Understand Metrics

Table of Contents

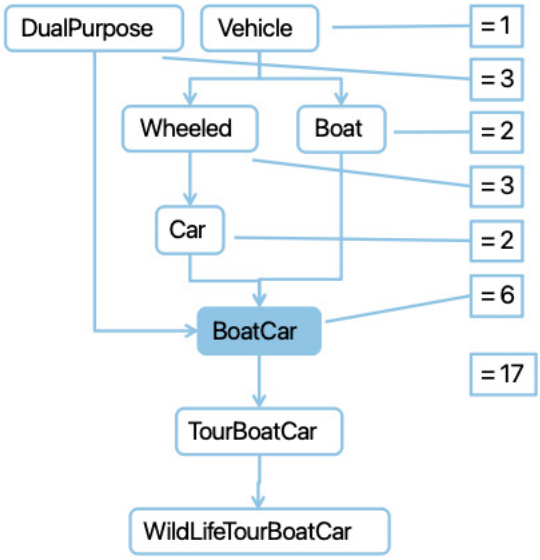
All Methods

API ID: CountDeclMethodAll
Also Known As: Number of Methods (NM), Response for a Class (RFC)
Languages: Basic, C#, C++, Java, Pascal, Python, Web
Targets: Classes

Number of methods, including inherited ones.

Also known as Chidamber & Kemerer - Response for a Class (RFC); Lorenz & Kidd -

Number of Methods (NM).



Targets By Language:

- **Basic:** Class, Struct
- **C#:** Class, Struct
- **C++:** Class, Struct, Union
- **Java:** Class, Interface
- **Pascal:** Class, Interface
- **Python:** Class
- **Web:** PHP Class, PHP Interface

Configuration:

- This metric can be disabled from "Project Configuration/Metrics/Object Orientated" with the "Show method and variable count metrics" option.

Auto-Implemented Properties

API ID: CountDeclPropertyAuto
Languages: C#
Targets: Architectures, Classes, Project

Number of auto-implemented properties.

Targets By Language:

- **C#:** Project, Class, Struct

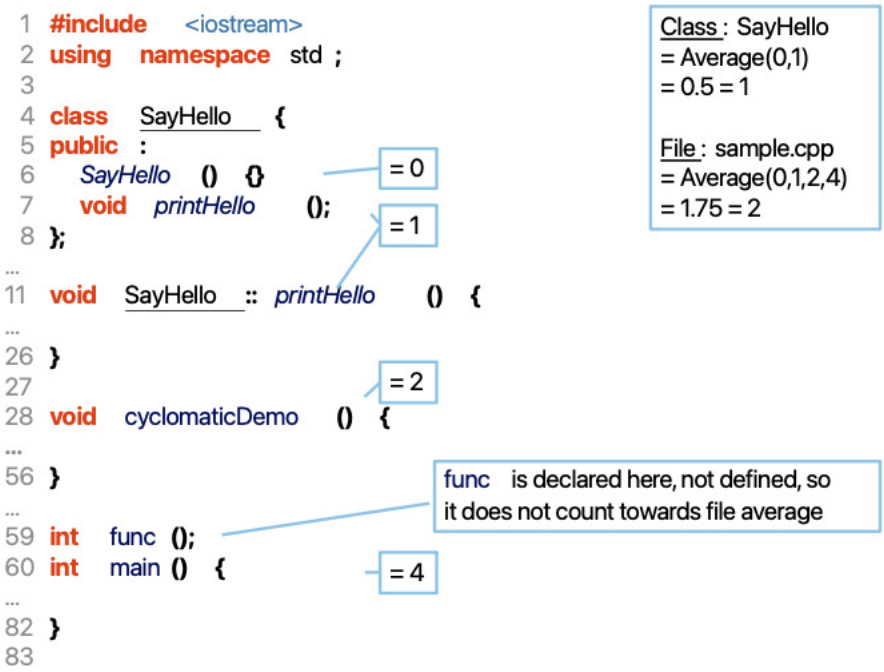
Configuration:

- For projects and architectures, this metric must also be enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show aggregated class metrics" option. Enabling this option can be time consuming.
- This metric can be disabled from "Project Configuration/Metrics/Object Orientated" with the "Show method and variable count metrics" option.

Average Blank Lines

API ID: AvgCountLineBlank
Languages: Ada, Basic, C#, C++, Fortran, Java, Jovial, Pascal, Python, Rust, Web
Targets: Classes, Files, Modules, Packages

Average number of blank lines for all nested functions or methods.



Targets By Language:

- **Ada:** File, Package
- **Basic:** File, Module, Class, Struct
- **C#:** File, Class, Struct
- **C++:** File, Class, Struct, Union
- **Fortran:** File

- **Java:** File, Class, Interface
- **Jovial:** File
- **Pascal:** File, Class, Interface
- **Python:** File, Class
- **Rust:** File
- **Web:** File, PHP Class, PHP Interface

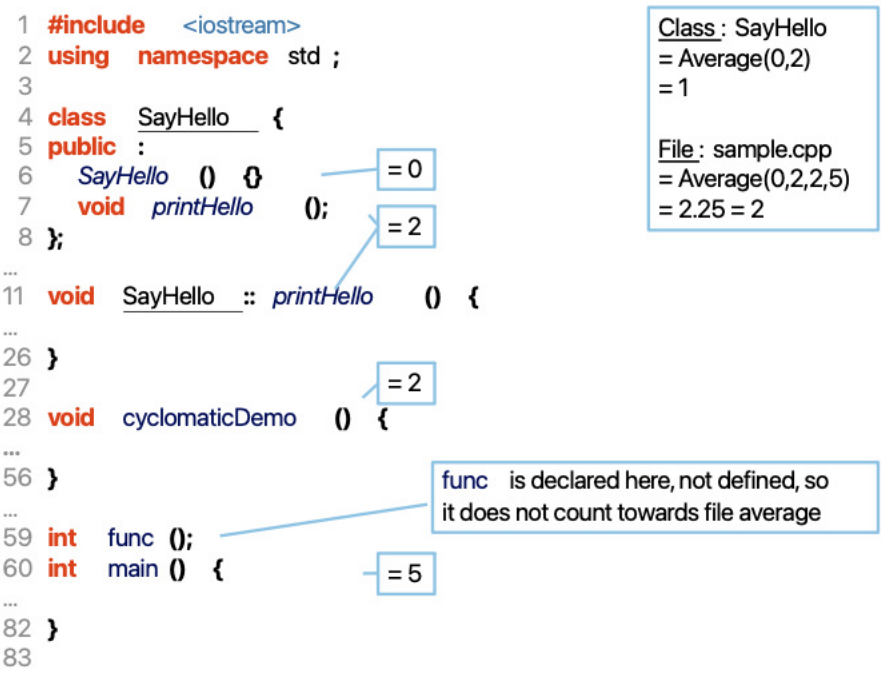
Configuration:

- This metric can be disabled from "Project Configuration/Metrics/Lines and Statements" with the "Show line count metrics" option.
- This metric can be disabled from "Project Configuration/Metrics/Lines and Statements" with the "Include inactive lines" option for languages that have inactive lines.

