

Assignment 2 part A Report

Name: Jessica Vu

Organization of Programming Languages

Main features in files:

1. Scanner (`scan.cpp`):

- Read input character and breaks it into tokens like `read`, `write`, identifiers (e.g., `x`), literals (e.g., `42`), and arithmetic operators (e.g., `+`)

2. Parser (`parse.cpp`):

- Processes the token and verifies that it follows the expected grammar for a calculator-like language.
- Each function (e.g., `program()`, `stmt()`, `expr()`) corresponds to a rule in the grammar and prints predictions and matches during parsing.

3. Makefile:

- The Makefile has been updated to use the `g++` compiler instead of `gcc` for compiling the C++ code.
- A `test` target has been added to the Makefile that runs a series of sample calculator programs to test various aspects of the scanner and parser, including both valid and invalid inputs.

Key Differences from C to C++:

1. String Handling:

- Use **C++ strings** (`std::string`) instead of C-style character arrays (`char[]`) to store and manipulate token images.
- Functions like `strcpy` and `strcmp` have been replaced by C++ string operations.

2. Input/Output Handling:

- All input/output operations have been converted to use **C++ I/O** (`std::cin`, `std::cout`, `std::cerr`) instead of C functions (`scanf`, `printf`, `fprintf`).

3. Standard Libraries:

- C Libraries: `<stdio.h>`, `<string.h>`, `<stdlib.h>`
- C++ Libraries: `<iostream>`, `<string>`, `<cstdlib>`

Notes:

- To build the project: `make` and then `./parse`
- To run test cases in the Makefile: `make test`
- To clean the project: `make clean`