

## Project Idea

Create a programming language and compiler. Below are some formal grammar rules and syntax for the language, E++.

## Syntax

- Variable *shadowing* is **not** allowed. Having a variable with the same name in an inner scope is invalid, unless the variable is first declared in the inner scope.
- You must terminate a statement with a semicolon, *unless* that statement **ends** with a scope.
- Single line comments are denoted by `'//'` and Multi-line comments are denoted with `'/* */'`
- You must say **please** or **PLEASE** enough to satisfy the compiler.

## Producers

$$\begin{aligned}
 [\text{Program}] &\rightarrow [\text{Statement}]^* \\
 [\text{Statement}] &\rightarrow \begin{cases} \text{please} \\ \text{PLEASE} \\ \text{exit}([\text{Expr}]); \\ \text{set ID} = [\text{Expr}]; \\ \text{reset ID} = [\text{Expr}]; \\ \text{if}([\text{Expr}])[\text{Scope}][\text{AfterIf}] \\ [\text{Scope}] \end{cases} \\
 [\text{Scope}] &\rightarrow \{[\text{Stmt}]^*\} \\
 [\text{AfterIf}] &\rightarrow \begin{cases} \text{elseif}([\text{Expr}])[\text{Scope}][\text{AfterIf}] \\ \text{else}([\text{Expr}])[\text{Scope}] \\ \epsilon \end{cases} \\
 [\text{Expr}] &\rightarrow \begin{cases} [\text{Term}] \\ [\text{BinaryExpr}] \end{cases} \\
 [\text{BinaryExpr}] &\rightarrow \begin{cases} [\text{Expr}] \times [\text{Expr}] & \textit{precedence} = 1 \\ [\text{Expr}] \div [\text{Expr}] & \textit{precedence} = 1 \\ [\text{Expr}] \% [\text{Expr}] & \textit{precedence} = 1 \\ [\text{Expr}] + [\text{Expr}] & \textit{precedence} = 0 \\ [\text{Expr}] - [\text{Expr}] & \textit{precedence} = 0 \end{cases} \\
 [\text{Term}] &\rightarrow \begin{cases} \text{int\_lit} \\ \text{ID} \\ ([\text{Expr}]) \end{cases}
 \end{aligned}$$