Average Blank Lines (Includes Inactive)

API ID: AvgCountLineBlankWithInactive
Languages: C++
Targets: Classes, Files

Average number of blank lines for all nested functions or methods, including inactive regions.



Targets By Language:

- **C++:** File, Class, Struct, Union

Configuration:

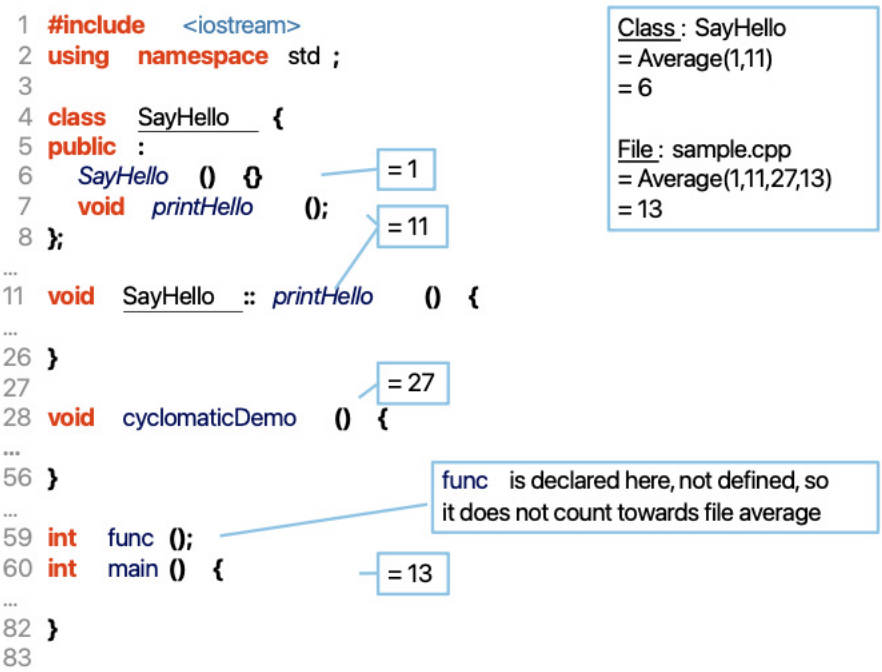
- This metric can be disabled from "Project Configuration/Metrics/Lines and Statements" with the "Show line count metrics" option.
- This metric can be enabled from "Project Configuration/Metrics/Lines and Statements" with the "Include inactive lines" option.

Average Code Lines

API ID: AvgCountLineCode
Also Known As: Average Method Size (AMS)
Languages: Ada, Basic, C#, C++, Fortran, Java, Jovial, Pascal, Python, Rust, Web
Targets: Classes, Files, Modules, Packages

Average number of lines containing source code for all nested functions or methods.

Also known as Lorenz & Kidd - Average Method Size (AMS).



Targets By Language:

- **Ada:** File, Package
- **Basic:** File, Module, Class, Struct
- **C#:** File, Class, Struct
- **C++:** File, Class, Struct, Union

- **Fortran:** File
- **Java:** File, Class, Interface
- **Jovial:** File
- **Pascal:** File, Class, Interface
- **Python:** File, Class
- **Rust:** File
- **Web:** File, PHP Class, PHP Interface

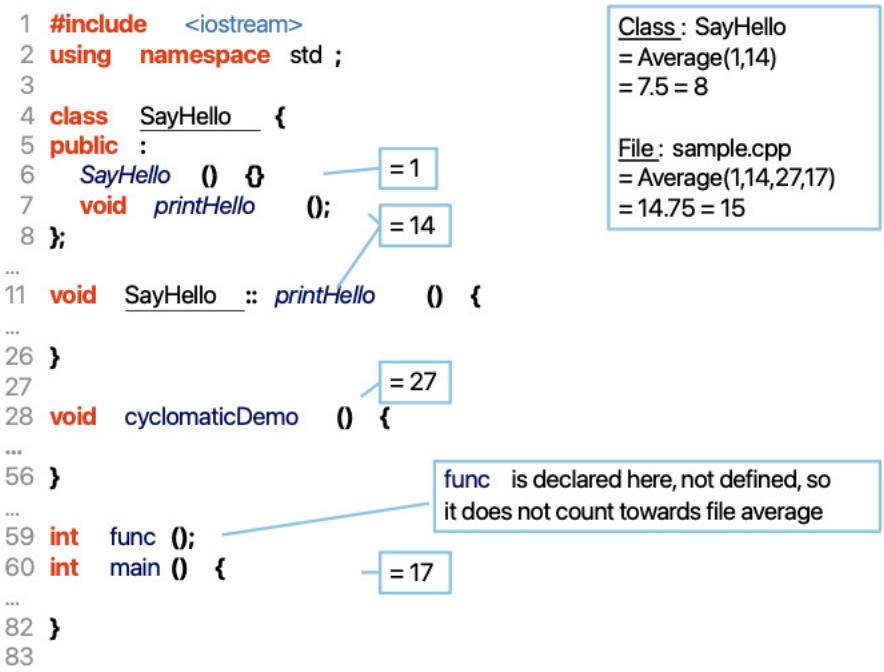
Configuration:

- This metric can be disabled from "Project Configuration/Metrics/Lines and Statements" with the "Show line count metrics" option.
- This metric can be disabled from "Project Configuration/Metrics/Lines and Statements" with the "Include inactive lines" option for languages that have inactive lines.

Average Code Lines (Includes Inactive)

API ID: AvgCountLineCodeWithInactive
Languages: C++
Targets: Classes, Files

Average number of lines containing source code for all nested functions or methods, including inactive regions.



