A Grammar for the C- Programming Language
(version S07)
January 12, 2007

1 Introduction

This is a grammar for the C- programming language. This language is very similar to C and has a lot of features in common with a real-world structured programming language. There are also some real differences between C and C-. For instance the declaration of procedure arguments, allowable variable names, what constitutes the body of a procedure etc.

For the grammar that follows here are the types of the various elements by type font:

- **Keywords are in this type font.**
- **TOKEN CLASSES ARE IN THIS TYPE FONT.**
- **Nonterminals are in this type font.**

The symbol $\epsilon$ means the empty string.

1.1 Some Token Definitions

```
letter = a | ... | z | A | ... | Z
digit = 0 | ... | 9
ID = letter+digit*
NUM = digit+
```

Also note that **white space** is ignored except that it must separate ID’s, NUM’s, and keywords. **Comments** are treated like white space. Comments begin with // and run to the end of the line. All **keywords** are in lowercase. You need not worry about being case independent since not all lex/flex programs make that easy.
2 The Grammar

1. program $\rightarrow$ declaration-list

2. declaration-list $\rightarrow$ declaration-list declaration | declaration

3. declaration $\rightarrow$ var-declaration | fun-declaration

4. var-declaration $\rightarrow$ type-specifier var-decl-list ;

5. var-decl-list $\rightarrow$ var-decl-list , var-decl-id | var-decl-id

6. var-decl-id $\rightarrow$ ID | ID [ NUM ]

7. type-specifier $\rightarrow$ int | void | bool

8. fun-declaration $\rightarrow$ type-specifier ID ( params ) statement

9. params $\rightarrow$ param-list | $\epsilon$

10. param-list $\rightarrow$ param-list ; param-type-list | param-type-list

11. param-type-list $\rightarrow$ type-specifier param-id-list

12. param-id-list $\rightarrow$ param-id-list , param-id | param-id

13. param-id $\rightarrow$ ID | ID [ ]

14. compound-stmt $\rightarrow$ { local-declarations statement-list }

15. local-declarations $\rightarrow$ local-declarations var-declaration | $\epsilon$

16. statement-list $\rightarrow$ statement-list statement | $\epsilon$

17. statement $\rightarrow$ expression-stmt | compound-stmt | selection-stmt | iteration-stmt | return-stmt | break-stmt

18. expression-stmt $\rightarrow$ expression ; | ;

19. selection-stmt $\rightarrow$ if ( expression ) statement | if ( expression ) statement else statement

20. iteration-stmt $\rightarrow$ while ( expression ) statement

21. return-stmt $\rightarrow$ return ; | return expression ;

22. break-stmt $\rightarrow$ break ;

23. expression $\rightarrow$ var = expression | var += expression | var -= expression | simple-expression

24. var $\rightarrow$ ID | ID [ expression ]
25. $\text{simple-expression} \rightarrow \text{simple-expression} \mid \text{or-expression} \mid \text{or-expression}$

26. $\text{or-expression} \rightarrow \text{or-expression} \& \text{unary-rel-expression} \mid \text{unary-rel-expression}$

27. $\text{unary-rel-expression} \rightarrow ! \text{unary-rel-expression} \mid \text{rel-expression}$

28. $\text{rel-expression} \rightarrow \text{add-expression} \ \text{relop} \ \text{add-expression} \mid \text{add-expression}$

29. $\text{relop} \rightarrow \leq \mid < \mid > \mid >= \mid == \mid !=$

30. $\text{add-expression} \rightarrow \text{add-expression} \ \text{addop} \ \text{term} \mid \text{term}$

31. $\text{addop} \rightarrow + \mid -$  

32. $\text{term} \rightarrow \text{term} \ \text{mulop} \ \text{unary-expression} \mid \text{unary-expression}$

33. $\text{mulop} \rightarrow \ast \mid / \mid \%$

34. $\text{unary-expression} \rightarrow - \ \text{unary-expression} \mid \text{factor}$

35. $\text{factor} \rightarrow ( \ \text{expression} \ ) \mid \text{var} \mid \text{call} \mid \text{constant}$

36. $\text{constant} \rightarrow \text{NUM} \mid \text{true} \mid \text{false}$

37. $\text{call} \rightarrow \text{ID} ( \ \text{args} \ )$

38. $\text{args} \rightarrow \text{arg-list} \mid \epsilon$

39. $\text{arg-list} \rightarrow \text{arg-list} \ , \ \text{expression} \mid \text{expression}$