```
   code --version
   cmake --version
   ninja --version
   doxygen --version
   pkg-config --version
   gsl-config --version
   dot -V
   ```

Configure Visual Studio Code

Type in the terminal:

```
code
```

It will start the Visual Studio Code. Add `C/C++ Extension Pack` following the link:

ms-vscode.cpptools-extension-pack

Install course package

1. Create a directory, where you will keep the material related to the course. Hereafter, we call this directory Vega.

2. Place file VegaQ.zip in directory Vega and extract it. Check that the directory tree looks like Vega:\VegaQ\test (not as Vega:\VegaQ\VegaQ\test).

3. Open directory Vega\VegaQ with Visual Studio Code. You will be asked to select a Kit for VegaQ. Choose[1]

   ```
   Clang 12.0.0. Using compilers: C = /usr/bin/clang, ...
   ```

   If everything goes well, you will see the output of the kind:

   ```
   [cmake] -- The CXX compiler identification is AppleClang ...
   ```

4. In the future, to get back to your project, just open Visual Studio Code. It remembers the last state.

5. In file \VegaQ\CMakeLists.txt, around line 40, type "YOUR_ID" instead of "Vega". Later, "YOUR_ID" will be the name of your team in Quantathon. You will choose it with your teammates.

6. Build all projects with `(Shift+Fn+F7)` and then choose `(all)`. Sometimes you have to do it a couple of times to clear the errors. Help files in `.html` format will appear in \VegaQ\build\doc. You may want to bookmark some of them for a quick access from your browser.

7. Run project `quantExamples` with `(Shift+Fn+F5)`. Text file quantExamples.txt will be created in directory Vega:\VegaQ\build\output\quantExamples. "YOUR_ID" will appear on the first line.

[1] The version number may be different. It should work just fine.

8. Debug project `quantExamples` with `(Ctrl+Fn+F5)`. The same text file will be created, but the dialog will look different. You may have to do it a couple of times to get a result. The debug mode allows you to add breakpoints with `(Fn+F9)` and then track the values of variables.

9. Check the instructions for `CMake Tools` following the link. Skip all sections related to CMake, just learn how to configure, build, and debug.

10. Useful shortcuts:

    `(Cmd+Shift+P)` opens Command Palette. Type `CMake` to get the commands from CMake Tools. Command Palette remembers the commands used recently. It is my preferred way to work with the Visual Studio Code.

    `(Shift+Fn+F7)` allows you to select and build a specific target.

    `(Fn+F7)` builds the active build target. You can select the build target by opening Command Palette with `(Cmd+Shift+P)` and then typing `(CMake: Set Build Target)`.

    `(Ctrl+Fn+F5)` debugs the active launch or debug target. You can select the launch target by opening Command Palette with `(Cmd+Shift+P)` and then typing `(CMake: Set Debug Target)`.

    `(Shift+Fn+F5)` runs the active launch or debug target.

    `(Shift+Alt+F)` formats your code to look nice.

Remark 1. Do not use the default debug command initiated by `(Fn+F5)`. Press instead `(Ctrl+Fn+F5)`. This way, CMake takes care of all the settings.

Remark 2. If `CMake` misbehaves, then do

Soft reset: `(Cmd+Shift+P)` + `(CMake: Delete Cache and Reconfigure)` + `(CMake: Clean Rebuild)`.

Hard reset: close `Visual Studio Code`, delete directory Vega:\VegaQ\build, and restart `Visual Studio Code`.