

# Minimal++ Grammar Rules

- $\langle \text{program} \rangle ::= \text{program id } \{ \langle \text{block} \rangle \}$
- $\langle \text{block} \rangle ::= \langle \text{declarations} \rangle \langle \text{subprograms} \rangle \langle \text{statements} \rangle$
- $\langle \text{declarations} \rangle ::= (\text{declare } \langle \text{varlist} \rangle ;)^*$
- $\langle \text{varlist} \rangle ::= \epsilon \mid \text{id } ( , \text{id } )^*$
- $\langle \text{subprograms} \rangle ::= (\langle \text{subprogram} \rangle)^*$
- $\langle \text{subprogram} \rangle ::= \text{function id } \langle \text{funcbody} \rangle \mid \text{procedure id } \langle \text{funcbody} \rangle$
- $\langle \text{funcbody} \rangle ::= \langle \text{formalpars} \rangle \{ \langle \text{block} \rangle \}$
- $\langle \text{formalpars} \rangle ::= ( \langle \text{formalparlist} \rangle )$
- $\langle \text{formalparlist} \rangle ::= \langle \text{formalparitem} \rangle ( , \langle \text{formalparitem} \rangle )^* \mid \epsilon$
- $\langle \text{formalparitem} \rangle ::= \text{in id} \mid \text{inout id}$
- $\langle \text{statements} \rangle ::= \langle \text{statement} \rangle \mid \{ \langle \text{statement} \rangle ( ; \langle \text{statement} \rangle )^* \}$
- $\langle \text{statement} \rangle ::= \langle \text{assignment-stat} \rangle \mid \langle \text{if-stat} \rangle \mid \langle \text{while-stat} \rangle \mid$   
 $\langle \text{doublewhile-stat} \rangle \mid \langle \text{loop-stat} \rangle \mid \langle \text{exit-stat} \rangle \mid \langle \text{forcase-stat} \rangle \mid \langle \text{incase-stat} \rangle \mid \langle \text{call-stat} \rangle \mid$   
 $\langle \text{return-stat} \rangle \mid \langle \text{input-stat} \rangle \mid \langle \text{print-stat} \rangle$
- $\langle \text{assignment-stat} \rangle ::= \text{id } := \langle \text{expression} \rangle$
- $\langle \text{if-stat} \rangle ::= \text{if } (\langle \text{condition} \rangle) \text{ then } \langle \text{statements} \rangle \langle \text{elsepart} \rangle$
- $\langle \text{elsepart} \rangle ::= \epsilon \mid \text{else } \langle \text{statements} \rangle$

- $\langle \text{while-stat} \rangle ::= \text{while } (\langle \text{condition} \rangle) \langle \text{statements} \rangle$
- $\langle \text{doublewhile-stat} \rangle ::= \text{doublewhile } (\langle \text{condition} \rangle) \langle \text{statements} \rangle$
- $\text{else } \langle \text{statements} \rangle$
- $\langle \text{loop-stat} \rangle ::= \text{loop } \langle \text{statements} \rangle$
- $\langle \text{exit-stat} \rangle ::= \text{exit}$
- $\langle \text{forcase-stat} \rangle ::= \text{forcase}$   
 $( \text{ when } (\langle \text{condition} \rangle) : \langle \text{statements} \rangle )^*$   
 $\text{default: } \langle \text{statements} \rangle$
- $\langle \text{incase-stat} \rangle ::= \text{incase}$   
 $( \text{ when } (\langle \text{condition} \rangle) : \langle \text{statements} \rangle )^*$
- $\langle \text{return-stat} \rangle ::= \text{return } \langle \text{expression} \rangle$
- $\langle \text{call-stat} \rangle ::= \text{call id } \langle \text{actualpars} \rangle$
- $\langle \text{print-stat} \rangle ::= \text{print } (\langle \text{expression} \rangle)$
- $\langle \text{input-stat} \rangle ::= \text{input } (\text{id})$
- $\langle \text{actualpars} \rangle ::= ( \langle \text{actualparlist} \rangle )$
- $\langle \text{actualparlist} \rangle ::= \langle \text{actualparitem} \rangle ( , \langle \text{actualparitem} \rangle )^* | \epsilon$
- $\langle \text{actualparitem} \rangle ::= \text{in } \langle \text{expression} \rangle | \text{inout id}$
- $\langle \text{condition} \rangle ::= \langle \text{boolterm} \rangle (\text{or } \langle \text{boolterm} \rangle )^*$
- $\langle \text{boolterm} \rangle ::= \langle \text{boolfactor} \rangle (\text{and } \langle \text{boolfactor} \rangle )^*$

- $\langle \text{boolfactor} \rangle ::= \text{not } [\langle \text{condition} \rangle] \mid [\langle \text{condition} \rangle] \mid$   
 $\langle \text{expression} \rangle \langle \text{relational-oper} \rangle \langle \text{expression} \rangle$
- $\langle \text{expression} \rangle ::= \langle \text{optional-sign} \rangle \langle \text{term} \rangle ( \langle \text{add-oper} \rangle \langle \text{term} \rangle )^*$
- $\langle \text{term} \rangle ::= \langle \text{factor} \rangle ( \langle \text{mul-oper} \rangle \langle \text{factor} \rangle )^*$
- $\langle \text{factor} \rangle ::= \text{constant} \mid ( \langle \text{expression} \rangle ) \mid \text{id} \langle \text{idtail} \rangle$
- $\langle \text{idtail} \rangle ::= \epsilon \mid \langle \text{actualpars} \rangle$
- $\langle \text{relational-oper} \rangle ::= = \mid < = \mid > = \mid > \mid < \mid < >$
- $\langle \text{add-oper} \rangle ::= + \mid -$
- $\langle \text{mul-oper} \rangle ::= * \mid /$
- $\langle \text{optional-sign} \rangle ::= \epsilon \mid \langle \text{add-oper} \rangle$

COPYRIGHT ©

This document is a translation of the original  
assignment handout for the course 'Compilers',  
written and owned by George Manis.  
Department of Computer Science And Engineering.  
Polytechnic School of the University of Ioannina.  
February 2020