**Syntax of The C Programming Language**

**Cross-Refferences**

* 1. **ASCII**::= See [**http://www.csci.csusb.edu/dick/samples/comp.text.ASCII.html**](http://www.csci.csusb.edu/dick/samples/comp.text.ASCII.html)

Used\_in The definition of C++ [ [**c++.syntax.html**](http://www.csci.csusb.edu/dick/samples/c++.syntax.html) ]

Used\_in The definition of Java [ [**java.syntax.html**](http://www.csci.csusb.edu/dick/samples/java.syntax.html) ]

**Notation**

This uses my XBNF Extended BNF Notation where "|" indicates "or", "(...)" indicates priority. For more information see [ [**intro\_ebnf.html**](http://csci.csusb.edu/dick/maths/intro_ebnf.html) ]

The following abbreviations are also used:

* + 1. **O**(\_)::= 0 or 1 occurrences,
    2. **N**(\_)::= 1 or more occurrence
    3. **L**(\_)::= a comma separated list
    4. **#**(\_)::= 0 or more occurrences.
    5. **S**(E,Op)::=**[serial\_operator\_expression](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "serial_operator_expression)**(E, Op)
    6. **serial\_operator\_expression**(E,Op)::= E #(Op E).

S(E,Op) = E Op E Op E Op ... E

It also uses the following shorthand

**Lexemes**

* + 1. **identifier**::=**[nondigit](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "nondigit)** #(**[nondigit](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "nondigit)** | [**digit**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#digit)),
    2. **nondigit**::="\_" | "a" | "A" | "b" | "B" | "c" | "C" | "d" | "D" | "e" | "E" | "f" | "F" | "g" | "G" | "h" | "H" | "i" | "I" | "j" | "J" | "k" | "K" | "l" | "L" | "m" | "M" | "n" | "N" | "o" | "O" | "p" | "P" | "q" | "Q" | "r" | "R" | "s" | "S" | "t" | "T" | "u" | "U" | "v" | "V" | "w" | "W" | "x" | "X" | "y" | "Y" | "z" | "Z",
    3. **digit**::="0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9",
    4. **punctuator**::="[" | "]" | "(" | ")" | "{" | "}" | "\*" | "," | ":" | "=" | ";" | "..." | "#",
    5. **operator**::="[" | "]" | "(" | ")" | "." | "->" | "++" | "--" | "&" | "\*" | "+" | "-" | "~" | "!" | "sizeof" | "/" | "%" | "<<" | ">>" | "<" | ">" | "<=" | ">=" | "==" | "!=" | "^" | "|" | "&&" | "||" | "?" | ":" | "=" | "\*=" | "/=" | "%=" | "+=" | "-=" | "<<=" | ">>=" | "&=" | "^=" | "||=" | "," | "#" | "##",
    6. **infix**::= "->" | "&" | "\*" | "+" | "-" | "/" | "%" | "<<" | ">>" | "<" | ">" | "<=" | ">=" | "==" | "!=" | "^" | "|" | "&&" | "||" | "=" | "\*=" | "/=" | "%=" | "+=" | "-=" | "<<=" | ">>=" | "&=" | "^=" | "||=" | "," ,
    7. **prefix**::= "++" | "--" | "&" | "\*" | "+" | "-" | "~" | "!" | "sizeof" ,
    8. **postfix**::= "++" | "--",
    9. **integer\_suffix**::=#(**[unsigned\_suffix](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "unsigned_suffix)**) | #(**[long\_suffix](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "long_suffix)**),
    10. **unsigned\_suffix**::="u" | "U",
    11. **long\_suffix**::="l" | "L",
    12. **sign**::="+" | "-",
    13. **octal\_constant**::="0" #(**[octal\_digit](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "octal_digit)**),
    14. **octal\_digit**::="0" | "1" | "2" | "3" | "4" | "5" | "6" | "7",
    15. **hex\_constant**::=("0x" | "0X") (**[hex\_digit](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "hex_digit)**),
    16. **hex\_digit**::="0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9" | "a" | "b" | "c" | "d" | "e" | "f" | "A" | "B" | "C" | "D" | "E" | "F",
    17. **decimal\_constant**::=**[non\_zero\_digit](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "non_zero_digit)** #([**digit**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#digit)),
    18. **non\_zero\_digit**::="1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9",
    19. **integer\_constant**::=(**[decimal\_constant](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "decimal_constant)** | [**octal\_constant**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#octal_constant) | [**hex\_constant**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#hex_constant)) | [**integer\_suffix**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#integer_suffix),
    20. **float\_suffix**::="f" | "l" | "F" | "L",
    21. **fraction**::=#[**digit**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#digit) "." [**digit**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#digit) #[**digit**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#digit),
    22. **exponent\_part**::=("e" | "E") [**sign**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#sign) #([**digit**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#digit)),
    23. **float\_constant**::=[**fraction**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#fraction) (**[exponent\_part](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "exponent_part)**|) (**[float\_suffix](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "float_suffix)**|)|(**[decimal\_constant](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "decimal_constant)** (**[exponent\_part](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "exponent_part)**|) [**float\_suffix**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#float_suffix),
    24. **enumeration\_constant**::=[**identifier**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#identifier),
    25. **char\_constant**::=[**char**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#char)~(**[double\_quote](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "double_quote)**|[**eoln**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#eoln)|[**backslash**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#backslash))| [**escape\_sequence**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#escape_sequence),
    26. **escape\_sequence**::=[**backslash**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#backslash) ([**char**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#char) | "0" #**[octal\_digit](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "octal_digit)** |"0x"#**[hexadecimal\_digit](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "hexadecimal_digit)**),
    27. **character\_constant**::="'" [**char\_constant**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#char_constant)"'" ,

