**While-language grammar** ( *N\** denotes 0, 1 or more repetitions of *N* )

*Program* → *Command\* ProcDecl\**

*Command* → **{** *Command\** **}**

→ *Var* **=** *Exp* **;**

→ **if (** *Exp* **)** *Command* **else** *Command*

→ **while (** *Exp* **)** *Command*

→ *id* **(** *ExpList* *OutActuals* **) ;**

*Exp* → *PrimaryExp* *op* *PrimaryExp*

→ *PrimaryExp*

*PrimaryExp* → *INTEGER\_LITERAL*

→ **true**

→ **false**

→ *Var*

→ **!** *PrimaryExp*

→ **(** *Exp* **)**

*Var*→ *id*

*ExpList* → *Exp ExpRest\** **;**

→

*ExpRest* → **,** *Exp*

*ProcDecl* → **proc** *id* **(** *InFormals OutFormals* **)** **{** *Command\** **}**

*InFormals* → **in** *VarList* **;**

→

*OutFormals* → **out** *VarList*

→

*OutActuals* → *VarList*

→

*VarList* → *id VarRest\**

*VarRest* → **,** *id*

Additional terminal symbol definitions:

*op* is one of the binary operators: **&& || < > == / + - \***

*id* is a sequence of letters, digits and underscores, starting with a letter.

*INTEGER\_LITERAL* is a non-empty sequence of decimal digits, optionally preceded by a minus sign.

Comments: these can either be placed between /\* and \*/ or make up the remainder of a line beginning with //