Hamza Rauf:

I use google test framework for unit testing.

Google Test is a unit testing library for the C++ programming language, based on the xUnit architecture. The library is released under the BSD 3-clause license. It can be compiled for a variety of POSIX and Windows platforms, allowing unit-testing of C sources as well as C++ with minimal source.

Installation:

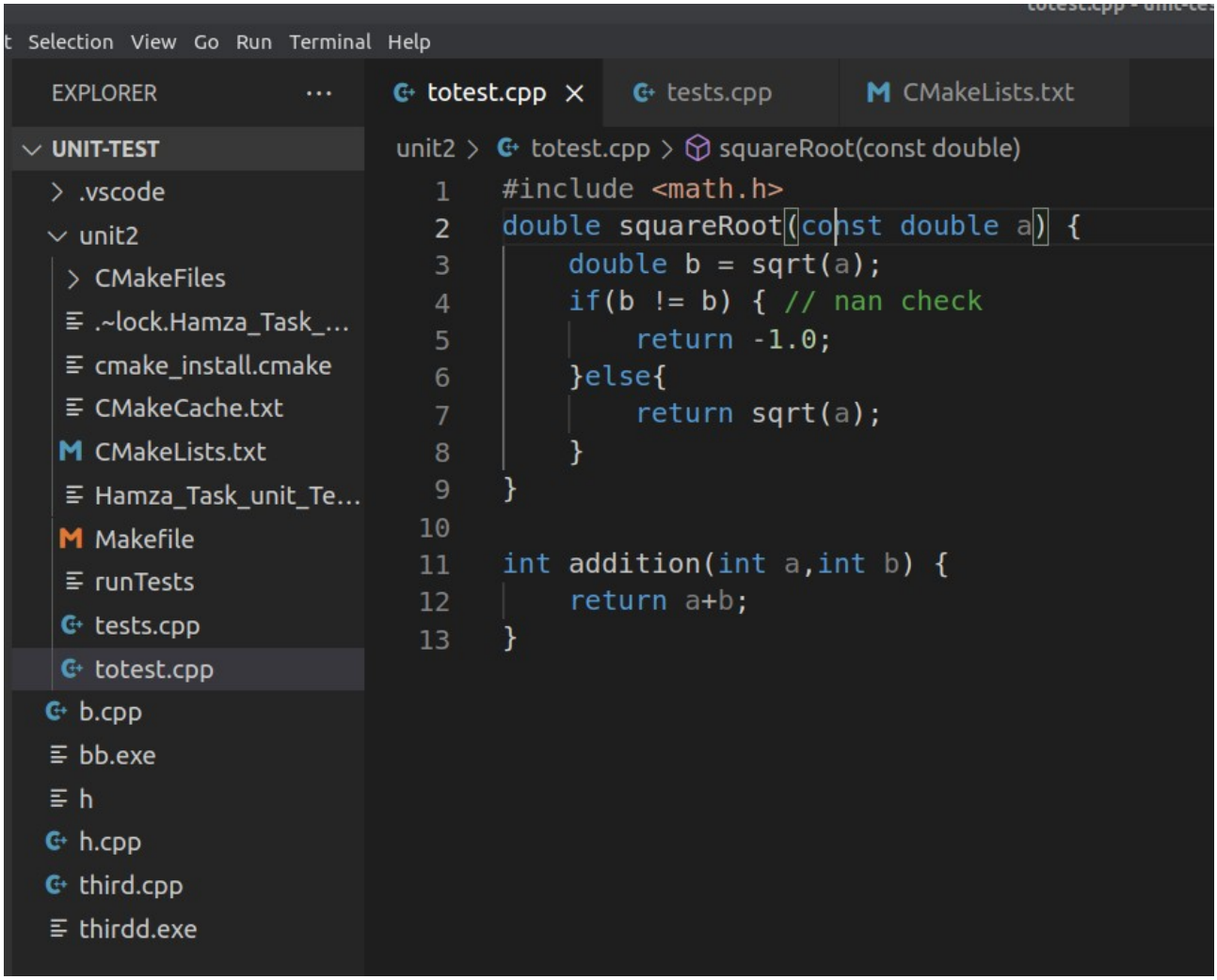
sudo apt-get install libgtest-dev

compile the library source file and copy it to the usr/lib/

```
sudo apt-get install cmake # install cmake
cd /usr/src/gtest
sudo cmake CMakeLists.txt
sudo make

# copy or symlink libgtest.a and libgtest_main.a to your /usr/lib folder
sudo cp *.a /usr/lib
```

