

The following details a proposed LOC counting standard for C++. It is taken from Humphrey's "A Discipline of Software Engineering".

| Count Type | Type | Comments |
|------------------|---------|--|
| Physical/Logical | Logical | Means we are counting logical lines of code and not physical lines of code |

| Statement Type | Included | Comments |
|---------------------|----------|-------------------|
| Executable | Yes | |
| Non-executable: | | |
| Declarations | Yes | See notes 3 and 4 |
| Compiler Directives | Yes | See note 4 |
| Comments | No | |
| With source | No | |
| Banners | No | |
| Blank lines | No | |

| Clarifications | Included | Examples/Cases |
|-----------------------|-------------------|--|
| Empty Statements | Yes | “;”, lone ;’s, etc. |
| Begin..end | See Note 1 | when executable |
| Begin..end | See Note 1 | when not executable |
| Test conditions | Yes | when used as sub-program arguments |
| Expression evaluation | Yes | when used as sub-program arguments |
| End symbols | See Notes 1 and 2 | when terminating executable statements |
| End symbols | See Notes 1 and 2 | when terminating declarations or bodies |
| Then, else, otherwise | See Note 1 | |
| Elseif | Yes | |
| Keywords | Yes | |
| Labels | Yes | branch destinations when on separate lines |

Notes:

1. Count once every occurrence of the following keywords: CASE, DO, ELSE, ENUM, FOR, IF, PRIVATE, PUBLIC, STRICT, SWITCH, UNION, WHILE
2. Count once every occurrence of the following: ; {} or };
3. Count once each variable and parameter declaration
4. Count once each #define, #ifdef, #include, etc. statement