Intro

The testing of our program will be done using Test Driven Development. Test Driven Development is a type of Unit Testing done before any program code has been written. Given the requirements of our methods and entities, specific test cases are created in which our software needs to pass. The software is then incrementally improved to pass all the test cases.

This type of testing is beneficial because it forces, at the minimal, a working program. After first completing our “must-haves”, we then are able to focus on “should-haves” followed by “nice-to-haves”, and not in a less optimal order. By doing the testing incrementally we negate the possibility of being unsure as to what part of our code is no longer working, for we would have adequately tested everything previous to the most recent addition.

All of our tests will be located in a main() method of each entity. This splits up the testing for each entity locally to itself. By using asserts for each test case we will eliminate the possibility for unknowingly assume a test case has passed, when in fact it has not, for an assert will log itself as having failed, and exits the running of the program.

Test Driven Development has certain advantages over other testing methodologies such as System Testing and Integration Testing. Namely, Integration Testing, which is the testing of individual software modules combined, is redundant. This is because, given our individual entities which have all passed their unit tests, our mock-inputs and inputs coming from another entity will be identical. With regards to System Testing: if this was the only test we did it would create a possible plethora of errors which would be exceedingly difficult to debug all at once. Since System Testing is simply the testing of the whole system at once, it would be a great assumption for one to make that the whole system works without any testing prior. Thus, given the size and scope of this project, Test Driven Development not only works, but will increase known functionality of the program.