Assignment D: Libraries

Sample code for all tasks can be found in the weekD.zip folder. In this assignment, you have to apply your new knowledge about libraries to the code from week A (The globe with Africa, remember?). There, you have to separate the main.cpp from the rest of the code (i.e., World.h and World.cpp become a library called libworld). You can find the code on our website. If you like to, you can use any other sample code.

Task 1: Make your first library [10 points]

Go into the directory 00_makelibrary and take a look at the Makefile. Run make inside his directory to see what is happening. Note that instead of linking .o object files to your main function directly, all other .o files are put in one static library called libobjexport.a. Apply the same to the code from week A.

Task 2: “External” libraries [10 points]

Go to the 01_makestatic directory. Inside, there are two subdirectories. First, go into libobjexport / and run make. Then, go into main/ and run make. Everything should be built as expected. Have a look at the Makefiles and figure out what have been changed in comparison to task A. Apply this to the code of week A.

Task 3: Shared libraries [20 points]

Go inside the directory 02_makedynamic. Compile and run the code. When executing your binary, it’ll not start properly. You have to write

```
LD_LIBRARY_PATH=.:LD_LIBRARY_PATH ./assignment_04
```

If you are working on a Mac, type

```
DYLD_LIBRARY_PATH=.:DYLD_LIBRARY_PATH ./assignment_04
```

So that local shared libraries are used. Apply the same to the code of week A. Please also note that the linking syntax is a little different on the Macs. Have a look at the Makefile for details.

Task 4: CMake and libraries [20 points]

Go into the directory 03_cmake. Note that it contains our two subdirectories and a short CMakeLists.txt. Execute cmake . and make. Everything should be build. Have a look at the CmakeLists.txt files in the subdirectories. If you understand how everything works, try your luck with the code from week A. Also try to build a SHARED instead of a STATIC library (hint: you just have to change a single word in a CMakeLists.txt).
**Task 5: External Libraries! OpenCV [20 points]**

Go into the directory 04_opencv. Have a look at the makefile and run make. Execute your binary and look what happens in the source code. For a basic introduction into OpenCV, go to http://opencv.org/

**Task 6: CMake and OpenCV [20 EXTRA points]**

Go to directory 05_opencv_cmake and have a look at the CMakeLists.txt. Unpack the CMake.zip folder somewhere and write this path to the CMakeLists.txt (CMAKE_MODULE_PATH).