San José State University
Department of Computer Science

CMPE 152
Compiler Design

Section 1
Fall 2020
Instructor: Ron Mak

Assignment #4

Assigned: Thursday, February 25
Due: Tuesday, March 9 at 2:30 PM
Team assignment, 120 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 Asgn04Cpp.zip contains the **`Pcl4.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 you 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, 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 ExprLabeledCpp.zip for 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 **`Pcl4.g4`** grammar file and the 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 C++ source files and any additional input test programs you wrote.
- Your **`Pcl4.g4`** grammar file.
- A PDF of your ANTLR-generated syntax diagrams (will require more than one page to fit).
- A PDF of your ANTLR-generated graphical parse tree from the **`TestCase.txt`** source file (will require more than one page to fit).
- Cut-and-paste text files of the runtime output from the following test source files from Assignment #3: **`HelloWorld.txt`**, **`TestWhile.txt`**, **`TestIf.txt`**, **`TextFor.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:	**30**
• **`WHILE`** statement	• 5
• **`IF`** statement	• 5
• **`FOR`** statement	• 5
• **`CASE`** statement	• 5
• syntax diagrams	• 5
• parse tree for **`TestCase.txt`**	• 5
`visit` **methods for:**	**65**
• assignment	• 5
• **`REPEAT`**	• **5**
• **`WHILE`**	• **5**
• **`IF`**	• **5**
• **`FOR`**	• **5**
• **`CASE`**	• **5**
• expression	• 5
• simple expression	• 5
• term	• 5
• parenthesized expression	• 5
• variable	• 5
• integer constant	• 5
• real constant	• 5
Execution output from:	**25**
• **`HelloWorld.txt`**	• **5**
• **`TestWhile.txt`**	• **5**
• **`TestIf.txt`**	• **5**
• **`TextFor.txt`**	• **5**
• **`TestCase.txt`**	• **5**