constant :=::=**[float\_constant](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "float_constant)** | [**integer\_constant**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#integer_constant) | [**enumeration\_constant**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#enumeration_constant) | [**character\_constant**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#character_constant),

* + 1. **string\_\_char**::=[**char**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#char)~(**[double\_quote](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "double_quote)**|[**eoln**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#eoln)|[**backslash**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#backslash))| [**escape\_sequence**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#escape_sequence),
    2. **string\_literal**::=**[double\_quote](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "double_quote)** #(**[string\_char](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "string_char)**) [**double\_quote**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#double_quote),

. . . . . . . . . ( end of section [**Lexemes**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#Lexemes)) [**<<**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#Contents)Contents | End[**>>**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#__End)

**Expressions**

Expressions are made up by applying operators to [**primary\_expressions**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#primary_expressions).

* + 1. **primary\_expression**::= [**variable**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#variable) | [**constant**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#constant) | [**string\_literal**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#string_literal) | "(" [**expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#expression) ")",
    2. **variable**::= [**identifier**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#identifier) & *declared and in scope of declaration*.
    3. **argument\_list**::=[**List**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#List)(**[assignment\_expression](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "assignment_expression)**),

**Operators**

|  |  |
| --- | --- |
| **Symbol** | **See** |
| "("... ")" | [**primary\_expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#primary_expression) [**cast\_expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#cast_expression) function\_call |
| "." | part of a structure |
| "-" | [**additive\_expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#additive_expression) [**unary\_expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#unary_expression) |
| "->" | part of a pointed at structure |
| "--" | [**unary\_expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#unary_expression) [**postfix\_expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#postfix_expression) |
| "-=" | [**assignment\_expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#assignment_expression) |
| "&" | [**AND\_expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#AND_expression) bitwise Boolean |
| "&=" | [**assignment\_expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#assignment_expression) |
| "&" | address\_of [**unary\_expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#unary_expression) |
| "&&" | [**logical\_AND\_expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#logical_AND_expression) |
| "\*" | [**multiplicative\_expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#multiplicative_expression) contents of pointer [**unary\_expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#unary_expression) |
| "\*=" | [**assignment\_expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#assignment_expression) |
| "+" | [**additive\_expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#additive_expression) [**unary\_expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#unary_expression) |
| "++" | [**unary\_expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#unary_expression) [**postfix\_expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#postfix_expression) |
| "+=" | [**assignment\_expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#assignment_expression) |
| "~" | bitwise negation prefix |
| "!" | logical negation prefix |
| "!=" | [**equality\_expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#equality_expression) |
| "sizeof" | [**unary\_expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#unary_expression) |
| "/" | [**multiplicative\_expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#multiplicative_expression) divide |
