VS code compiler tools are needed for running the multiple files with a single command.   
In this document, my idea is to explore CMake and Make compilar tools.  
Understand basic syntax followed for CMake creation.  
1. Create a Cmake file for 2 files.  
2. Create a Cmake file for running 2-3 multi-threaded applications.

3. Create a Cmake file with include standard libraries path(example openCV, cuda)  
  
  
Youtube  
1. <https://www.youtube.com/watch?v=_BWU5mWqVA4>  
  
  
  
**CMake building procedure:**  
  
Example for simple CMakeLists.txt  
  
‘’’

# Project name and version

project(MyProject)

# Minimum CMake version required

cmake\_minimum\_required(VERSION 3.10)

# Specify the C++ standard

set(CMAKE\_CXX\_STANDARD 17)

set(CMAKE\_CXX\_STANDARD\_REQUIRED True)

# Add the executable target

add\_executable(MyExecutable main.cpp)

‘’’

**1.** **Create Cmakelist for one simple .cpp file**

Created src folder, inside create main.cpp file

Now CMakeLists.txt file changes into  
‘’’

# Project name and version

project(**hello\_cmake**)

# Minimum CMake version required

cmake\_minimum\_required(VERSION 3.10)

# Specify the C++ standard

set(CMAKE\_CXX\_STANDARD 17)

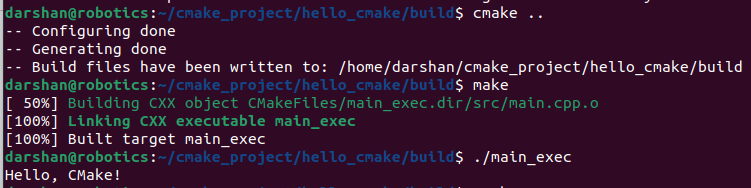
set(CMAKE\_CXX\_STANDARD\_REQUIRED True)

# Add the executable target

add\_executable(**main\_exec src/main.cpp**)

‘’’  
  
**Compilation:**  
mkdir build && cd build

cmake ..  
make

./main\_exec  


**2. Adding extra files with headers**

Now update 2 more files into cmake as support to main.cpp  
Create helper.cpp and utils.cpp which includes helper.h and utils.h files  
  
The CMakeLists.txt will becomes as

‘’’  
# Project name and version

project(**hello\_cmake**)

# Minimum CMake version required

cmake\_minimum\_required(VERSION 3.10)

# Specify the C++ standard

set(CMAKE\_CXX\_STANDARD 17)

set(CMAKE\_CXX\_STANDARD\_REQUIRED True)

**# Add include directory**

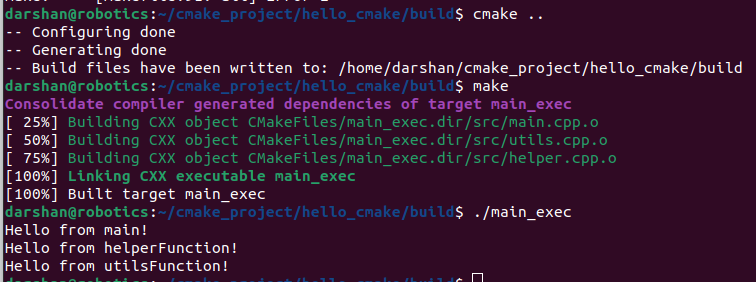
**include\_directories(include)**

# Add the executable target

add\_executable(**main\_exec src/main.cpp**

src/helper.cpp

src/utils.cpp)

‘’’  
**Compilation:**  


**3. Including the standard library(OpenCV) and using it in code.**The CMakeLists.txt will becomes as

‘’’  
# Project name and version

project(**hello\_cmake**)

# Minimum CMake version required

cmake\_minimum\_required(VERSION 3.10)

# Specify the C++ standard

set(CMAKE\_CXX\_STANDARD 17)

set(CMAKE\_CXX\_STANDARD\_REQUIRED True)

**# Add include directory**

**include\_directories(include)**

**# Find OpenCV**

**find\_package(OpenCV REQUIRED)**

# Add the executable target

add\_executable(**main\_exec src/main.cpp**

src/helper.cpp

src/utils.cpp)

**# Gather only the OpenCV-related source file**

**set(OPENCV\_SOURCE "src/image\_view.cpp")**

**# Add OpenCV executable**

**add\_executable(ImageViewer ${OPENCV\_SOURCE})**

**target\_link\_libraries(ImageViewer ${OpenCV\_LIBS})**

‘’’