Targets By Language:

- **C++:** File, Class, Struct, Union

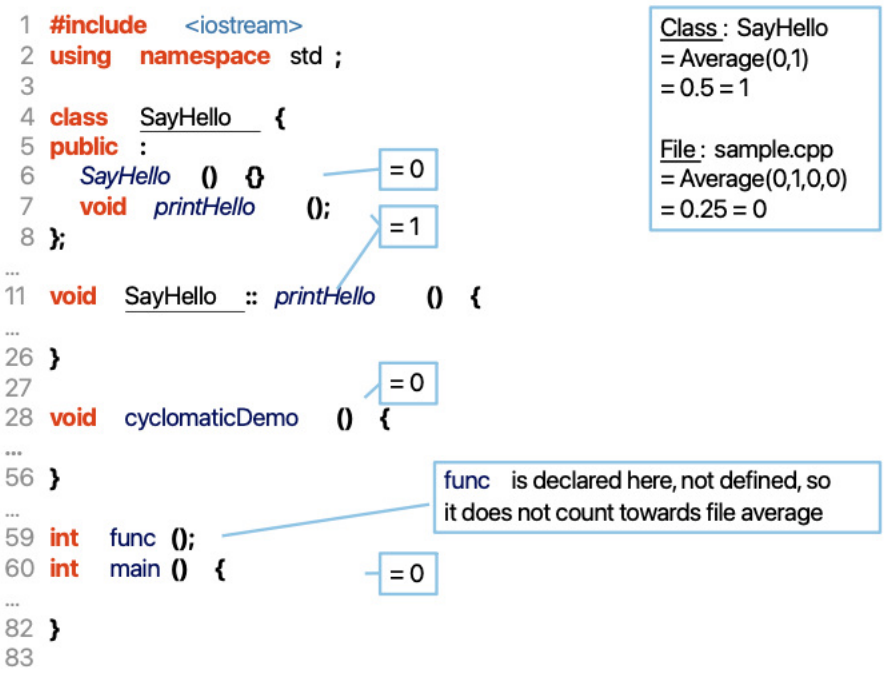
Configuration:

- This metric can be disabled from "Project Configuration/Metrics/Lines and Statements" with the "Show line count metrics" option.
- This metric can be enabled from "Project Configuration/Metrics/Lines and Statements" with the "Include inactive lines" option.

Average Comment Lines

API ID: AvgCountLineComment
Languages: Ada, Basic, C#, C++, Fortran, Java, Jovial, Pascal, Python, Rust, Web
Targets: Classes, Files, Modules, Packages

Average number of lines containing comment for all nested functions or methods.



Targets By Language:

- **Ada:** File, Package
- **Basic:** File, Module, Class, Struct
- **C#:** File, Class, Struct
- **C++:** File, Class, Struct, Union
- **Fortran:** File

- **Java:** File, Class, Interface
- **Jovial:** File
- **Pascal:** File, Class, Interface
- **Python:** File, Class
- **Rust:** File
- **Web:** File, PHP Class, PHP Interface

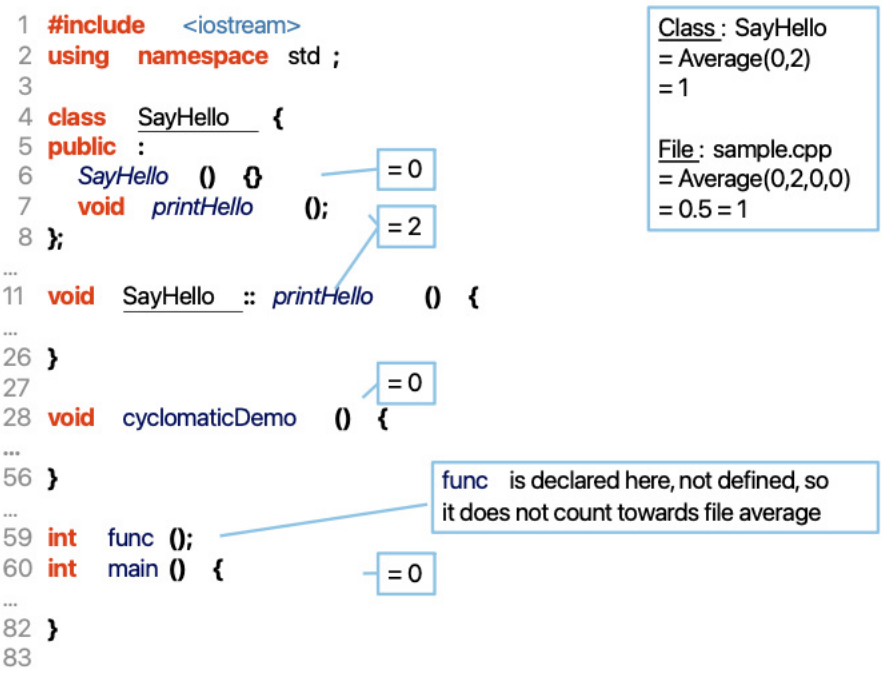
Configuration:

- This metric can be disabled from "Project Configuration/Metrics/Lines and Statements" with the "Show line count metrics" option.
- This metric can be disabled from "Project Configuration/Metrics/Lines and Statements" with the "Include inactive lines" option for languages that have inactive lines.

Average Comment Lines (Includes Inactive)

API ID: AvgCountLineCommentWithInactive
Languages: C++
Targets: Classes, Files

Average number of lines containing comment for all nested functions or methods, including inactive regions.



Targets By Language:

- **C++:** File, Class, Struct, Union

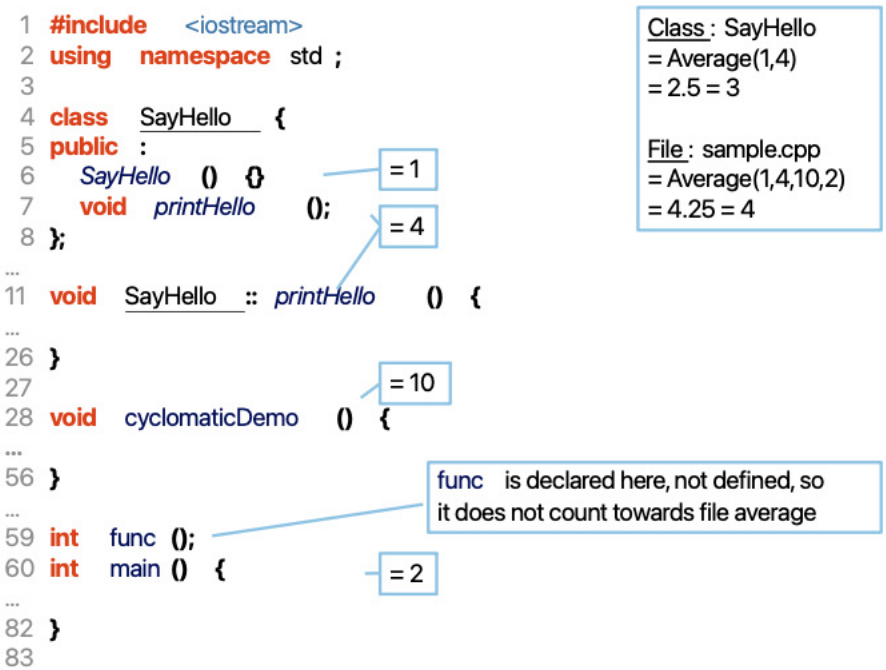
Configuration:

- This metric can be disabled from "Project Configuration/Metrics/Lines and Statements" with the "Show line count metrics" option.
- This metric can be enabled from "Project Configuration/Metrics/Lines and Statements" with the "Include inactive lines" option.