| "/=" | [**assignment\_expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#assignment_expression) |
| "%" | [**multiplicative\_expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#multiplicative_expression) mod |
| "%=" | [**assignment\_expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#assignment_expression) |
| "<" | [**relational\_expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#relational_expression) |
| "<<" | [**shift\_expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#shift_expression) left |
| "<<=" | [**assignment\_expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#assignment_expression) |
| "<=" | [**relational\_expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#relational_expression) |
| ">" | [**relational\_expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#relational_expression) |
| ">>" | [**shift\_expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#shift_expression) right |
| ">=" | [**relational\_expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#relational_expression) |
| ">>=" | [**assignment\_expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#assignment_expression) |
| "==" | [**equality\_expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#equality_expression) |
| "=" | [**assignment\_expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#assignment_expression) |
| "^" | [**XOR\_expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#XOR_expression) exclusive-or bitwise |
| "^=" | [**assignment\_expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#assignment_expression) |
| "|" | [**OR\_expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#OR_expression) bitwise or |
| "||" | [**logical\_OR\_expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#logical_OR_expression) |
| "||=" | [**assignment\_expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#assignment_expression) |
| ..."?"... ":"... | [**conditional\_expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#conditional_expression) |
| "," | [**expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#expression) (discard previous value) |

**Arithmetic**

* + 1. **post\_fix**::="++" | "--",
    2. **post\_fix\_expression**::=(**[primary\_expression](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "primary_expression)**) #(**[post\_fix](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "post_fix)**),
    3. **unary\_operator**::="&" | "\*" | "+" | "-" | "!" | "-",
    4. **pre\_fix**::="++" | "--" | "sizeof",
    5. **unary\_expression**::=#([**pre**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#pre)-[**fix**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#fix)) [**post\_fix\_expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#post_fix_expression) | [**unary\_operator**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#unary_operator) [**cast\_expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#cast_expression) | "sizeof" "(" [**type\_name**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#type_name)")",
    6. **cast\_expression**::=#(**[type\_name](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "type_name)**) [**unary\_expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#unary_expression). This implies that casts are done after doing post-fix operations..
    7. **multiplicative\_expression**::=**[S](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "S)**(**[cast\_expression](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "cast_expression)**, [**multiplicative\_operator**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#multiplicative_operator)). [ [**serial\_operator\_expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#serial_operator_expression) ]

The rule above means that 'casts' are done before multiplication and division, and that multiplication and division are done from left to right.

* + 1. **multiplicative\_operator**::="\*" | "%" | "/",
    2. **additive\_expression**::=**[S](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "S)**(**[multiplicative\_expression](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "multiplicative_expression)**, [**additive\_operator**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#additive_operator)). This means that addition and subtraction occurs after multiplication and from left to right.
    3. **additive\_operator**::="+" | "-",

**Shifts**

* + 1. **shift\_expression**::=[**S**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#S)(**[additive\_expression](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "additive_expression)**, [**shift\_operator**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#shift_operator)),
    2. **shift\_operator**::=">>" | "<<", "<<" is left shift of bits (multiply by 2), and ">>" is the reverse and divides by 2.

