

[Tyler Narmore](#) - tjn67
[Brannon Jordan](#) - bmj167
[Josh Possel](#) - jrp52
[Brandon Stone](#) - bs1443

GitHub: <https://github.com/TylerNarmore/SW-Testing-HW-1>

Test-Driven Development's defining characteristic is the code is written to only pass the test written at that point. The benefits that my team and I found from this was that it helped us think about all the edge cases for our code and to account for them while writing the actual program. We also thought that the testing phase for TDD development goes a lot smoother than usual because we had the tests written already and had less of a chance to have incompatible or incorrect tests since the code was built around the tests. The drawbacks for us were that it clashed with our normal coding styles. We weren't used to having strongly defined constraints on how we had to write our programs. In general, coding the modules after writing the tests took us longer when using TDD because we had to actively think about how it would run in our tests instead of how it was supposed to work. Overall, I believe we could all work with TDD if we had to, but we prefer to code using BDD.

Behavioral-Driven Development is when the program is written with how it will appear to the user in mind. We found this development style to be much friendlier to use. We tend to do this naturally. When combined with TDD, it was much more manageable than just using TDD by itself. We found that having some tests in mind was useful for transitioning from coding to testing. For example, while coding the retirement module, a section responsible for error checking input would never be able to run, and the testing helped find that error. This also meshed very well the unit testing aspect of the project. Having some tests in mind helped us make sure that every section of our code was tested and working correctly. When BDD, TDD, and unit testing are all used together, it's a very efficient process, however in general, we wouldn't want to apply for a TDD project because we feel that we wouldn't be able to work at our best performance.

Install Instructions

Install instructions are in the readme on the github page

Test Coverage

```
C:\WINDOWS\system32\cmd.exe

TestEmailVerifier.py .....
TestMenu.py .....
TestRetirement.py .....

----- coverage: platform win32, python 3.5.2-final-0 -----
Name                               Stmts  Miss  Cover
-----
Retirement.py                      51      0   100%
TestBMICalc.py                     23      0   100%
TestDistance.py                     10      0   100%
TestEmailVerifier.py               30      0   100%
TestMenu.py                         48      0   100%
TestRetirement.py                 28      0   100%
bmiCalculator.py                   45      0   100%
distanceForm.py                     5      0   100%
emailVerifier.py                   17      0   100%
main.py                             2      2     0%
menu.py                             53      3    94%
-----
TOTAL                               312      5    98%

===== 36 passed in 0.33 seconds =====
F:\Users\nitro_000\Documents\GitHub\SW-Testing-HW-1>
```