

# CS75: Compiler Construction, Spring 2007

## C-- Programming Language Specification

<i>Program</i>	→	<i>VarDeclList FunDeclList</i>
<i>VarDeclList</i>	→	$\epsilon$ <i>VarDecl VarDeclList</i>
<i>VarDecl</i>	→	<i>Type id</i> ; <i>Type id [ num ]</i> ;
<i>FunDeclList</i>	→	<i>FunDecl</i> <i>FunDecl FunDeclList</i>
<i>FunDecl</i>	→	<i>Type id ( ParamDeclList ) Block</i>
<i>ParamDeclList</i>	→	$\epsilon$ <i>ParamDeclListTail</i>
<i>ParamDeclListTail</i>	→	<i>ParamDecl</i> <i>ParamDecl, ParamDeclListTail</i>
<i>ParamDecl</i>	→	<i>Type id</i> <i>Type id [ ]</i>
<i>Block</i>	→	{ <i>VarDeclList StmtList</i> }
<i>Type</i>	→	<b>int</b> <b>char</b>
<i>StmtList</i>	→	<i>Stmt</i> <i>Stmt StmtList</i>
<i>Stmt</i>	→	; <i>Expr</i> ; <b>return</b> <i>Expr</i> ; <b>read id</b> ; <b>write</b> <i>Expr</i> ; <b>writeln</b> ; <b>break</b> ; <b>if</b> ( <i>Expr</i> ) <i>Stmt</i> <b>else</b> <i>Stmt</i> <b>while</b> ( <i>Expr</i> ) <i>Stmt</i> <i>Block</i>
<i>Expr</i>	→	<i>Primary</i> <i>UnaryOp Expr</i> <i>Expr BinOp Expr</i> <b>id</b> = <i>Expr</i> <b>id</b> [ <i>Expr</i> ] = <i>Expr</i>
<i>Primary</i>	→	<b>id</b> <b>num</b> ( <i>Expr</i> ) <b>id</b> ( <i>ExprList</i> ) <b>id</b> [ <i>Expr</i> ]
<i>ExprList</i>	→	$\epsilon$ <i>ExprListTail</i>
<i>ExprListTail</i>	→	<i>Expr</i> <i>Expr, ExprListTail</i>
<i>UnaryOp</i>	→	-   !
<i>BinOp</i>	→	+   -   *   / ==   !=   <   <=   >   >= &&