**Relations**

* + 1. **relational\_expression**::= [**S**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#S)(**[shift\_expression](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "shift_expression)**, [**relational\_operator**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#relational_operator)),
    2. **relational\_operator**::="<" | ">" | "<=" | ">=",
    3. **equality\_expression**::=[**S**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#S)(**[relational\_expression](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "relational_expression)**, [**equality\_operator**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#equality_operator)),
    4. **equality\_operator**::="==" | "!=",

**Bitwise Expressions**

These use the lowest level machine code operations that manipulate the bits in words. They are very fast and powerful. They are mostly used in system programming: drivers, operating systems, compilers, interpreters, shells, ... . They are also a very fast and tight coding for subsets of small sets: one bit per element has 1 for membership and 0 for nonmembership. For example if a class can meet on any collection of the 5 working days in a week (Monday=1, ..., Friday=5) then

FRWTM

10101 = MWF

|  |  |
| --- | --- |
| **Decimal** | **Binary (last 4 bits/ 1 byte)** |
| 0 | 0000 |
| 1 | 0001 |
| 2 | 0010 |
| 3 | 0011 |
| 4 | 0100 |
| 5 | 0101 |
| 6 | 0110 |
| 7 | 0111 |
| 8 | 1000 |

* + 1. **AND\_expression**::=[**S**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#S)(**[equality\_expression](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "equality_expression)**, [**and\_operator**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#and_operator)),
    2. **and\_operator**::="&", This operator takes each bit in the value of its arguments in turn to calculate the bit in the answer. A bit is 1 if and only if both arguments have bits in that place that are 1.

|  |  |
| --- | --- |
| **Decimal** | **Binary** |
| 3 | 0011 |
| 5 | 0101 |
| 3&5 | 0001 |

* + 1. **XOR\_expression**::=[**S**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#S)(**[AND\_expression](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "AND_expression)**, [**XOR\_operator**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#XOR_operator)),
    2. **XOR\_operator**::="^", XOR is short for eXclusive-OR. The n'th bit in the value is 1 precisly when the n'th bits in the two arguments are different.

|  |  |
| --- | --- |
| **Decimal** | **Binary** |
| 3 | 0011 |
| 5 | 0101 |
| 3^5 | 0110 |

* + 1. **OR\_expression**::=[**S**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#S)(**[XOR\_expression](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "XOR_expression)**, [**OR\_operator**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#OR_operator)),
    2. **OR\_operator**::="|", This operator takes each bit in the value of its arguments in turn to calculate the bit in the answer. The n'th bit is 1 if either n'th bits is 1.

|  |  |
| --- | --- |
| **Decimal** | **Binary** |
| 3 | 0011 |
| 5 | 0101 |
| 3|5 | 0111 |

* + 1. **Logical Expressions**
    2. In C, logical false is reresented by any zero value and true by any nonzero value. Here is a list of operators
       1. **and**::="&&".
       2. **or**::="||",
       3. **not**::="!",
    3. **logical\_AND\_expression**::=[**S**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#S)(**[OR\_expression](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "OR_expression)**, [**logical\_AND\_operator**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#logical_AND_operator)),
    4. **logical\_AND\_operator**::=**[and](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "and)**, A&&B is true precisely when both A and B evaluate to be true. If A evaluates to false, B is not evaluated.
    5. **logical\_OR\_expression**::=[**S**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#S)(**[logical\_AND\_expression](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "logical_AND_expression)**, [**logical\_OR\_operator**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#logical_OR_operator)),
    6. **logical\_OR\_operator**::=**[or](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "or)**, A||B is true if A evaluates to be true, or when A is false and B evaluates to be true. If both evaluate to false (zero) then A||B is false.

**Conditional Expressions**

* + 1. **conditional\_expression**::=**[logical\_OR\_expression](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "logical_OR_expression)** | [**logical\_OR\_expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#logical_OR_expression) "?" [**expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#expression) ":" [**conditional\_expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#conditional_expression),

**Assignment Statements**

* + 1. **assignment\_expression**::=[**S**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#S)(**[unary\_expression](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "unary_expression)**, [**assignment\_operator**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#assignment_operator)),
    2. **assignment\_operator**::="=" | "\*=" | "/=" | "%=" | "+=" | "<<=" | ">>=" | "&=" | "^=" | "|=",
    3. **expression**::=[**List**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#List)(**[assignment\_expression](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "assignment_expression)** ),
    4. **constant\_expression**::=**[conditional\_expression](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "conditional_expression)**,