Average Cyclomatic Complexity

API ID: AvgCyclomatic
Languages: Ada, Basic, C#, C++, Fortran, Java, Jovial, Pascal, Python, Rust, VHDL, Web
Targets: Architectures, Classes, Files, Modules, Packages, Project

Average cyclomatic complexity for all nested functions or methods.



Targets By Language:

- **Ada:** Project, File, Package
- **Basic:** Project, File, Module, Class, Struct
- **C#:** Project, File, Class, Struct
- **C++:** Project, File, Class, Struct, Union
- **Fortran:** Project, File

- **Java:** Project, File, Class, Interface
- **Jovial:** Project, File
- **Pascal:** Project, File, Class, Interface, Compunit
- **Python:** Project, File, Class
- **Rust:** Project, File
- **VHDL:** Project, File, Architecture
- **Web:** Project, File, PHP Class, PHP Interface

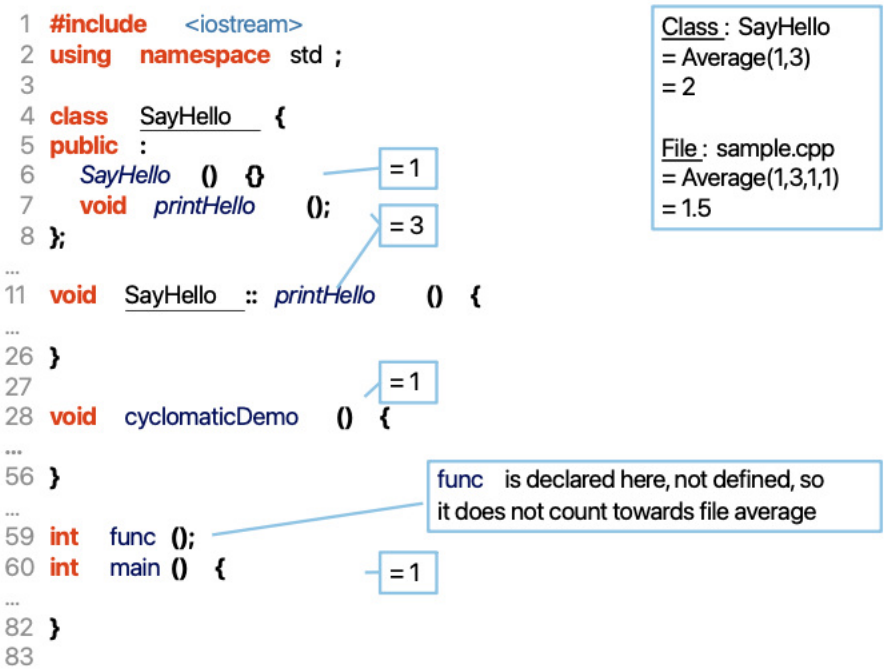
Configuration:

- For projects and architectures, this metric must also be enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show aggregated function metrics" option.
- The cyclomatic metric used is configured from "Project Configuration/Metrics/Complexity". This metric results from having "Count each decision in a multi-way-decision structure" on and having "Count logical conjuctions (OR) and logical and (AND)" off. Checking "Show all available cyclomatic complexity metric variants" will also enable the metric.

Average Essential Complexity

API ID: AvgEssential
Languages: Ada, Basic, C#, C++, Fortran, Java, Jovial, Python, Rust, Web
Targets: Architectures, Classes, Files, Modules, Packages, Project

Average Essential complexity for all nested functions or methods.



Targets By Language:

- **Ada:** Project, File, Package
- **Basic:** Project, File, Module, Class, Struct
- **C#:** Project, File, Class, Struct
- **C++:** Project, File, Class, Struct, Union
- **Fortran:** Project, File
- **Java:** Project, File, Class, Interface
- **Jovial:** Project, File
- **Python:** Project, File, Class
- **Rust:** Project, File
- **Web:** Project, File, PHP Class, PHP Interface

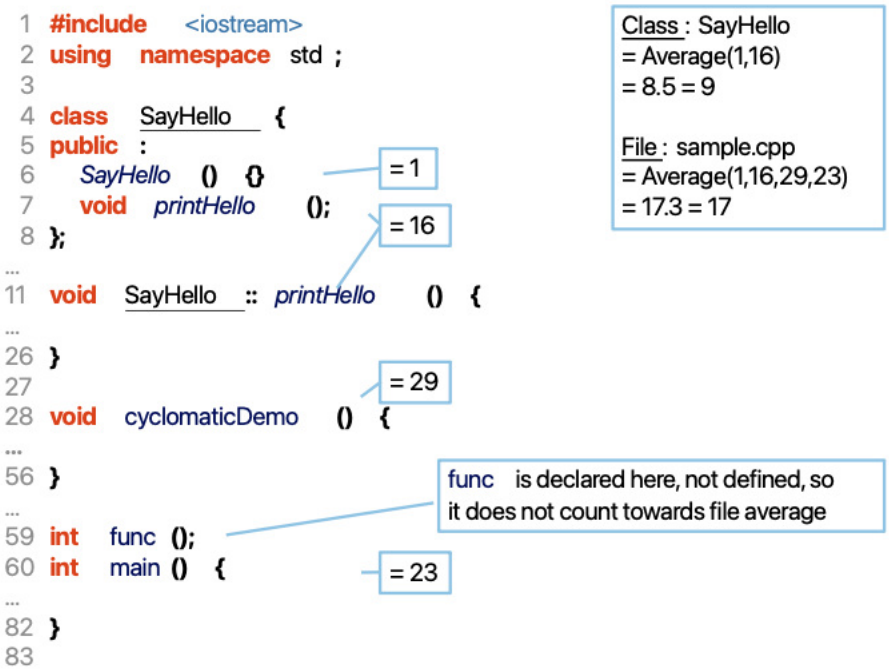
Configuration:

- For projects and architectures, this metric must also be enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show aggregated function metrics" option.
- This metric can be enabled from "Project Configuration/Metrics/Complexity" with the "Essential" option.

Average Lines

API ID: AvgCountLine
Languages: Ada, Basic, C#, C++, Fortran, Java, Jovial, Pascal, Python, Rust, Web
Targets: Classes, Files, Modules, Packages

Average number of lines for all nested functions or methods.



