Francisco Bryan Ortega

CS-405-13765-M01

Secure Coding

Ahlam Alhweiti

09/29/2024

4-2 Milestone: Unit Testing

While unit testing is a familiar concept, I was new to the GoogleTest framework and most of my challenges came from exploring the resources around the framework. For each test I decided to employ the GIVEN-WHEN-THEN structure (Fowler, 2013) to systematically convert the predefined conditions for each test into a logical pattern that validated the expected behavior and result. I then spent a good while understanding the existing test cases, macros, and syntax that was used here to make the remaining tests equally readable and logical.

The first few tests were straightforward as I simply validated that the initial conditions for the collection were empty and that the size remained 0 in the GIVEN block. The WHEN block was the primary change for each as I made sure to take the action as defined by the test here. Lastly, the THEN block checked the same conditions that were initially checked and validated that the action taken earlier resulted in the expected change for the test.

An important thing to note about two specific tests was the IsMaxSizeGreaterThanOrEqualToSize and IsCapacityGreaterThanOrEqualToSize were refactored near the end to make them more manageable. I originally wrote them to ASSERT that the collection was empty, added the relevant number of entries, and then validated the max size or capacity was greater than or equal to the size for each action before adding more. This resulted in lengthy unit tests that were not adherent to the GIVEN-WHEN-THEN structure so a for loop was used to dynamically test the given number of entries per case in a readable and extensible manner.

The next few tests presented no significant challenges. The main consideration was simply looking in the distinct functions being tested, e.g. resize(), clear(), erase(), reserve(), etc., and then ensuring that the THEN block validated the correct conditions for the test.

Lastly, probably the most difficult aspect of unit testing was understanding the nuance between a positive and negative test. The provided definition in the guidelines helped with this and a short while exploring the ASSERT\_THROW macro allowed me to confidently write the unit test and consider a good test case for the other required negative test. The GoogleTest framework and functionality of Visual Studio like Test Explorer made unit testing much more approachable and logical to think through.

References

Fowler, M. (2013, August 21). *Bliki: Given when then*. martinfowler.com. https://martinfowler.com/bliki/GivenWhenThen.html

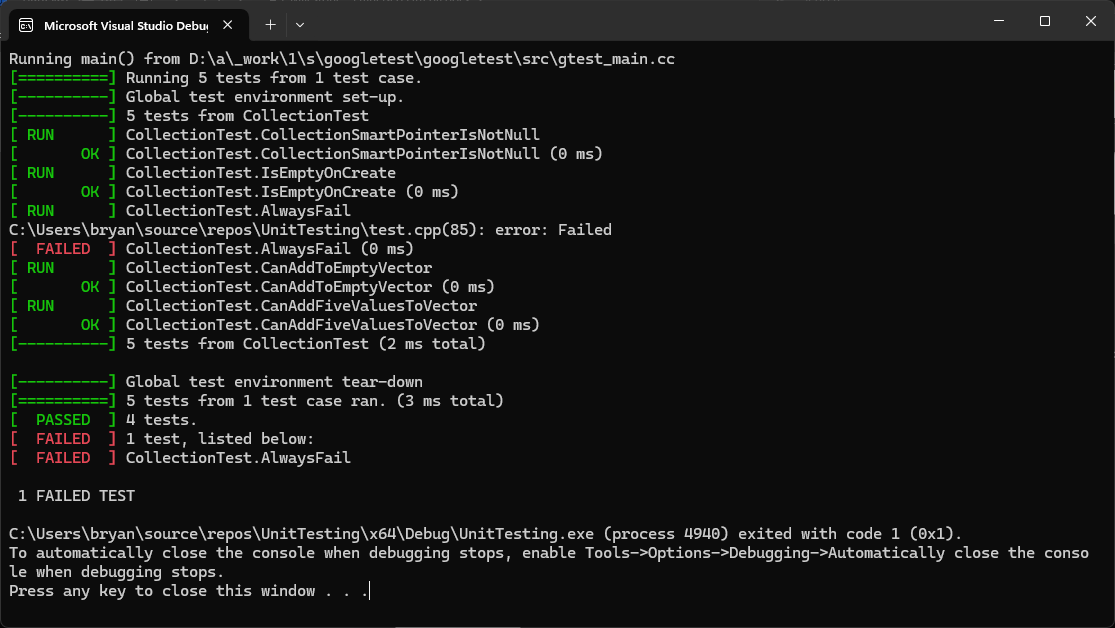


Figure 1 – Screenshot of Unit Testing program before code change.

A screenshot of a computer program

Description automatically generated

Figure 2 – Screenshot of Unit Testing program after code change.