. . . . . . . . . ( end of section [**Expressions**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#Expressions)) [**<<**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#Contents)Contents | End[**>>**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#__End)

**Declarations**

* + 1. **declaration**::=**[declaration\_specifier](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "declaration_specifier)** | [**declarator\_list**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#declarator_list),
    2. **declarator\_list**::=[**List**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#List)(**[declarator\_initialized](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "declarator_initialized)**),
    3. **declaration\_specifier**::=(**[storage\_class](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "storage_class)** | [**type\_specifier**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#type_specifier) | [**type\_qualifier**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#type_qualifier)),
    4. **storage\_class**::="typedef" | "extern" | "static" | "auto" | "register",

**Types**

* + 1. **type\_specifier**::="void" | "char" | "short" | "int" | "long" | "float" | "double" | "signed" | "unsigned" | [**struct\_union\_specifier**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#struct_union_specifier) | [**enumeration\_specifier**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#enumeration_specifier) | [**typedef\_name**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#typedef_name),
    2. **type-qualifier**::="const" | "volatile",
    3. **typedef\_name**::=[**identifier**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#identifier),

**Initialization**

* + 1. **initializer**::=**[assignment\_expression](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "assignment_expression)** | [**initializer\_list**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#initializer_list),
    2. **initializer\_list**::=[**List**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#List)(**[initializer](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "initializer)**),
    3. **declarator\_initialized**::=**[declarator](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "declarator)** ("=" [**initializer**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#initializer)),

**Structs and Unions**

* + 1. **structure\_declarator**::=**[declarator](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "declarator)** | [**declarator**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#declarator) ":" [**constant\_expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#constant_expression),
    2. **structure\_declarator\_list**::=[**List**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#List)(**[structure\_declarator](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "structure_declarator)**),
    3. **structure\_declaration**::=(**[type\_specifier](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "type_specifier)** | [**type\_qualifier**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#type_qualifier)) [**structure\_declarator\_list**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#structure_declarator_list) ";" ,
    4. **struct\_union\_specifier**::=**[struct\_union](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "struct_union)** [**identifier**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#identifier) | [**struct\_union**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#struct_union) [**identifier**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#identifier) "{"**[structure\_declarator\_list](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "structure_declarator_list)** "}",
    5. **struct\_union**::=( "struct" | "union" ),

**Enums**

* + 1. **enumeration\_value**::=**[enumeration\_constant](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "enumeration_constant)** ("=" [**constant\_expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#constant_expression)|)
    2. **enumeration\_list**::=[**List**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#List)(**[enumeration\_value](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "enumeration_value)** ),
    3. **enumeration\_specifier**::=**[enumeration\_identifier](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "enumeration_identifier)** | "enum" [**identifier**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#identifier) "{"**[enumeration\_list](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "enumeration_list)**"}",

**Functions**

* + 1. **function\_definition**::=**[declaration\_specifier](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "declaration_specifier)** [**declarator**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#declarator) | [**declaration\_list**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#declaration_list) | [**compound\_statement**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#compound_statement),
    2. **parameter\_declaration**::=#**[declaration\_specifier](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "declaration_specifier)** [**declarator**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#declarator) | [**abstract\_declarator**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#abstract_declarator),
    3. **parameter\_list**::=**[List](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "List)**(**[parameter\_declaration](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "parameter_declaration)**) (",..."|),

**Main Function**

A complete C program has to have a function with name 'main'. This is the function called by the operating system. It must return an *int* value indicating whether the prograam executed correctly or if there was an error. In UNIX, the main program returns 0 to indicate no errors. Their are several valid forms:

* + 1. *int main()*
    2. *int main(argc, argv)*
    3. *int main(argc, argv, envp)* The parameters are set up by the operating system when the program starts. The traditional *arg* stands for *argument*.

