

Measuring the complexity of a program statement due to coupling

Note: Coupling which occurs due to **built-in methods** are **not** considered under this factor.

➤ Complexity of program statement due to coupling (Ccp) is computed as follows:

$$\begin{aligned} & (W_r * N_r) + (W_{mcms} * N_{mcms}) + (W_{mcmd} * N_{mcmd}) + (W_{mcrrms} * N_{mcrrms}) + (W_{mcrrmd} * N_{mcrrmd}) + \\ C_{cp} = & (W_{rmcrrms} * N_{rmcrrms}) + (W_{rmcrrmd} * N_{rmcrrmd}) + (W_{rmcms} * N_{rmcms}) + (W_{rmcmd} * N_{rmcmd}) + \\ & (W_{rmrgvs} * N_{rmrgvs}) + (W_{rmrgvd} * N_{rmrgvd}) + (W_{rmrgvs} * N_{rmrgvs}) + (W_{rmrgvd} * N_{rmrgvd}) \end{aligned}$$

Where:

W _r	= Weight of a recursive call
N _r	= Number of recursive calls
W _{mcms}	= Weight of a regular method calling another regular method in the same file
N _{mcms}	= Number of calls from regular method(s) to other regular methods in the same file
W _{mcmd}	= Weight of a regular method calling another regular method in a different file
N _{mcmd}	= Number of calls from regular method(s) to other regular methods in different files
W _{mcrrms}	= Weight of a regular method calling a recursive method in the same file
N _{mcrrms}	= Number of calls from regular method(s) to recursive methods in the same file
W _{mcrrmd}	= Weight of a regular method calling a recursive method in a different file
N _{mcrrmd}	= Number of calls from regular method(s) to recursive methods in different files
W _{rmcrrms}	= Weight of a recursive method calling another recursive method in the same file
N _{rmcrrms}	= Number of calls from recursive method(s) to other recursive methods in the same file
W _{rmcrrmd}	= Weight of a recursive method calling another recursive method in a different file
N _{rmcrrmd}	= Number of calls from recursive method(s) to other recursive methods in different files
W _{rmcms}	= Weight of a recursive method calling a regular method in the same file
N _{rmcms}	= Number of calls from recursive method(s) to regular methods in the same file
W _{rmcmd}	= Weight of a recursive method calling a regular method in a different file
N _{rmcmd}	= Number of calls from recursive method(s) to regular methods in different files
W _{rmrgvs}	= Weight of a regular method referencing a global variable in the same file
N _{rmrgvs}	= Number of references from regular method(s) to global variables in the same file
W _{rmrgvd}	= Weight of a regular method referencing a global variable in a different file
N _{rmrgvd}	= Number of references from regular method(s) to global variables in different files
W _{rmrgvs}	= Weight of a recursive method referencing a global variable in the same file
N _{rmrgvs}	= Number of references from recursive method(s) to global variables in the same file
W _{rmrgvd}	= Weight of a recursive method referencing a global variable in a different file
N _{rmrgvd}	= Number of references from recursive method(s) to global variable in different files

- The weight allocated for a program statement due to coupling differs as follows:

Coupling Type	Weight
A recursive call (Refer to Ex1 in fig. 1)	2
A regular method calling another regular method in the same file	2
A regular method calling another regular method in a different file	3
A regular method calling a recursive method in the same file (Refer to Ex4 in fig. 1)	3
A regular method calling a recursive method in a different file	4
A recursive method calling another recursive method in the same file (Refer to Ex2 in fig. 1)	4
A recursive method calling another recursive method in a different file	5
A recursive method calling a regular method in the same file	3
A recursive method calling a regular method in a different file	4
A regular method referencing a global variable in the same file (Refer to Ex3 in fig. 1)	1
A regular method referencing a global variable in a different file	2
A recursive method referencing a global variable in the same file	1
A recursive method referencing a global variable in a different file	2

Measuring the complexity of a program statement due to control structures

Note: Only the program statements with control structures are considered under this factor.

- Complexity of a program statement with a control structure is computed as follows:

$$Ccs = (Wtcs * NC) + Ccspps$$

Where:

Ccs = Complexity of a program statement with a control structure

Wtcs = Weight due to control structure type

NC = Number of conditions in the control structure

Ccspps = Control structure complexity of the previous program statement. Hence, always the value of Ccspps would be **zero** for control structures which reside at the **first nesting level** or **outer most nesting level**.

- Depending on the type, each control structure is assigned with the following weights:

Control Structure Type	Weight
A conditional control structure such as an 'if' or 'else-if' condition	2
An iterative control structure such as a 'for', 'while', or 'do-while' loop	3
The 'switch' statement in a 'switch-case' control structure	2
Each 'case' statement in a 'switch-case' control structure	1