

Project #5
CpSc 8270: Language Translation
Computer Science Division, Clemson University
Python Interpretation & Symbol Table
Brian Malloy, PhD
November 16, 2016

Due Date:

In order to receive credit for this assignment, your submission must be submitted, using the `web handin` command, by 8 AM, Tuesday, November 22nd of 2016. If you are unable to complete the project by the first due date, you may submit the project within three days after the due date with a ten point deduction.

Project Specification:

1. Incorporate the new scanner into your interpreter. You can find the new scanner in the course repo at: `8270Assets-2016/projects/5/newscanner`
2. Design and implement an Abstract Syntax Tree (AST) to represent and interpret your Python code.
3. Your solution should handle integer and float values and variables, `print`, `assignment`, and the same expressions as the previous project: $\{x + y, x - y, x * y, x/y, x//y, x\%y, x**e, (x), -x, +x\}$. Don't forget that Python uses floor for integer division, so that $-1/2$ is -1 , and $-1/2-1/2$ is also -1 .
4. To implement assignment you must build a symbol table. In addition to simple assignment to a variable, your solution should also interpret the following additional forms of assignment:
 $\{x+ = y, x- = y, x* = y, x/ = y, x// = y, x\% = y\}$.
5. In all cases, the oracle for correctness is a Python 2.7.n interpreter; that is, your expressions should evaluate, sans extended precision, to the same result that a Python 2.7.n interpreter would produce.
6. In the directory that contains your working interpreter, place a new directory titled `cases` that contains test cases that adequately test your interpreter.
7. Write a test harness, `test.py`, and place it in your project folder so that it runs the test cases in `cases`.
8. Your code should be well organized, formatted, readable, free of memory leaks, and exploit proper object orientation.

Consider the following sample execution:

```
malloy@riverwood:~/8270-2016/projects/5/code/soln$ r
x = 9
print x + 1
>>> 10
x = 6.0
print x/2
>>> 3.0
print 2**(0.5)
>>> 1.41421356237
x = -0.5
print 2**x
>>> 0.707106781187
```