Targets By Language:

- **Ada:** File, Package
- **Basic:** File, Module, Class, Struct
- **C#:** File, Class, Struct
- **C++:** File, Class, Struct, Union
- **Fortran:** File
- **Java:** File, Class, Interface
- **Jovial:** File
- **Pascal:** File, Class, Interface
- **Python:** File, Class
- **Rust:** File
- **Web:** File, PHP Class, PHP Interface

Configuration:

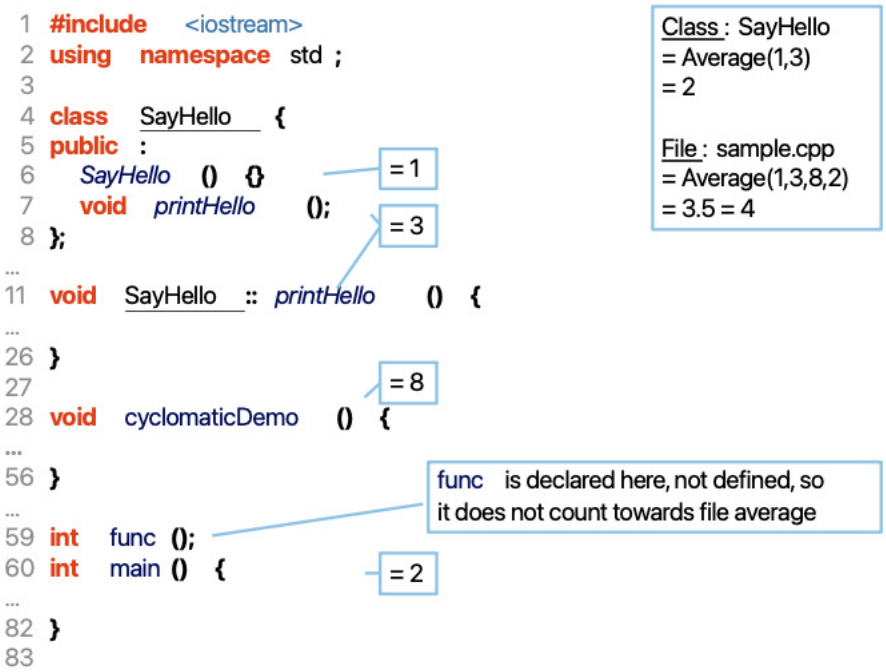
- This metric can be disabled from "Project Configuration/Metrics/Lines and Statements" with the "Show line count metrics" option.

Average Modified Cyclomatic Complexity

API ID: AvgCyclomaticModified
Languages: Ada, Basic, C#, C++, Fortran, Java, Jovial, Pascal, Python, Rust, VHDL, Web

Targets: Architectures, Classes, Files, Modules, Packages, Project

Average modified cyclomatic complexity for all nested functions or methods.



Targets By Language:

- **Ada:** Project, File, Package
- **Basic:** Project, File, Module, Class, Struct
- **C#:** Project, File, Class, Struct
- **C++:** Project, File, Class, Struct, Union
- **Fortran:** Project, File
- **Java:** Project, File, Class, Interface
- **Jovial:** Project, File
- **Pascal:** Project, File, Class, Interface, Compunit
- **Python:** Project, File, Class
- **Rust:** Project, File
- **VHDL:** Project, File, Architecture
- **Web:** Project, File, PHP Class, PHP Interface

Configuration:

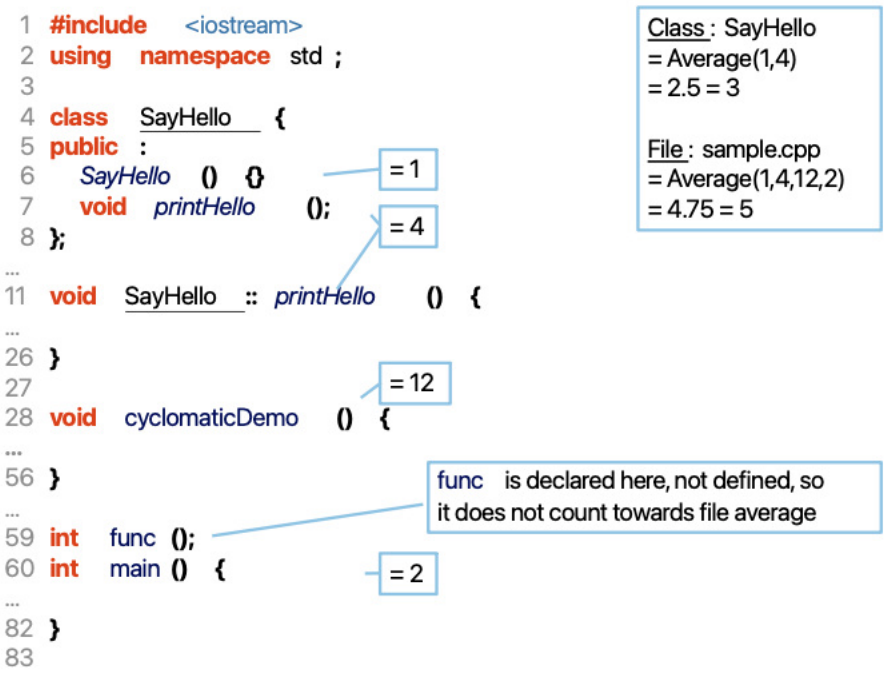
- For projects and architectures, this metric must also be enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show aggregated function metrics" option.
- The cyclomatic metric used is configured from "Project Configuration/Metrics/Complexity". This metric results from having "Count each decision in

a multi-way-decision structure" off and having "Count logical conjuctions (OR) and logical and (AND)" off. Checking "Show all available cyclomatic complexity metric variants" will also enable the metric.

Average Strict Cyclomatic Complexity

API ID: AvgCyclomaticStrict
Languages: Ada, Basic, C#, C++, Fortran, Java, Jovial, Pascal, Python, Rust, Web
Targets: Architectures, Classes, Files, Modules, Packages, Project

Average strict cyclomatic complexity for all nested functions or methods.



Targets By Language:

- **Ada:** Project, File, Package
- **Basic:** Project, File, Module, Class, Struct
- **C#:** Project, File, Class, Struct
- **C++:** Project, File, Class, Struct, Union
- **Fortran:** Project, File
- **Java:** Project, File, Class, Interface
- **Jovial:** Project, File
- **Pascal:** Project, File, Class, Interface, Compunit
- **Python:** Project, File, Class
- **Rust:** Project, File
- **Web:** Project, File, PHP Class, PHP Interface

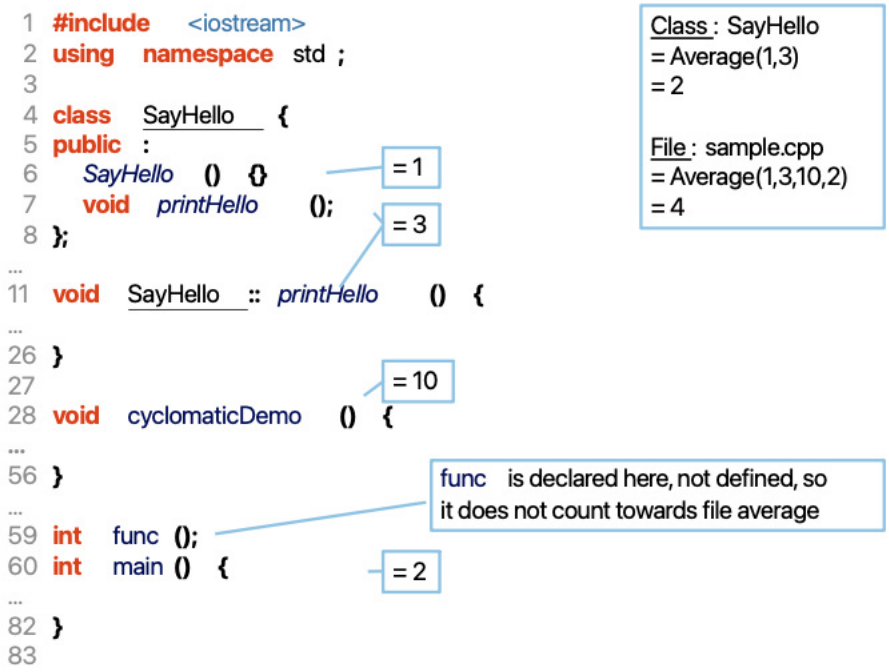
Configuration:

