CMPSC461 Spring-2024
Programming Language Concepts
Instructor: Dr. Suman Saha

Project #2: Scope and Type Checking

Logistics

- **Filename:** The file name for the parser should be `project2_parser.py`.
- **Preparation:** Before starting the project, review the following:
  - The latest grammar file (`grammar.txt`).
  - The provided template parser file (`project2_parser.py`).
  - The test utility file (`test_utility.py`).
  - Watch the overview video for additional guidance.
- **Testing:** There are no hidden test cases. All test cases are provided in the `test_utility.py` file.
- **Scoring:** The total score for Project 2 is 100 points. Not all test cases are weighted equally.

Objective

This project is an extension of the previous project where we built a lexer and parser. In this project, we are focusing on adding scope and symbol table management, as well as type checking. The main aspects to consider are:

1. **Variable Declaration Before Use:** Variables must be declared before they are used.
2. **Type Consistency:** Expressions cannot involve variables or values of different types. For example, if `int x = 5` and `float z = y`, you cannot add `x` and `z`.
3. **Unique Variable Names in Scope:** Variables must have unique names within the same scope.

Your code should throw appropriate errors for these rules. Please refer to the `test_utility.py` file for examples.

Implementation Guide

Lexer

- We have provided a significant portion of the boilerplate code for the lexer to help you get started.
- You need to update the lexer to accommodate the new updates in the grammar.
- You can either use the provided boilerplate code and make the required changes, or if you have already worked on Project 1 and have your own code, update it to match the new requirements.

Parser

- For the parser, your functions should now return a node instead of a tuple as in Project 1.
- We have also provided the required code for the node structure to help you understand the changes.
- Update your parser functions to accommodate these changes and return nodes.
- Additionally, build a symbol table and complete the relevant functions required for type checking.