main code file which contains functions as a units:



writing test as an assertions:

```
#include "totest.cpp"
#include <gtest/gtest.h>

TEST(SquareRootTest, PositiveNos) {
    ASSERT_EQ(5, squareRoot(36.0));
    ASSERT_EQ(18.0, squareRoot(324.0));
    ASSERT_EQ(25.4, squareRoot(645.16));
    ASSERT_EQ(0, squareRoot(0.0));
}

TEST(SquareRootTest, NegativeNos) {
    ASSERT_EQ(-1.0, squareRoot(-15.0));
    ASSERT_EQ(-1.0, squareRoot(-0.2));
}

TEST(AdditionTest, test2) {
    ASSERT_EQ(2, addition(1,1));
    ASSERT_EQ(0, addition(0,0));

}

int main(int argc, char **argv) {
    testing::InitGoogleTest(&argc, argv);
    return RUN_ALL_TESTS();
}
```

the making cmake file to add the directives and instructions to add google test source files :

```
cmake_minimum_required(VERSION 2.6)

# Locate GTest
find_package(GTest REQUIRED)
include_directories(${GTEST_INCLUDE_DIRS})

# Link runTests with what we want to test and the GTest and pthread library
add_executable(runTests tests.cpp)
target_link_libraries(runTests ${GTEST_LIBRARIES} pthread)
```

make our project:

```
TEST(SquareRootTest, PositiveNos) {
    ASSERT_EQ(5, squareRoot(36.0));
    ASSERT_EQ(18.0, squareRoot(324.0));
    ASSERT_EQ(25.4, squareRoot(645.16));
    ASSERT_EQ(0, squareRoot(0.0));
}
```

```
[  PASSED  ] 3 tests.
hamzapc@hamzapc-ROG-Zephyrus-G14-GA401IV-GA401IV:~/Downloads/unit-test/unit2$ make
Scanning dependencies of target runTests
[ 50%] Building CXX object CMakeFiles/runTests.dir/tests.cpp.o
[100%] Linking CXX executable runTests
[100%] Built target runTests
hamzapc@hamzapc-ROG-Zephyrus-G14-GA401IV-GA401IV:~/Downloads/unit-test/unit2$ ./runTests
[==========] Running 3 tests from 2 test suites.
[----------] Global test environment set-up.
[----------] 2 tests from SquareRootTest
[ RUN      ] SquareRootTest.PositiveNos
/home/hamzapc/Downloads/unit-test/unit2/tests.cpp:5: Failure
Expected equality of these values:
  5
  squareRoot(36.0)
    Which is: 6
[  FAILED  ] SquareRootTest.PositiveNos (0 ms)
[ RUN      ] SquareRootTest.NegativeNos
[       OK ] SquareRootTest.NegativeNos (0 ms)
[----------] 2 tests from SquareRootTest (0 ms total)

[----------] 1 test from AdditionTest
[ RUN      ] AdditionTest.test2
[       OK ] AdditionTest.test2 (0 ms)
[----------] 1 test from AdditionTest (0 ms total)

[----------] Global test environment tear-down
[==========] 3 tests from 2 test suites ran. (0 ms total)
[  PASSED  ] 2 tests.
[  FAILED  ] 1 test, listed below:
[  FAILED  ] SquareRootTest.PositiveNos

 1 FAILED TEST
hamzapc@hamzapc-ROG-Zephyrus-G14-GA401IV-GA401IV:~/Downloads/unit-test/unit2$
```

and execute the runtest executables file to run all the tests:
as 1 test failed cause the square root of 36 is 6 we assume it to be 5.