**Pointers**

* + 1. **pointer**::=#( "\*" | #**[type\_qualifier](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "type_qualifier)**),
    2. **declarator**::=[**pointer**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#pointer) | [**direct\_declarator**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#direct_declarator),

**Functions and Arrays**

* + 1. **post\_declarator**::="["**[constant\_expression](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "constant_expression)**"]" | "("**[parameter\_list](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "parameter_list)**")" | "("**[identifier\_list](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "identifier_list)**")"
    2. **direct\_declarator**::=[**identifier**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#identifier) | "("**[declarator](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "declarator)**")" | [**direct\_declarator**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#direct_declarator) [**post\_declarator**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#post_declarator),
    3. **abstract\_declarator**::=[**pointer**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#pointer) | [**pointer**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#pointer) [**direct\_abstract\_declarator**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#direct_abstract_declarator),
    4. **direct\_abstract\_declarator**::= "(" [**abstract\_declarator**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#abstract_declarator) ")" | [**O**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#O)( [**direct\_abstract\_declarator**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#direct_abstract_declarator)) [**O**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#O)("[" [**O**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#O)(**[constant\_expression](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "constant_expression)**) "]" | "(" [**O**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#O)(**[parameter\_list](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "parameter_list)**) ")" ),

. . . . . . . . . ( end of section [**Declarations**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#Declarations)) [**<<**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#Contents)Contents | End[**>>**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#__End)

**Statements**

* + 1. **statement**::=**[labeled\_statement](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "labeled_statement)** | [**compound\_statement**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#compound_statement) | [**expression\_statement**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#expression_statement) | [**selection\_statement**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#selection_statement) | [**iteration\_statement**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#iteration_statement) | [**jump\_statement**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#jump_statement)

**Branch**

* + 1. **jump\_statement**::="goto" [**identifier**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#identifier)";" | "continue" ";" | "break;" | "return" [**expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#expression) ";",

**Structured**

* + 1. **loop**::=**[iteration\_statement](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "iteration_statement)**.
    2. **iteration\_statement**::="while" "("[**expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#expression)")" [**statement**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#statement) | "do" [**statement**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#statement) "while" "("[**expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#expression)")" ";" | [**for\_statement**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#for_statement).
    3. **for\_statement**::="for" "("[**expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#expression) ";" [**expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#expression) ";" [**expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#expression)")" [**statement**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#statement),
    4. **selection\_statement**::=**[if\_statement](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "if_statement)** | "switch" "("[**expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#expression)")" [**statement**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#statement),
    5. **if\_statement**::="if ("[**expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#expression)")" [**statement**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#statement) | "if" "("[**expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#expression)")" [**statement**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#statement) "else" [**statement**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#statement).
    6. **expression\_statement**::= [**expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#expression) ";",
    7. **labeled\_statement**::=[**identifier**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#identifier) ":" [**statement**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#statement) | "case" [**constant\_expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#constant_expression) ":" [**statement**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#statement) | "default" ":" [**statement**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#statement),

**Compound**

* + 1. **compound\_statement**::=[**block**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#block) | "{" #[**statement**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#statement) "}",
    2. **block**::="{" [**declaration**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#declaration) #[**declaration**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#declaration) #[**statement**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#statement) "}",

. . . . . . . . . ( end of section [**Statements**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#Statements)) [**<<**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#Contents)Contents | End[**>>**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#__End)

