

ProgramNode  $\rightarrow$  Statements

Statements  $\rightarrow$  StatementNode Statements  $\mid \epsilon$

StatementNode  $\rightarrow$  VarDeclNode  $\mid$  AssignNode  $\mid$  CompoundAssignNode  $\mid$  UnaryOpNode  $\mid$

IfElseNode  $\mid$  LoopNode  $\mid$  PrintNode  $\mid$  BlockNode  $\mid$  TryCatchNode  $\mid$  MatchNode

VarDeclNode  $\rightarrow$  type identifier VarDeclTail

VarDeclTail  $\rightarrow$  "," identifier VarDeclTail  $\mid$  "=" ExpressionNode ExprList  $\mid$  ";"  $\mid \epsilon$

ExprList  $\rightarrow$  "," ExpressionNode ExprList  $\mid \epsilon$

type  $\rightarrow$  "int"  $\mid$  "float"  $\mid$  "bool"  $\mid$  "char"  $\mid$  "string"  $\mid$  "array"

AssignNode  $\rightarrow$  identifier "=" ExpressionNode ";"  $\mid$  BinaryOpNode<INDEX> "=" ExpressionNode  
","

CompoundAssignNode  $\rightarrow$  identifier CompoundOp ExpressionNode ";"  $\mid$  BinaryOpNode<INDEX>

CompoundOp ExpressionNode ";"

UnaryOpNode  $\rightarrow$  identifier UnaryOp ";"  $\mid$  BinaryOpNode<INDEX> UnaryOp ";"

CompoundOp  $\rightarrow$  "+="  $\mid$  "-="  $\mid$  "\*="  $\mid$  "/="  $\mid$  "%="

UnaryOp  $\rightarrow$  "++"  $\mid$  "--"

IfElseNode  $\rightarrow$  "if" "(" ExpressionNode ")" BlockNode ElseClause

ElseClause  $\rightarrow$  "else" ElseBlock  $\mid \epsilon$

ElseBlock  $\rightarrow$  BlockNode  $\mid$  IfElseNode

LoopNode  $\rightarrow$  ForLoopNode  $\mid$  ForeachLoopNode

ForLoopNode  $\rightarrow$  "for" "(" VarDeclNode ";" ExpressionNode ";" AssignNode ")" BlockNode  $\mid$  "for"  
"(" VarDeclNode ";" ";" AssignNode ")" BlockNode  $\mid$  "for" "(" VarDeclNode ";" ExpressionNode ";"  
")" BlockNode  $\mid$  "for" "(" VarDeclNode ";" ";" ")" BlockNode  $\mid$  "for" "(" AssignNode ";"  
ExpressionNode ";" AssignNode ")" BlockNode  $\mid$  "for" "(" AssignNode ";" ";" AssignNode ")"  
BlockNode  $\mid$  "for" "(" AssignNode ";" ExpressionNode ";" ")" BlockNode  $\mid$  "for" "(" AssignNode ";"  
";" ")" BlockNode  $\mid$  "for" "(" ";" ExpressionNode ";" AssignNode ")" BlockNode  $\mid$  "for" "(" ";" ";"  
AssignNode ")" BlockNode  $\mid$  "for" "(" ";" ExpressionNode ";" ")" BlockNode  $\mid$  "for" "(" ";" ";" ")"  
BlockNode

ForeachLoopNode  $\rightarrow$  "foreach" "(" identifier "in" ExpressionNode ")" BlockNode

PrintNode  $\rightarrow$  "print" "(" ExpressionNode ")" ";"

BlockNode  $\rightarrow$  "{" Statements "}"

TryCatchNode  $\rightarrow$  "try" BlockNode "catch" "(" "error" identifier ")" BlockNode

