D-SOFT IoTLab
C++ Training Material

Written by: Phuoc NGUYEN
March 22nd 2022
Revision 1.0

Prerequisite

All exercises/tasks need to conform to a coding style.
In our case, we will follow Google Coding Style for C++.
In the case of using any external libraries, you need to install it on your own.
However, you will be asked about the installation procedure, so take note of those steps.

CMake - C++ Class - Unit Test with Google

Deadline: you are required to submit your solution before **12PM on Thursday 24th 2022**.

- Your code needs to be compiled successfully when review
- You have to answer every aspect in your code
- You can discuss the approach with your fellows, but each needs to write and understand the code clearly.
    - You don't have any grades here, you are doing this for your own benefit.
    - In case of getting stuck (even after consulting every material), you can come and ask me in person

Task 1: getting familiar with CMake

You are given a sample program with one CMake file. Run it and understand its basic content.
You are required to write a program which:

- utilizes multiple source files (as a library) and
- uses CMake to compile it

```
.
├── CMakeLists.txt <given>
├── include/
│   ├── foo.h
├── main.cpp <given>
└── src/
    ├── CMakeLists.txt
    ├── cmake_multiple_sources/
    │   └── main.cpp
    ├── lib/
    │   ├── foo.cpp
```

Task 2: write two types of Png classes in C++

In this task, you are required to write to C++ classes with a member function that opens a PNG image file and prints the image info (such as height and width in pixels).
One class is named `**PngppImage**` and the other is `**LodepngImage**`.

- Each class must have the following function:
  bool open_image(std::string image_path);
    - Return true if the image can be opened successfully, using its corresponding library
    - Return false otherwise
- `**PngppImage**` must use **libpng++** library
- `**LodepngImage**` must use **LodePNG** library
- Hint: to check for file existence, you can use **filesystem** library
- Feel free to add any other functions if needed

Write a sample program to invoke those two classes.
Sample run:

```
$ ./app_images
Sample Test Program for PngppImage and LodepngImage.
Use class `PngppImage` to read an image file.
W: 160
H: 160
Test of PngppImage is successful
Use class `LodepngImage` to read an image file.
W: 160
H: 160
Test of LodepngImage is successful
```

Directory structure:

- NOTE: you need to adapt in case of adding a new library, but basically, you need to conform to the following convention:
    - Header of library must be in **include**/
    - Source file of library must be in **lib**/
    - Test file must be in **test**/ (applicable to task 3)

```
.
├── CMakeLists.txt
├── data/
│   ├── data_info.txt
│   └── linux_test.png
├── include/
│   ├── foo.h
│   ├── lodepng_image.h
│   └── pngpp_image.h
├── main.cpp
├── README.md
└── src/
    ├── cmake_images/
    │   └── main.cpp
    ├── CMakeLists.txt
    ├── cmake_multiple_sources/
    │   └── main.cpp
    ├── lib/
    │   ├── foo.cpp
    │   ├── lodepng_image.cpp
    │   └── pngpp_image.cpp
    └── test/
```

Task 3: write Unit Test

Write a unit test to test for member function implementation of the two classes in task 2.
You will use **Google Test** to write the unit tests.
Sample run:

```
[==========] Running 2 tests from 1 test suite.
[----------] Global test environment set-up.
[----------] 2 tests from PngppImageTest
[ RUN      ] PngppImageTest.ValidFilePath
[       OK ] PngppImageTest.ValidFilePath (0 ms)
[ RUN      ] PngppImageTest.InvalidFilePath
[       OK ] PngppImageTest.InvalidFilePath (0 ms)
[----------] 2 tests from PngppImageTest (0 ms total)

[----------] Global test environment tear-down
[==========] 2 tests from 1 test suite ran. (0 ms total)
[  PASSED  ] 2 tests.
```

References

- Cmake
  - https://cmake.org/cmake/help/latest/
- Google Test
  - https://github.com/google/googletest
- Filesystem
  - https://en.cppreference.com/w/cpp/filesystem
- Pngpp
  - https://www.nongnu.org/pngpp/doc/0.2.9/
- Lodepng
  - https://github.com/lvandeve/lodepng
- Google Style for C++
  - https://google.github.io/styleguide/cppguide.html#Naming
- TBA