- For projects and architectures, this metric must also be enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show aggregated function metrics" option.
- The cyclomatic metric used is configured from "Project Configuration/Metrics/Complexity". This metric results from having "Count each decision in a multi-way-decision structure" on and having "Count logical conjuctions (OR) and logical and (AND)" on. Checking "Show all available cyclomatic complexity metric variants" will also enable the metric.

Average Strict Modified Cyclomatic Complexity

API ID: AvgCyclomaticStrictModified
Languages: Ada, Basic, C#, C++, Fortran, Java, Jovial, Pascal, Python, Rust, Web
Targets: Architectures, Classes, Files, Modules, Packages, Project

Average strict modified cyclomatic complexity for all nested functions or methods.



Targets By Language:

- **Ada:** Project, File, Package
- **Basic:** Project, File, Module, Class, Struct
- **C#:** Project, File, Class, Struct
- **C++:** Project, File, Class, Struct, Union

- **Fortran:** Project, File
- **Java:** Project, File, Class, Interface
- **Jovial:** Project, File
- **Pascal:** Project, File, Class, Interface, Compunit
- **Python:** Project, File, Class
- **Rust:** Project, File
- **Web:** Project, File, PHP Class, PHP Interface

Configuration:

- For projects and architectures, this metric must also be enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show aggregated function metrics" option.
- The cyclomatic metric used is configured from "Project Configuration/Metrics/Complexity". This metric results from having "Count each decision in a multi-way-decision structure" off and having "Count logical conjuctions (OR) and logical and (AND)" on. Checking "Show all available cyclomatic complexity metric variants" will also enable the metric.

Average Strict Modified Essential Complexity

API ID: AvgEssentialStrictModified
Languages: Ada
Targets: Architectures, Files, Packages, Project

Average strict modified essential complexity for all nested functions or methods.

Targets By Language:

- **Ada:** Project, File, Package

Configuration:

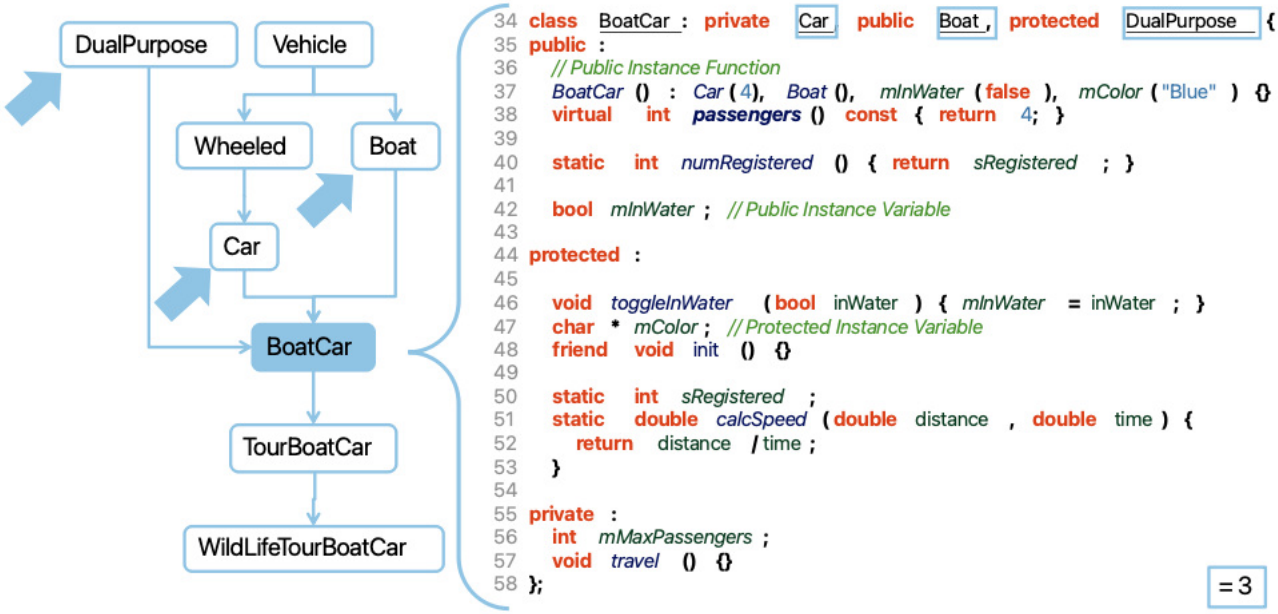
- For projects and architectures, this metric must also be enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show aggregated function metrics" option.
- This metric can be enabled from "Project Configuration/Metrics/Complexity" with the "Essential" option.

Base Classes

API ID: CountClassBase
Also Known As: IFANIN
Languages: Basic, C#, C++, Java, Pascal, Python, Web
Targets: Classes

Number of immediate base classes.

Also known as IFANIN.



Targets By Language:

- **Basic:** Class, Struct
- **C#:** Class, Struct
- **C++:** Class, Struct, Union
- **Java:** Class, Interface
- **Pascal:** Class, Interface
- **Python:** Class
- **Web:** PHP Class, PHP Interface

Configuration:

- This metric can be disabled from "Project Configuration/Metrics/Object Orientated" with the "Show inheritance metrics" option.