MatchNode  $\rightarrow$  "match" ExpressionNode "{" MatchCases "}"  
 MatchCases  $\rightarrow$  MatchCaseNode MatchCaseTail |  $\epsilon$   
 MatchCaseTail  $\rightarrow$  "," MatchCaseNode MatchCaseTail | MatchCaseNode MatchCaseTail |  $\epsilon$   
 MatchCaseNode  $\rightarrow$  ExpressionNode "->" MatchBody | "\_" "->" MatchBody  
 MatchBody  $\rightarrow$  BlockNode | StatementNode  
 ExpressionNode  $\rightarrow$  PrimaryNode BinaryExprTail  
 BinaryExprTail  $\rightarrow$  binary-op PrimaryNode BinaryExprTail | "?" ExpressionNode ":"  
 ExpressionNode |  $\epsilon$   
 PrimaryNode  $\rightarrow$  IntLiteralNode | FloatLiteralNode | BoolLiteralNode | CharLiteralNode |  
 StrLiteralNode | VarRefNode | ArrayLiteralNode | BinaryOpNode<INDEX> | ConcatNode |  
 BinaryOpNode<ABS> | BinaryOpNode<POW> | BinaryOpNode<METHOD\_CALL> |  
 UnaryOpNode<NEGATE> | UnaryOpNode<LENGTH> | UnaryOpNode<MIN> |  
 UnaryOpNode<MAX> | BinaryOpNode<MULTIPLY\_ARRAY> | BinaryOpNode<ADD\_ARRAY> |  
 BinaryOpNode<SUBTRACT\_ARRAY> | BinaryOpNode<DIVIDE\_ARRAY> | "(" ExpressionNode  
 ")"  
 binary-op  $\rightarrow$  "+" | "-" | "\*" | "/" | "%" | "==" | "!=" | "<" | ">" | "<=" | ">=" | "&&" | "||" | "^"  
 IntLiteralNode  $\rightarrow$  digitlist | "+" digitlist | "-" digitlist  
 FloatLiteralNode  $\rightarrow$  digitlist "." digitlist | "+" digitlist "." digitlist | "-" digitlist "." digitlist  
 BoolLiteralNode  $\rightarrow$  "true" | "false"  
 CharLiteralNode  $\rightarrow$  "" character ""  
 StrLiteralNode  $\rightarrow$  "" characters ""  
 digitlist  $\rightarrow$  digit digitlist | digit  
 characters  $\rightarrow$  character characters |  $\epsilon$   
 VarRefNode  $\rightarrow$  identifier  
 ArrayLiteralNode  $\rightarrow$  "[" ArrayElements "]" | "[" "]"  
 ArrayElements  $\rightarrow$  ExpressionNode ArrayElementsTail  
 ArrayElementsTail  $\rightarrow$  "," ExpressionNode ArrayElementsTail |  $\epsilon$   
 BinaryOpNode<INDEX>  $\rightarrow$  identifier "[" ExpressionNode "]"  
 ConcatNode  $\rightarrow$  "concat" "(" ExpressionNode "," ExpressionNode ")"  
 BinaryOpNode<ABS>  $\rightarrow$  "abs" "(" ExpressionNode ")"  
 BinaryOpNode<POW>  $\rightarrow$  "pow" "(" ExpressionNode "," ExpressionNode ")"

BinaryOpNode<METHOD\_CALL> → identifier "." identifier "("

UnaryOpNode<NEGATE> → "-" ExpressionNode "

UnaryOpNode<LENGTH> → "length" "(" ExpressionNode "

UnaryOpNode<MIN> → "min" "(" ExpressionNode "

UnaryOpNode<MAX> → "max" "(" ExpressionNode "

BinaryOpNode<MULTIPLY\_ARRAY> → "multiply" "(" ExpressionNode "," ExpressionNode "

BinaryOpNode<ADD\_ARRAY> → "add" "(" ExpressionNode "," ExpressionNode "

BinaryOpNode<SUBTRACT\_ARRAY> → "subtract" "(" ExpressionNode "," ExpressionNode "

BinaryOpNode<DIVIDE\_ARRAY> → "divide" "(" ExpressionNode "," ExpressionNode "

identifier → letter identifier\_tail

identifier\_tail → letter identifier\_tail | digit identifier\_tail | "\_" identifier\_tail | ε