Name: *Katie Stevens*

Assignment: 3

Complete & Correct:	26 / 30
Tests:	9 / 10
Format and Comments:	5 / 5
Writeup:	5 / 5
Total Score:	45 / 50

Comments:

1. (-1) needed to include a good amount of unit tests to ensure parser working correctly on positive and negative cases (looking for a minimum of 10 but mwanted to see more like 20+)

2. (-4) failed tests (see below): 7, 13, 16, 19

Unit test output:

```
Loading:
Loading: 0 packages loaded
Analyzing: target //:parser-test (0 packages loaded, 0 targets configured)
INFO: Analyzed target //:parser-test (2 packages loaded, 134 targets configured).
INFO: Found 1 test target...
[0 / 6] [Prepa] BazelWorkspaceStatusAction stable-status.txt
[8 / 9] [Prepa] Testing //:parser-test
INFO: From Testing //:parser-test:
=================== Test output for //:parser-test:
JUnit4 Test Runner
...............
Time: 0.022

OK (15 tests)


BazelTestRunner exiting with a return value of 0
JVM shutdown hooks (if any) will run now.
The JVM will exit once they complete.

-- JVM shutdown starting at 2022-02-21 14:53:24 --

================================================================================
Target //:parser-test up-to-date:
  bazel-bin/parser-test.jar
  bazel-bin/parser-test
INFO: Elapsed time: 1.895s, Critical Path: 1.39s
INFO: 9 processes: 4 internal, 3 linux-sandbox, 2 worker.
INFO: Build completed successfully, 9 total actions
//:parser-test                                                   PASSED in 0.3s

Executed 1 out of 1 test: 1 test passes.
There were tests whose specified size is too big. Use the --test_verbose_timeout_warnings command line option to see
     which ones these are.
INFO: Build completed successfully, 9 total actions
```

Program test output:

```
0: bazel-bin/mypl --parse examples/parser.mypl

1: bazel-bin/mypl --parse <<< "var x"
PARSE_ERROR: Expecting fun, found 'var' at line 1, column 1

2: bazel-bin/mypl --parse <<< "fun f()"
PARSE_ERROR: Expecting ID, found '(' at line 1, column 6

3: bazel-bin/mypl --parse <<< "fun void f() {"
PARSE_ERROR: Expecting '}', found 'end-of-file' at line 2, column 1

4: bazel-bin/mypl --parse <<< "fun void f() {} var x"
PARSE_ERROR: Expecting fun, found 'var' at line 1, column 17

5: bazel-bin/mypl --parse <<< "fun void f(x, int y) {}"
PARSE_ERROR: Expecting id, found ',' at line 1, column 13

6: bazel-bin/mypl --parse <<< "fun void f(int x,) {}"
PARSE_ERROR: Expecting id, found ')' at line 1, column 18

7: bazel-bin/mypl --parse <<< "fun int f() { var x = }"

8: bazel-bin/mypl --parse <<< "fun int f() { var x int = 0}"
PARSE_ERROR: Expecting '=', found 'int' at line 1, column 21

9: bazel-bin/mypl --parse <<< "fun int f() { var x int = 0 + }"
PARSE_ERROR: Expecting '=', found 'int' at line 1, column 21

10: bazel-bin/mypl --parse <<< "fun int f() { x. = 0}"
PARSE_ERROR: Expecting id, found '=' at line 1, column 18

11: bazel-bin/mypl --parse <<< "fun int f() { if true { }"
PARSE_ERROR: Expecting '}', found 'end-of-file' at line 2, column 1

12: bazel-bin/mypl --parse <<< "fun int f() { if true {} else {} elif {} }"
PARSE_ERROR: Expecting '}', found 'elif' at line 1, column 34

13: bazel-bin/mypl --parse <<< "fun int f() { if x <= not {} }"

14: bazel-bin/mypl --parse <<< "fun int f() { var x = 3 + ( 2 * 3)) }"
LEXER_ERROR: missing decimal digit in double value '3' at line 1, column 33

15: bazel-bin/mypl --parse <<< "fun int f() { var x = 1 + not y.x ( 2 * 3) }"
PARSE_ERROR: Expecting '}', found '(' at line 1, column 35

16: bazel-bin/mypl --parse <<< "fun int f() { g(x,) }"

17: bazel-bin/mypl --parse <<< "fun int f() { x = new 'a' }"
PARSE_ERROR: Expecting id, found 'a' at line 1, column 23

18: bazel-bin/mypl --parse <<< "fun int f() { x = y.true }"
PARSE_ERROR: Expecting id, found 'true' at line 1, column 21

19: bazel-bin/mypl --parse <<< "fun int f() { x = delete x }"

20: bazel-bin/mypl --parse <<< "fun int f() { x = new x . y}"
PARSE_ERROR: Expecting '}', found '.' at line 1, column 25

>>> Done
```