

Testing

We strongly encourage testing your code yourself, especially after each part (rather than all at the end). The given tests for this project (even more so than previous projects) are **not** comprehensive tests: it **is** possible to write incorrect code that passes them all.

Things that you might consider testing for include: anything that we specify in the comments or in this document that a method should do that you don't see a test already testing for, and any edge cases that you can think of. Think of what valid inputs might break your code and cause it not to perform as intended, and add a test to make sure things are working.

To help you get started, here is one case that is **not** in the given tests (and will be included in the hidden tests): if a transaction holds IX(database), IS(table), S(page) and promotes the database lock to a SIX lock via `LockContext#promote`, `numChildLocks` should be updated to be 0 for both the database and table contexts.

To add a unit test, open up the appropriate test file (all test files are located in `src/test/java/edu/berkeley/cs186/database` or subdirectories of it), and simply add a new method to the test class, for example:

```
@Test
public void testPromoteSIXSaturation() {
    // your test code here
}
```

Many test classes have some setup code done for you already: take a look at other tests in the file for an idea of how to write the test code.

[Previous](#)
[Part 2: Multigranularity](#)

[Next](#)
[Submitting the Assignment](#)

Last updated 4 months ago