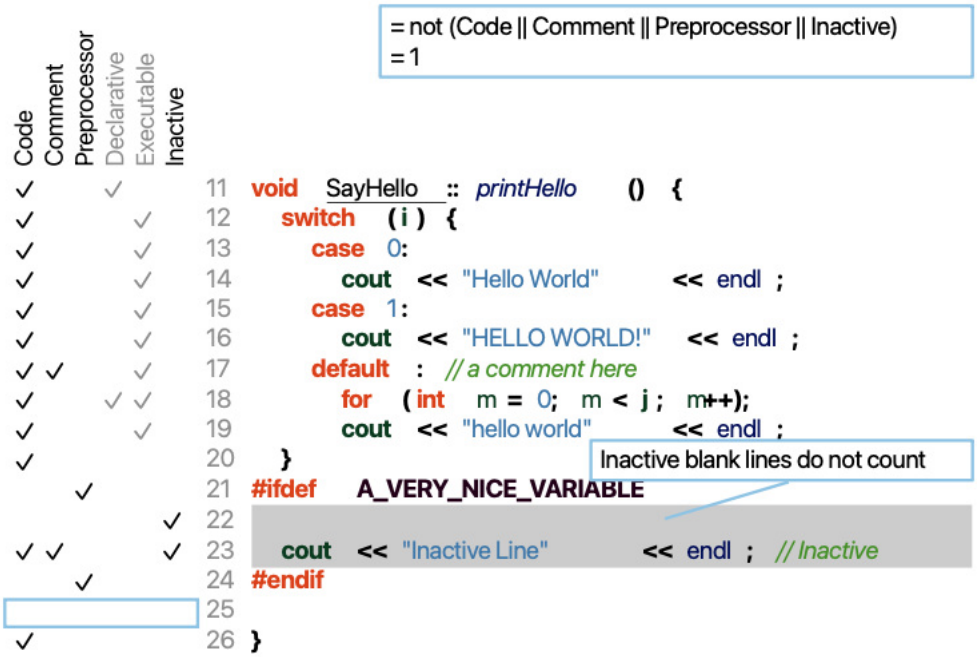
Blank Lines

API ID: CountLineBlank
Also Known As: Blank Lines of Code (BLOC)
Languages: Ada, Assembly, Basic, C#, C++, Fortran, Java, Jovial, Pascal, Python, Rust, VHDL, Web

Targets: Architectures, Classes, Files, Functions, Modules, Packages, Project, Subprograms

Number of blank lines.

Also known as Blank Lines of Code (BLOC).



Targets By Language:

- **Ada:** Project, File, Type, Entry, Function, Package, Procedure, Protected, Task
- **Basic:** Project, File, Method, Module, Class
- **C#:** Project, File, Class, Method
- **C++:** Project, File, Class, Struct, Union, Function
- **Fortran:** Project, File, Module, Block Data, Function, Program, Subroutine
- **Java:** Project, File, Class, Interface, Method
- **Jovial:** Project, File, Module, Subroutine
- **Pascal:** Project, File, Class, Interface, Compunit, Function, Procedure
- **Python:** Project, File, Class, Function
- **Rust:** Project, File, Function
- **VHDL:** Project, File, Procedure, Function, Process, Architecture
- **Web:** Project, File, PHP Class, PHP Interface

Configuration:

- This metric can be disabled from "Project Configuration/Metrics/Lines and Statements" with the "Show line count metrics" option.

- This metric can be disabled from "Project Configuration/Metrics/Lines and Statements" with the "Include inactive lines" option for languages that have inactive lines.
- For projects and architectures, this metric is enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show default summary metrics" option.

Blank Lines (HTML)

API ID: CountLineBlankHtml
Languages: Web
Targets: Architectures, Files, Project

Number of blank HTML lines.

Targets By Language:

- **Web:** Project, File

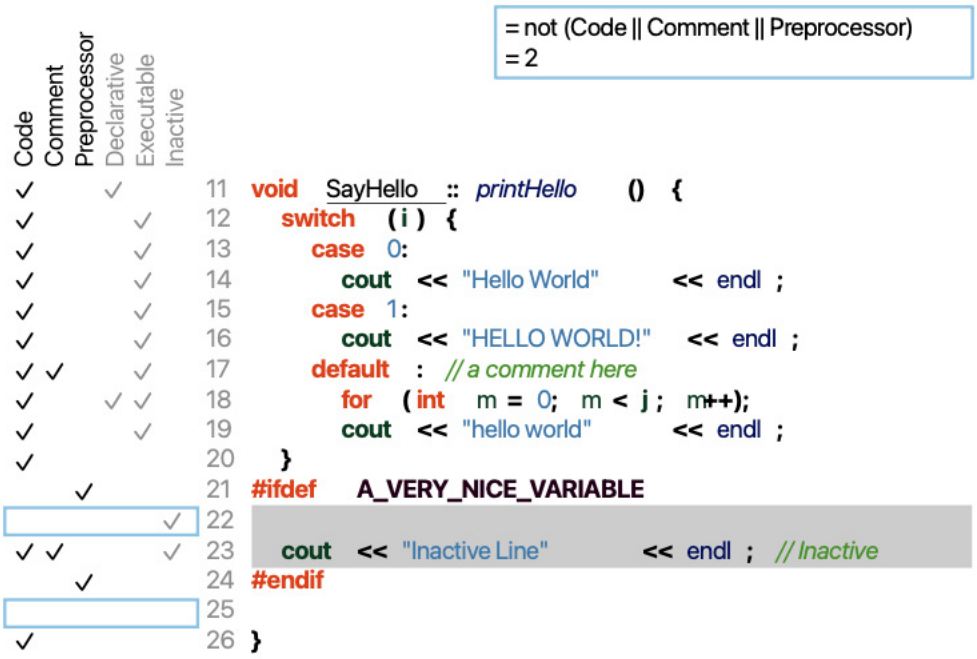
Configuration:

- This metric can be disabled from "Project Configuration/Metrics/Lines and Statements" with the "Show line count metrics" option.
- This metric can be enabled from "Project Configuration/Metrics/Lines and Statements" with the "Breakdown counts by web language" option.

Blank Lines (Includes Inactive)

API ID: CountLineBlankWithInactive
Languages: C++
Targets: Architectures, Classes, Files, Functions, Project

Number of blank lines, including inactive regions.



Targets By Language:

- **C++:** Project, File, Class, Struct, Union, Function

Configuration:

- This metric can be disabled from "Project Configuration/Metrics/Lines and Statements" with the "Show line count metrics" option.
- This metric can be enabled from "Project Configuration/Metrics/Lines and Statements" with the "Include inactive lines" option.

Blank Lines (JavaScript)

API ID: CountLineBlankJavascript
Languages: Web
Targets: Architectures, Files, Project

Number of blank JavaScript lines.

Targets By Language:

- **Web:** Project, File

Configuration:

- This metric can be disabled from "Project Configuration/Metrics/Lines and Statements" with the "Show line count metrics" option.

- This metric can be enabled from "Project Configuration/Metrics/Lines and Statements" with the "Breakdown counts by web language" option.

Blank Lines (PHP)

API ID: CountLineBlankPhp
Languages: Web
Targets: Architectures, Classes, Files, Project

Number of blank PHP lines.

Targets By Language:

- **Web:** Project, File, PHP Class, PHP Interface

Configuration:

- This metric can be disabled from "Project Configuration/Metrics/Lines and Statements" with the "Show line count metrics" option.
- This metric can be enabled from "Project Configuration/Metrics/Lines and Statements" with the "Breakdown counts by web language" option.

