Parser Project Description

In this project you are required to implement a parser for Python language, using Context-free Grammar (CFG) with ANTLR (or something equivalent to ANTLR, for example, Bison, yacc, etc.). You are expected to write a grammar for Python for the parsing operation.

Tasks and requirements for the Parser

Please remember that we are not expecting a complete parser for Python. There will be specific features you are expected to parse of a Python language. You are expected to implement your parser for Python 3.x. To learn more about which features are expected to be implemented, please check section "Expected Features for your Python Parser". Once you checked the expected features for your parser, you must notice that there is no function or class implementation. Remember that the output should be a parse tree for the parser.

NOTE: Remember that Python is indentation sensitive!

Expected Features for your Python Parser

- `if`/`else` blocks
- Variable definitions
- `while` and `for` Loops
- Arithmetic operators (+, -, *, /, %, ^)
- Assignment operators (=, +=, -=, *=, /=, ^=, %=)
- Conditional statements (<, <=, >, >=, ==, !=, and, or, not)
- Support for comments
- **BONUS #1:** Syntax error message (this is where we did the accept/reject string. If the given code aka. grammar is not a Python language, reject it. In other words, throw a syntax error message).
- **BONUS #2:** Visualization of your parse tree. If you would like, you can use Graphviz.

Grading Criteria

- A working parser – 60%
  - (15 pts)`if`/`else` blocks
  - (15 pts) Variable definitions
  - (15 pts) `while` and `for` Loops
  - (15 pts) Arithmetic operators (+, -, *, /, %, ^)
  - (15 pts) Assignment operators (=, +=, -=, *=, /=, ^=, %=)
  - (15 pts) Conditional statements (<, <=, >, >=, ==, !=)
  - (10 pts) Comments
- Report (Writing README.md on your Github repo., see "Github Repository Page" section) – 10%
- Individual contribution – 30%
- **BONUS #1 –** 5 pts
- **BONUS #2 –** 10 pts

Rules & Recommendations

- You can have support from external libraries for writing grammars like **ANTLR**, **yacc** and **bison**, which we mentioned in our class before. There should be C, C++ and Java version of these libraries. The TAs shared a small example of how to use ANTLR with Python on Canvas.
- While evaluating and grading your projects, we will be using a web-based plagiarism tool to detect if you have got the code from somewhere else. Any code we detect that has been obtained from another repository, the overall Parser Project will be graded as 0.
- The deadline for the project is, **December 15th, 2020, 11:59pm**. You are also required to upload your projects as a zip file on Canvas for official grading. You can simply use the "Code → Download Zip" feature from Github and upload it on Canvas. While submitting your projects, please share your GitHub repository link as well. One of the team members would be enough to do this submission.
- Even though we will be reviewing your git commits for checking who contributed to the project, everyone should submit a text file that grades their group members (except themselves). Please write a short description on how your group member contributed to the project that supports your grade. The grade should be between 0-100.
- No presentation is required, but a demo video is required for your parser project. You can upload your video on Canvas along with your project, or you can upload it YouTube, but a link will be required. You can also embed the demo video into your README.md file on your GitHub repo. The video should present a working version and demonstration of the parser.
- Your project will be tested with the given sample python code (python_test_code.py). You can use the code for testing your parser.

Github Repository Page

- Use your README.md file.
    - Explain your project.
    - Include your team members.
    - Include the requirements that is required for your parser. For example, what are the steps for the setup, what environment is needed?
    - Write a "How to use/run" the parser.
    - **Optional:** Demo video