**Pre-Processor Commands**

* + 1. **preprocess\_token**::=[**identifier**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#identifier) | [**constant**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#constant) | [**string\_literal**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#string_literal) | [**operator**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#operator) | [**punctuator**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#punctuator) | [***each***](http://www.csci.csusb.edu/dick/samples/c.syntax.html#each)[***Non***](http://www.csci.csusb.edu/dick/samples/c.syntax.html#Non)*-*[***white***](http://www.csci.csusb.edu/dick/samples/c.syntax.html#white)[***space***](http://www.csci.csusb.edu/dick/samples/c.syntax.html#space)[***not***](http://www.csci.csusb.edu/dick/samples/c.syntax.html#not)[***one***](http://www.csci.csusb.edu/dick/samples/c.syntax.html#one)[***of***](http://www.csci.csusb.edu/dick/samples/c.syntax.html#of)[***the***](http://www.csci.csusb.edu/dick/samples/c.syntax.html#the)[***previous***](http://www.csci.csusb.edu/dick/samples/c.syntax.html#previous),
    2. **header\_char**::=[***any***](http://www.csci.csusb.edu/dick/samples/c.syntax.html#any)[***character***](http://www.csci.csusb.edu/dick/samples/c.syntax.html#character)[***except***](http://www.csci.csusb.edu/dick/samples/c.syntax.html#except)[***new\_line***](http://www.csci.csusb.edu/dick/samples/c.syntax.html#new_line) *|* [***and***](http://www.csci.csusb.edu/dick/samples/c.syntax.html#and) *| >*,
    3. **header\_name**::=#(**[header\_char](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "header_char)**),
    4. **new\_line**::=**[new\_line](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "new_line)** [**character**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#character),
    5. **Left\_paren**::=[***left***](http://www.csci.csusb.edu/dick/samples/c.syntax.html#left)[***parenthesis***](http://www.csci.csusb.edu/dick/samples/c.syntax.html#parenthesis)[***with***](http://www.csci.csusb.edu/dick/samples/c.syntax.html#with)[***no***](http://www.csci.csusb.edu/dick/samples/c.syntax.html#no)[***white***](http://www.csci.csusb.edu/dick/samples/c.syntax.html#white)[***space***](http://www.csci.csusb.edu/dick/samples/c.syntax.html#space)[***before***](http://www.csci.csusb.edu/dick/samples/c.syntax.html#before)[***it***](http://www.csci.csusb.edu/dick/samples/c.syntax.html#it),
    6. **control\_line**::="#include" (#(**[preprocess\_token](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "preprocess_token)** | [**header\_name**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#header_name)) [**new\_line**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#new_line) | "#define" [**identifier**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#identifier) #(**[preprocess\_token](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "preprocess_token)**) [**new\_line**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#new_line) | "#define" [**identifier**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#identifier) [**left\_paren**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#left_paren) [**identifier\_list**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#identifier_list) #(**[preprocess\_token](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "preprocess_token)**) [**new\_line**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#new_line), | "#undef" [**identifier**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#identifier) [**new\_line**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#new_line) | "#line" [**preprocess\_token**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#preprocess_token) [**new\_line**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#new_line) | "#error" [**preprocess\_token**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#preprocess_token) [**new\_line**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#new_line) | "#pragma" [**preprocess\_token**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#preprocess_token) [**new\_line**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#new_line) | "#"**[new\_line](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "new_line)**,
    7. **endif\_line**::="#endif" [**new\_line**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#new_line),
    8. **elif\_group**::="#elif" [**constant\_expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#constant_expression) [**new\_line**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#new_line) [**pp\_group**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#pp_group),
    9. **else\_group**::="#else" [**new\_line**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#new_line) [**pp\_group**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#pp_group),
    10. **if\_group**::=("#if" [**constant\_expression**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#constant_expression) | "#ifdef" [**identifier**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#identifier) | "#ifndef" [**identifier**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#identifier)) [**new\_line**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#new_line) [**pp\_group**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#pp_group),
    11. **if\_part**::=**[if\_group](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "if_group)** #(**[elif\_group](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "elif_group)**) [**else\_group**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#else_group) [**endif\_line**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#endif_line),
    12. **pp\_part**::=#**[preprocess\_token](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "preprocess_token)** [**new\_line**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#new_line) | [**if\_part**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#if_part) | [**control\_line**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#control_line),
    13. **pp\_group**::=#(**[pp\_part](http://www.csci.csusb.edu/dick/samples/c.syntax.html" \l "pp_part)**),

. . . . . . . . . ( end of section [**Pre-Processor Commands**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#Pre-Processor Commands)) [**<<**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#Contents)Contents | End[**>>**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#__End)

. . . . . . . . . ( end of section [**Syntax of The C Programming Language**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#Syntax of The C Programming Language)) [**<<**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#Contents)Contents | End[**>>**](http://www.csci.csusb.edu/dick/samples/c.syntax.html#__End)

**End**