

## Assignment 2

The first thing I did was make sure that everything that passed and failed before passed and failed again. Immediately I ran into some worrisome errors. It turns out though that this was because there were undeclared identifiers and other errors that can now be found, but couldn't before, in my old test cases. Everything that failed in parse before though still failed in parse, except for the quotes with strings in them.

Next I built test cases that should pass. The first one was a giant monolithic test case that would pass without any faults. It had parallel scopes, within which errors would occur if parallel scope wasn't properly handled. It also has 5 levels deep of scope. The compiler did a fine job handling the parallel scopes along with the depth. The second passing test case I built was similar but also had places where warnings should be triggered. Once again it performed as expected.

The fail test cases were rather more complicated to build because the compiler now halts upon detection of any error. To start I built a few simpler test cases, cases 1, 2, 3, 4 and 5 which caught basic type mismatch errors in assignment and math. I then built more complex test cases, 6, 7, 8, 9 and 10. These test cases showed the same behavior performing across and within scope.

As of right now there are two known bugs. The first is that if a string with spaces does not have a space before the " such as `print("hey this doesn't work")` it won't work. The second bug is shown in test cases 8 and 10. This error is when there is an openbrace that has no children aside from another openbrace and then variable assignment is done. For example `{{int b}}` will not error but `{{int b \n b = 3}}`.