Please note this is a language like C or Java where whitespaces have no meaning, and whitespace can be inserted between keywords, identifiers, constants, and specials to accommodate programmer style. This grammar does not include rules about whitespace because that would add immense clutter.

<prog> ::= **program** <decl-seq> **begin** <stmt-seq> **end** | **program begin** <stmt-seq> **end**

<decl-seq> ::= <decl> | <decl><decl-seq>

<stmt-seq> ::= <stmt> | <stmt><stmt-seq>

<decl> ::= <decl-int> | <decl-class>

<decl-int> ::= **int** <id-list> **;**

<decl-class> ::= **ref** <id-list> **;**

<id-list> ::= **id** | **id ,** <id-list>

<stmt> ::= <assign> | <if> | <loop> | <in> | <out> | <decl>

<assign> ::= **id =** <expr> **;** | **id = new ;** | **id = ref id ;**

<in> ::= **input id ;**

<out> ::= **output** <expr> **;**

<if> ::= **if** <cond> **then** <stmt-seq> **endif**

| **if** <cond> **then** <stmt-seq> **else** <stmt-seq> **endif**

<loop> ::= **while** <cond> **begin** <stmt-seq> **endwhile**

<cond> ::= <cmpr> | **! (** <cond> **)**

| <cmpr> **or** <cond>

<cmpr> ::= <expr> **==** <expr> | <expr> **<** <expr>

| <expr> **<=** <expr>

<expr> ::= <term> | <term> **+** <expr> | <term> **–** <expr>

<term> ::= <factor> | <factor> ***** <term>

<factor> ::= **id** | **const** | **(** <expr> **)**