CS4850: L4850 Interpreter Project 1
Due Date: Monday, October 14, 2013 @ 8am

Purpose

This project is intended to give you experience constructing a parser and building an abstract syntax tree (AST). This project is intended to be completed using JavaCC and Java. However, if you wish to use some other language and parser generator, you may do so with my permission.

Project Summary

Your task is to construct a parser that accepts the L4850 programming language and constructs an abstract syntax tree of your design. Your code will emit the a textual representation of the grammar rules in the order in which the rule reductions occur.

The Scanner and Parser

You are to construct a scanner and parser using the JavaCC generator beginning with the skeleton grammar file provided. Then, you need to add actions to the parser rules to generate and abstract syntax tree. The abstract syntax tree need to be designed using good object-oriented design. Once the AST is constructed, you need to construct a print visitor that prints our the the orginal source program using indentation. You should try to match the input source format.

1. Read all of the documentation on JavaCC linked at `https://javacc.java.net`.
2. Download the zip file `L4850.zip` from E-Learning to your own directory and import the source files into your working project.
3. The JavaCC template file is in `srcs/parser/L485Grammar.jj`. The scanner portion contains a descriptions of all tokens in the L4850 language. A skeleton grammar has been provided.

Your Task

Encode the L4850 grammar found in the L4850 language description into `L485Grammar.jj`. Then, add actions for each rule to construct an abstract syntax tree. Then you are to use a visitor pattern to construct a print visitor to emit the original source from the AST.

Requirements

You may use any object-oriented language that you wish, although Java is the most advisable. You are to use either `make` or `ant` build files to construct an executable for your source.

Input. There are no hidden data for this project. All input is provided in the `Project1Input.zip` found on E-Learning.

Submission. Your code should be well-documented. You will submit all of your files in a zip file named `L4850.p1.zip` via E-Learing. Make sure you do a `'clean'` of your project before creating the zip file.