

Implementing the unit tests took a little bit of trial and error before I got used to the syntax. I’ve learned that I can define tests in a block starting with TEST\_F, with the first parameter being a derived class of Test and the second being the test name. Inside, I can use methods such as ASSET\_EQ, ASSERT\_GE, ASSET\_NE, ASSERT\_THROWS, ASSERT\_TRUE, ASSERT\_FALSE, and many more to test specific conditions. Writing negative tests proved to be interesting as I learned how many of the vector methods don’t necessarily throw exceptions but rather result in undefined behavior if invalid or unexpected parameters are passed. For example, running <Vector>.push\_back() on an empty vector results in undefined behavior—in my case, increasing the size by an insane amount. I wanted to find how I could use undefined behavior in my test cases, but it would cause the tests to hang.