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:

```
Selection View Go Run Terminal Help

⊕ totest.cpp ×

    tests.cpp
    tests.c
                                                                                                                                                                                                                                                                            M CMakeLists.txt
         EXPLORER

∨ UNIT-TEST

                                                                                                                      #include <math.h>
         > .vscode
                                                                                                                                                        double squareRoot(const double a) {
                                                                                                                                   2

√ unit2

                                                                                                                                                                              double b = sqrt(a);
             > CMakeFiles
                                                                                                                                                                              if(b != b) { // nan check
             return -1.0;
             ≡ cmake install.cmake
                                                                                                                                                                              }else{
             return sqrt(a);
           M CMakeLists.txt
                                                                                                                                                                              }

    Hamza_Task_unit_Te...

           M Makefile
                                                                                                                                                         int addition(int a,int b) {
                                                                                                                              11
            ≡ runTests
                                                                                                                                                                              return a+b;
                                                                                                                              12
            c tests.cpp
                                                                                                                              13
            c totest.cpp

← b.cpp

       ≣ bb.exe
       ≣ h

← h.cpp

      c third.cpp
```

writing test as an assertions:

```
EXPLORER

    ⊕ totest.cpp

                                                                                                                              UNIT-TEST
                                                                           unit2 > G tests.cpp > 分 TEST(SquareRootTest, NegativeNos)
                                                                                                 #include "totest.cpp"
 > .vscode
                                                                                                 #include <gtest/gtest.h>
∨ unit2
   > CMakeFiles
                                                                                                 TEST(SquareRootTest, PositiveNos) {
   ≡ .~lock.Hamza_Task_...
                                                                                                                ASSERT_EQ(5, squareRoot(36.0));
   ≡ cmake_install.cmake
                                                                                                                ASSERT_EQ(18.0, squareRoot(324.0));
   ASSERT_EQ(25.4, squareRoot(645.16));
  M CMakeLists.txt
                                                                                                                ASSERT EQ(0, squareRoot(0.0));
   ≣ Hamza_Task_unit_Te...
  M Makefile
                                                                                                  TEST(SquareRootTest, NegativeNos) {
                                                                               11

    □ runTests

                                                                                                                ASSERT EQ(-1.0, squareRoot(-15.0));
   c tests.cpp
                                                                                                                 ASSERT_EQ(-1.0, squareRoot(-0.2));

    totest.cpp
    totest.cpp

€ b.cpp
                                                                                                  TEST(AdditionTest, test2) {
 ≣ bb.exe
                                                                                                                 ASSERT_EQ(2, addition(1,1));
 ≣ h
                                                                                                                ASSERT_EQ(0, addition(0,0));
C+ h.cpp
c third.cpp
 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:

```
CMakeLists.txt - unit-test - Visual Studio Cod
 EXPLORER
                                                              M CMakeLists.txt ×
                 unit2 > M CMakeLists.txt

∨ UNIT-TEST

                           1 cmake_minimum_required(VERSION 2.6)
 > .vscode

✓ unit2

 > CMakeFiles 3 # Locate GTest

≤ .~lock.Hamza_Task_...

E cmake_install.cmake 6

E CMakeCache.txt 7 # Link runTests with what we want to test and the GTest and pthread library

M CMakeLists.txt 8 add_executable(runTests tests.cpp)
  M Makefile

≡ runTests

    tests.cpp

    totest.cpp

 € b.cpp
 ≣ bb.exe
 ≣h
 G+ h.cpp
 c third.cpp
  ≡ thirdd.exe
```

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));
}
```

```
amzapc@hamzapc-ROG-Zephyrus-G14-GA401IV-GA401IV:-/Downloads/unit-test/unit2$ make amzapc@hamzapc-ROG-Zephyrus-G14-GA401IV:-/Downloads/unit-test/unit2$ make amzapc@hamzapc-ROG-Zephyrus-G14-GA401IV:-/Downloads/unit-test/unit2$ make amzapc.gd dependencies of target runTests

50%] Building CXX object CMakeFiles/runTests.dir/tests.cpp.o

100%] Linking CXX executable runTests
[100%] Built target runTests
                         OG-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
                 SquareRootTest.PositiveNos
/home/hamzapc/Downloads/unit-test/unit2/tests.cpp:5: Failure
Expected equality of these values:
  squareRoot(36.0)
    Which is: 6
                 SquareRootTest.PositiveNos (0 ms)
                SquareRootTest.NegativeNos
          OK ] SquareRootTest.NegativeNos (0 ms)
----] 2 tests from SquareRootTest (0 ms total)
          ----] 1 test from AdditionTest
                ] AdditionTest.test2
               AdditionTest.test2 (0 ms)

1 test from AdditionTest (0 ms total)
    ===] 3 tests from 2 test suites ran. (0 ms total)
   PASSED ] 2 tests.
                  1 test, listed below:
                  SquareRootTest.PositiveNos
   mzapc@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.