

Extended BNF Grammar for C Minus

Metasymbols: The following metasymbols will be used for this grammar. $\{ statement \}$ means 0 or more repetitions of *statement*. $[statement]_+$ means that the statement is optional.

1. $program \rightarrow declaration\text{-}list$
2. $declaration\text{-}list \rightarrow declaration \{ declaration \}$
3. $declaration \rightarrow var\text{-}declaration \mid fun\text{-}declaration$
4. $var\text{-}declaration \rightarrow type\text{-}specifier \ var\text{-}list ;$
5. $var\text{-}list \rightarrow ID [[NUM]]_+ \{ , ID [[NUM]]_+ \}$
6. $type\text{-}specifier \rightarrow \mathbf{int} \mid \mathbf{void}$
7. $fun\text{-}declaration \rightarrow type\text{-}specifier \mathbf{ID} (params) compound\text{-}stmt$
8. $params \rightarrow \mathbf{void} \mid param\text{-}list$
9. $param\text{-}list \rightarrow param \{ , param \}$
10. $param \rightarrow type\text{-}specifier \mathbf{ID} [[]]_+$
11. $compound\text{-}stmt \rightarrow \{ local\text{-}declarations \ statement\text{-}list \}$
12. $local\text{-}declarations \rightarrow \{ var\text{-}declarations \}$
13. $statement\text{-}list \rightarrow \{ statement \}$
14. $statement \rightarrow expression\text{-}stmt$
 $\quad \mid compound\text{-}stmt$
 $\quad \mid selection\text{-}stmt$
 $\quad \mid iteration\text{-}stmt$
 $\quad \mid assignment\text{-}stmt$
 $\quad \mid return\text{-}stmt$
 $\quad \mid read\text{-}stmt$
 $\quad \mid write\text{-}stmt$
15. $expression\text{-}stmt \rightarrow expression ; \mid ;$
16. $selection\text{-}stmt \rightarrow \mathbf{if} (expression) statement [\mathbf{else} statement]_+$

17. $iteration\text{-}stmt \rightarrow \mathbf{while} (expression) statement$
18. $return\text{-}stmt \rightarrow \mathbf{return} [expression]_+$
19. $read\text{-}stmt \rightarrow \mathbf{read} var ;$
20. $write\text{-}stmt \rightarrow \mathbf{write} expression ;$
 $\quad \quad \quad | \mathbf{write} string ;$
21. $assignment\text{-}stmt \rightarrow var = simple\text{-}expression$
22. $var \rightarrow \mathbf{ID} [[expression]]_+$
23. $expression \rightarrow simple\text{-}expression$
24. $simple\text{-}expression \rightarrow additive\text{-}expression [relop additive\text{-}expression]_+$
22. $relop \rightarrow <= | < | > | >= | == | !=$
23. $additive\text{-}expression \rightarrow term \{ addop term \}$
24. $addop \rightarrow + | -$
25. $term \rightarrow factor \{ multop factor \}$
26. $multop \rightarrow * | /$
27. $factor \rightarrow (expression) | \mathbf{NUM} | var | call | -factor$
28. $call \rightarrow \mathbf{ID} (args)$
29. $args \rightarrow arg\text{-}list | empty$
30. $arg\text{-}list \rightarrow expression \{ , expression \}$

Regular Expressions

1. $\mathbf{ID} = letter (_letter)^*$ -- IDs can have underbar in them, but not to start with.
2. $\mathbf{NUM} = digit digit^*$
3. $\mathbf{letter} = a | b | \dots | z | A | B | \dots | Z$

4. *digit* = *0* | *1* | ... | *9*
5. *string* = |".*|"
6. *//.** are comments and should be consumed by lex