COP5556 Assignment 2

Using your (corrected) Scanner from Assignment 1, implement a simpler recursive descent parser for the programming language with the following context-free grammar:

|  |
| --- |
| Program ::= ( Declaration SEMI | Statement SEMI )\* |
| Declaration :: = VariableDeclaration | ImageDeclaration |
| VariableDeclaration ::= VarType IDENT ( ASSIGN Expression | ϵ ) |
| VarType ::= KW\_int | KW\_string |
| ImageDeclaration ::= KW\_image (LSQUARE Expression COMMA Expression RSQUARE | ϵ) IDENT (((LARROW | ASSIGN ) Expression) | ϵ ) added missing ‘(‘ 10/5 |
| Statement ::= AssignmentStatement | ImageOutStatement | ImageInStatement | LoopStatement |
| ImageOutStatement ::= IDENT RARROW Expression | IDENT RARROW KW\_SCREEN ( LSQUARE Expression COMMA Expression RSQUARE | ϵ ) |
| ImageInStatement ::= IDENT LARROW Expression |
| AssignmentStatement ::= IDENT ASSIGN Expression |
| LoopStatement ∷= IDENT ASSIGN STAR ConstXYSelector COLON (Expression | ϵ ) COLON Expression |
| Expression ::= OrExpression Q Expression COLON Expression | OrExpression |
| OrExpression ::= AndExpression ( OR AndExpression)\* |
| AndExpression ::= EqExpression ( AND EqExpression )\* |
| EqExpression ::= RelExpression ( (EQ | NEQ ) RelExpression )\* |
| RelExpression ::= AddExpression ( ( LT | GT | LE | GE ) AddExpression)\* |
| AddExpression ::= MultExpression ( (PLUS | MINUS ) MultExpression )\* |
| MultExpression := UnaryExpression ( ( STAR | DIV | MOD ) UnaryExpression )\* (Corrected 9/26, Changed TIMES to STAR) |
| UnaryExpression ::= (PLUS | MINUS) UnaryExpression | UnaryExpressionNotPlusMinus |
| UnaryExpressionNotPlusMinus ::= EXCL UnaryExpression | HashExpression |
| HashExpression ∷= Primary ( HASH Attribute)\* |
| Primary ::= (INTLIT | IDENT | LPAREN Expression RPAREN | STRINGLIT | KW\_X | KW\_Y | CONSTANT |PixelConstructor | ArgExpression ) (PixelSelector | ϵ )  (Corrected to add alternative IDENT 9/27, Changed treatment of PixelSelector 9/30) |
| PixelConstructor ∷= LPIXEL Expression COMMA Expression COMMA Expression RPIXEL |
| PixelSelector ∷= LSQUARE Expression COMMA Expression RSQUARE (Corrected 9/30, deleted Expression) |
| Attribute ∷= KW\_WIDTH | KW\_HEIGHT | KW\_RED | KW\_GREEN | KW\_BLUE |
| ArgExpression ∷= AT Primary (Corrected 9/27, changed Expression to Primary) |
| ConstXYSelector ::= LSQUARE KW\_X COMMA KW\_Y RSQUARE |

* If an illegal sentence is encountered, your parser should throw a SyntaxException. The token where the error was manifested and an error message are parameters of the SyntaxException constructor. The contents of the message will not be graded, but you will appreciate it later if it is informative.
* If the given input is legal in the language, the parser simply returns normally from the parse() method.