

$$\begin{aligned}
\langle \text{program} \rangle &\rightarrow \langle \text{roots} \rangle \\
\langle \text{roots} \rangle &\rightarrow \langle \text{roots} \rangle \\
&\quad | \quad \langle \text{root} \rangle \langle \text{roots} \rangle \\
\langle \text{root} \rangle &\rightarrow \langle \text{function} \rangle \\
&\quad | \quad \langle \text{dcl} \rangle; \\
\langle \text{dcl} \rangle &\rightarrow \langle \text{type} \rangle \langle \text{id} \rangle \mid \langle \text{type} \rangle \langle \text{assign} \rangle \\
\langle \text{stmts} \rangle &\rightarrow \langle \text{stmt} \rangle \\
&\quad | \quad \langle \text{stmt} \rangle \langle \text{stmts} \rangle \\
\langle \text{stmt} \rangle &\rightarrow \langle \text{assign} \rangle; \\
&\quad | \quad \langle \text{if} \rangle \\
&\quad | \quad \langle \text{while} \rangle \\
&\quad | \quad \langle \text{from} \rangle \\
&\quad | \quad \langle \text{empty} \rangle \\
&\quad | \quad \langle \text{dcl} \rangle; \\
&\quad | \quad \langle \text{functioncall} \rangle; \\
&\quad | \quad \langle \text{switch} \rangle \\
\langle \text{id} \rangle &\rightarrow \langle \text{letter} \rangle \\
&\quad | \quad \langle \text{id} \rangle \langle \text{letter} \rangle \\
&\quad | \quad \langle \text{id} \rangle \langle \text{digit} \rangle \\
\langle \text{letter} \rangle &\rightarrow [\text{a-z}] \\
&\quad | \quad [\text{A-Z}] \\
\langle \text{digit} \rangle &\rightarrow [0-9] \\
\langle \text{assign} \rangle &\rightarrow \langle \text{id} \rangle \leftarrow \langle \text{expr} \rangle \\
\langle \text{expr} \rangle &\rightarrow \langle \text{expr} \rangle + \langle \text{term} \rangle \\
&\quad | \quad \langle \text{expr} \rangle - \langle \text{term} \rangle \\
&\quad | \quad \langle \text{term} \rangle \\
\langle \text{term} \rangle &\rightarrow \langle \text{term} \rangle * \langle \text{factor} \rangle \\
&\quad | \quad \langle \text{term} \rangle / \langle \text{factor} \rangle \\
&\quad | \quad \langle \text{factor} \rangle \\
\langle \text{factor} \rangle &\rightarrow (\langle \text{expr} \rangle) \\
&\quad | \quad \langle \text{id} \rangle \\
\langle \text{switch} \rangle &\rightarrow \text{switch } (\langle \text{expr} \rangle) \text{ begin } \langle \text{cases} \rangle \text{ end} \\
\langle \text{cases} \rangle &\rightarrow \text{case } \langle \text{expr} \rangle : \langle \text{stmts} \rangle \langle \text{endcase} \rangle
\end{aligned}$$

$\langle \text{endcase} \rangle \rightarrow \langle \text{cases} \rangle$
| break;
| break; $\langle \text{cases} \rangle$
| default: $\langle \text{stmts} \rangle$ break;
| break; default: $\langle \text{stmts} \rangle$ break;

 $\langle \text{from} \rangle \rightarrow \text{from } \langle \text{expr} \rangle \text{ to } \langle \text{logexpr} \rangle \text{ step } \langle \text{expr} \rangle \text{ begin } \langle \text{stmts} \rangle \text{ end}$

 $\langle \text{while} \rangle \rightarrow \text{while}(\langle \text{logexpr} \rangle) \text{ begin } \langle \text{stmts} \rangle \text{ end}$

 $\langle \text{if} \rangle \rightarrow \text{if}(\langle \text{logexpr} \rangle) \text{ begin } \langle \text{stmts} \rangle \langle \text{endif} \rangle$

 $\langle \text{endif} \rangle \rightarrow \text{end else } \langle \text{if} \rangle$
| end else begin $\langle \text{stmts} \rangle$ end
| end

 $\langle \text{logexpr} \rangle \rightarrow \langle \text{logexpr} \rangle \text{ OR } \langle \text{andcomp} \rangle$
| $\langle \text{andcomp} \rangle$

 $\langle \text{andcomp} \rangle \rightarrow \langle \text{andcomp} \rangle \text{ AND } \langle \text{comp} \rangle$
| $\langle \text{comp} \rangle$

 $\langle \text{comp} \rangle \rightarrow \langle \text{boolean-operand} \rangle \langle \text{comparison-operator} \rangle \langle \text{boolean-operand} \rangle$

 $\langle \text{boolean-operand} \rangle \rightarrow \text{true}$
| false
| $\langle \text{expr} \rangle$
| $\langle \text{boolean} \rangle$

 $\langle \text{boolean} \rangle \rightarrow \neg(\langle \text{logexpr} \rangle)$
| $(\langle \text{logexpr} \rangle)$

 $\langle \text{comparison-operator} \rangle \rightarrow >$
| <
| <=
| >=
| !=
| =

 $\langle \text{type} \rangle \rightarrow \langle \text{primitive-type} \rangle$
| $\langle \text{array-type} \rangle$

 $\langle \text{primitive-type} \rangle \rightarrow \langle \text{numeric-type} \rangle$
| bool

 $\langle \text{numeric-type} \rangle \rightarrow \langle \text{integral-type} \rangle$
| double

$\langle integral\text{-}type \rangle \rightarrow \text{int}$
 | char

$\langle array\text{-}type \rangle \rightarrow \langle type \rangle []$

$\langle function \rangle \rightarrow \text{function } \langle id \rangle \text{ returns } \langle type \rangle \text{ using } (\langle using \rangle) \text{ begin } \langle stmts \rangle \text{ return } \langle expr \rangle \text{ end}$
 | $\text{function} \rightarrow \text{function } \langle id \rangle \text{ returns nothing using } (\langle using \rangle) \text{ begin } \langle stmts \rangle \text{ end}$

$\langle using \rangle \rightarrow \langle subusing \rangle$
 | $\langle empty \rangle$

$\langle subusing \rangle \rightarrow \langle type \rangle \langle id \rangle, \langle subusing \rangle$
 | $\langle type \rangle \langle id \rangle$

$\langle functioncall \rangle \rightarrow \text{call } \langle id \rangle (\langle using \rangle)$