

```

CompilationUnit → Imports Unit
Imports → Import Imports |
Import → import ID IDMore ;
IDMore → ID IDMore |

Unit → Program | Class
Program → program ID Body
Class → class IDClass extends IDClass Body
Body → { Declarations Block Declarations }

Declarations → Declaration Declarations |
Declaration → Qualification MethodDeclaration
Declaration → Intent Binding ObjectDeclaration
Qualification → Access Intent
Access → public | protected | private |
Intent → static |
Binding → final |

MethodDeclaration → Function | Procedure
Function → IDClass IDFunction FormalList Block
Procedure → IDProcedure FormalList Block

FormalList → { Formals }
Formals → Formal FormalsMore |
FormalsMore → Formal FormalsMore |
Formal → IDClass ID

Block → { StatementList }
StatementList → Statement StatementList
StatementList → return Expression ; |
Statement → Binding ObjectDeclaration | Instruction ; | If | While

ObjectDeclaration → Formal Initialization ;
Initialization → Instantiation ArgumentList |
Instantiation → new | set

Instruction → ID InstructionMethod Invocation | Call
InstructionMethod → Instantiation | MethodID
MethodID → ID | IDSR
Invocation → ArgumentList CallsMore
ArgumentList → { Actuals }
Call → MethodID ArgumentList
CallsMore → Calls |
Calls → Call CallsMore

Actuals → Expression ActualsMore |
ActualsMore → Expression ActualsMore |
Expression → Term ExpressionMore
ExpressionMore → OperatorID Term ExpressionMore |
Term → { Expression } | Item
Item → Literal | Operation | ID
Operation → ID . MethodID Invocation | Call

If → if { Condition } Block Else
Else → else Block |
While → while { Condition } Block
Condition → Expression

IDClass → ID | IDG
IDFunction → ID | IDSD
IDProcedure → new | IDFunction
Literal → BooleanLiteral | CharacterLiteral | StringLiteral | IntegerLiteral | RealLiteral
```

Deleted: ¶

Formatted: Highlight

Deleted: ¶

Semantic and Code Generation Note: Condition is separate from Expression to allow for objects to be checked to make sure that they are not "null references" as part of evaluating the condition.

Deleted: ¶

Lexeme	Token #	
-7	-8	Real Literal
-6	-7	Integer Literal
-5	-6	String Literal
-4	-5	Character Literal
-3	-4	Identifier-Synonymous Definition (IDSD); uses “ ” to delimit synonyms
-2	-3	Identifier-Synonymous Reference (IDSR)
	-2	Identifier-Generic Notation (IDG); contains “< anything >”
	-1	Identifier (ID); excludes all Java reserved words
	0	<i>Indicates end of line in source program</i>
class	1	
else	2	
extends	3	
false	4	
final	5	
if	6	
import	7	
new	8	
private	9	
program	10	
protected	11	
public	12	
return	13	
set	14	
static	15	
true	16	
while	17	
(18	
)	19	
{	20	
}	21	
,	22	
;	23	
.	24	
	-11	Invalid Lexeme – Malformed Character Literal
	-12	Invalid Lexeme - Extraneous ‘\’ or ‘ ’
	-13	Invalid Lexeme – Malformed Binary Integer Literal
	-14	Invalid Lexeme – Malformed Octal Integer Literal
	-15	Invalid Lexeme – Malformed Hex Integer Literal
	-16	Invalid Lexeme – Malformed Binary, Octal or Hex Integer Literal
	-17	Invalid Lexeme – Malformed Real Literal
	-18	Invalid Lexeme – Extraneous Character ~!@#%&*=[]:<>/?
	-19	Invalid Lexeme – Malformed Synonymous Identifier
	-98	Comment; /* */ form
	-99	Comment; // form