

Cameron Palmer, Programming Languages, February 2007

Jeff Grammar

Program -> FunctionsList
FunctionsList -> Function | FunctionsList
Function -> Name (ParameterList) Block
ParameterList -> Name // C Style function call with no parameters
ParameterList -> Name , ParameterList
Block -> begin StatementList end
Assignment -> Identifier [[Expression]] = Expression\n // Array notation included
StatementList -> Statement StatementList
Statement -> \n || Block || Assignment || ForStatement || IfStatement || WhileStatement ||
BreakStatement || ReturnStatement
SwitchStatement -> switch (Identifier) CaseStatements
CaseStatements -> CaseStatement CaseStatements || CaseStatement
CaseStatement -> case Literal : { StatementList || BreakStatement }
ForStatement -> for (initialize ; condition ; increment) Block
IfStatement -> if (Expression) Block [else Block]
WhileStatement -> while (Expression) Block
BreakStatement -> break
ReturnStatement -> return statement

Expression -> Conjunction { || Conjunction }
Conjunction -> Equality { && Equality }
Equality -> Relation [EquOp Relation]
EquOp -> == | !=
Relation -> Addition [RelOp Addition]
RelOp -> < || <= || > || >=
Addition -> Term { AddOp Term }
AddOp -> + || -
Term -> Factor { MulOp Factor }
MulOp -> * || / || %
Factor -> [UnaryOp] Primary
UnaryOp -> - || !
Primary -> Identifier [[Expression]] || Literal || (Expression) || Type (Expression)

Identifier -> Letter { Letter | Digit }
Letter -> a-z || A-Z
Digit -> 0-9
Literal -> Integer || Boolean || Float || Char
Integer -> Digit { Digit }
Boolean -> true || false
Float -> Integer . Integer
Char -> 'ASCIIChar'