

San José State University
Department of Computer Science

CS 153

Concepts of Compiler Design

Fall 2024
Instructor: Ron Mak

Assignment #4

Assigned: Thursday, September 19
Due: Thursday, October 3 at 11:59 PM
Team assignment, 150 points max

Pcl interpreter using ANTLR

The purpose of this assignment is to give you practice writing a parser and an interpreter for Pcl, a subset of Pascal. In effect, you will recreate the interpreter you wrote for Assignment #3, but this time you will use the ANTLR-generated lexer, parser, and backend `visit` methods.

The zip file [Asgn04Skeleton.zip](#) contains the `Pc14.g4` grammar file to which you must add production rules for the **WHILE**, **FOR**, **IF**, and **CASE** statements. Make any other modifications to the grammar file that you deem necessary. To confirm that your grammar is correct, use either the ANTLR plug-in or `rrd-antlr4-0.1.2.jar` on the command line (<https://github.com/bkiers/rrd-antlr4>) to generate syntax diagrams.

Generate a graphical parse tree for the test source file `TestCase.txt` using either the ANTLR plug-in or the command-line `grun` Java program. The grammar file and the graphical parse tree together tell you what child nodes each tree node has.

The skeleton `Executor` class demonstrates how the `visit` methods can use the context objects (the `ctx` parameters) to access children of the parse tree nodes. To complete the `Executor` class's `visitWhileStatement()`, `visitIfStatement()`, `visitForStatement()`, and `visitCaseStatement()` methods, you can use code from your solution to Assignment #3 or from the suggested solution. Look at the `Executor` class in the expression interpreter example [ExprLabeled.zip](#) for more examples of `visit` methods. Note that the parse tree created by the ANTLR-generated parser has a different structure than the one created by our hand-coded parser.

After you've completed the `Pc14.g4` grammar file and you've written the `visit` methods of the `Executor` class, you should be able to execute all the sample source files from Assignment #3. The runtime output should be the same as in that assignment.

What to submit to Canvas

A zip file that contains:

- All of your Java source files and any additional input test programs you wrote.
- Your `Pc14.g4` grammar file.
- A PDF(s) of your ANTLR-generated syntax diagrams.
- A PDF of your ANTLR-generated graphical parse tree from each of the test source files.
- Cut-and-paste text files of the runtime output from the following test source files: `TestWhile.txt`, `TestIf.txt`, `TestFor.txt`, and `TestCase.txt`.

Submit to **Assignment #4: Pcl Interpreter using ANTLR**

Rubric

Your submission will be graded according to these criteria:

Criteria	Max points
Pcl4 grammar that includes: <ul style="list-style-type: none">• <code>WHILE</code> statement• <code>IF</code> statement• <code>FOR</code> statement• <code>CASE</code> statement• syntax diagrams• parse tree for <code>TestCase.txt</code>	40 <ul style="list-style-type: none">• 5• 5• 10• 10• 5• 5
visit methods for: <ul style="list-style-type: none">• <code>WHILE</code>• <code>IF</code>• <code>FOR</code>• <code>CASE</code>	70 <ul style="list-style-type: none">• 15• 15• 20• 20
Execution output from: <ul style="list-style-type: none">• <code>TestWhile.txt</code>• <code>TestIf.txt</code>• <code>TestFor.txt</code>• <code>TestCase.txt</code>	40 <ul style="list-style-type: none">• 10• 10• 10• 10