Changed Lines

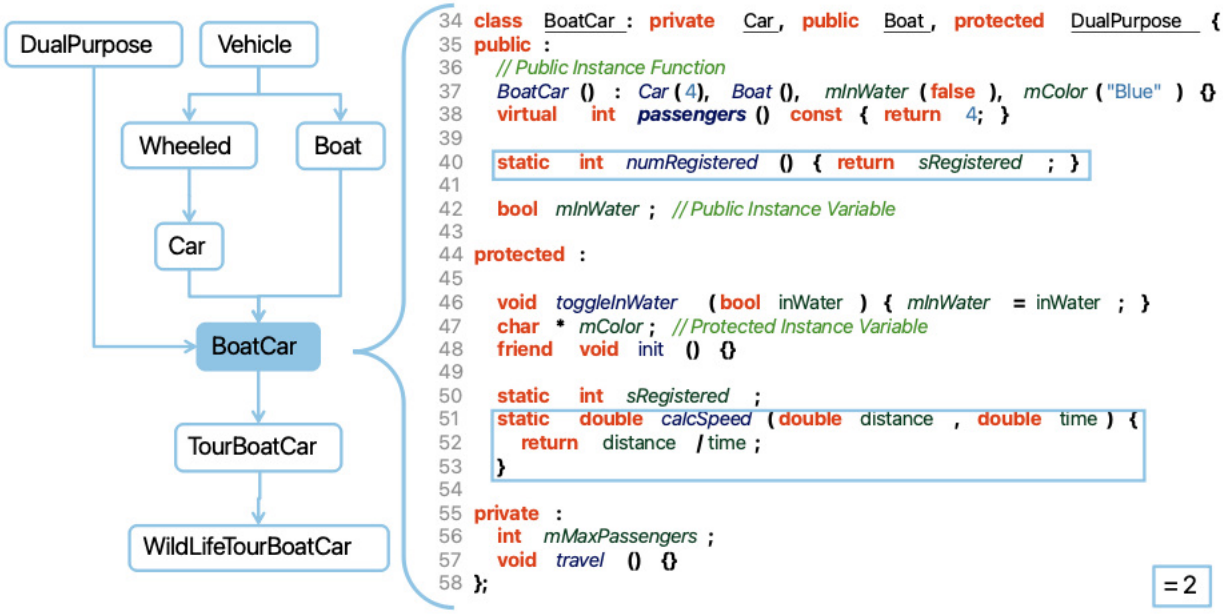
API ID: CountLineChanged
Languages: Ada, Assembly, Basic, C#, C++, Fortran, Java, Jovial, Pascal, Python, Rust, VHDL, Web
Targets: Classes, Files, Functions, Modules, Packages, Subprograms

Number of changed lines

Class Methods

API ID: CountDeclClassMethod
Languages: Basic, C#, C++, Java, Pascal, Web
Targets: Architectures, Classes, Files, Project

Number of class methods.



Targets By Language:

- **Basic:** Project, Class, Struct
- **C#:** Project, Class, Struct
- **C++:** Project, Class, Struct, Union
- **Java:** Project, File, Class, Interface
- **Pascal:** Project, Class, Interface
- **Web:** Project, PHP Class, PHP Interface

Configuration:

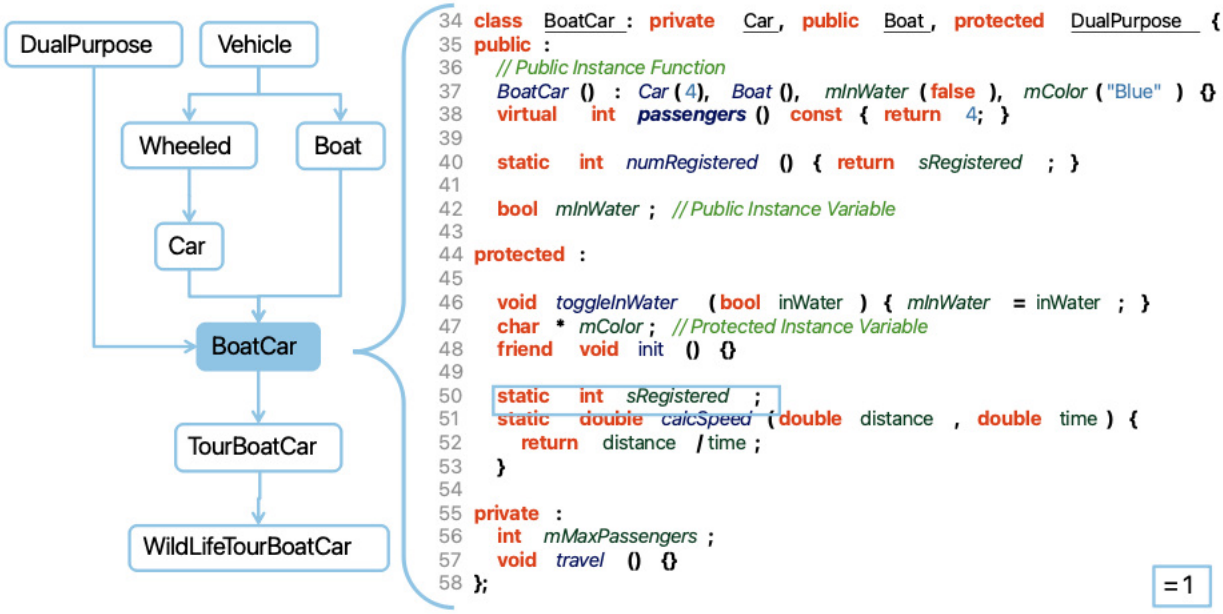
- For projects and architectures, this metric must also be enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show aggregated class metrics" option. Enabling this option can be time consuming.
- This metric can be disabled from "Project Configuration/Metrics/Object Orientated" with the "Show method and variable count metrics" option.

Class Variables

API ID: CountDeclClassVariable
Also Known As: Number of Variables (NV)
Languages: Basic, C#, C++, Java, Pascal, Web
Targets: Architectures, Classes, Files, Project

Number of class variables.

Also known as Lorenz & Kidd - Number of Variables (NV).



Targets By Language:

- **Basic:** Project, Class, Struct
- **C#:** Project, Class, Struct
- **C++:** Project, Class, Struct, Union
- **Java:** Project, File, Class, Interface
- **Pascal:** Project, Class, Interface
- **Web:** Project, PHP Class, PHP Interface

Configuration:

- For projects and architectures, this metric must also be enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show aggregated class metrics" option. Enabling this option can be time consuming.
- This metric can be disabled from "Project Configuration/Metrics/Object Orientated" with the "Show method and variable count metrics" option.

Classes

API ID: CountDeclClass
Languages: Basic, C#, C++, Java, Pascal, Python, Web
Targets: Architectures, Files, Project

Number of classes.

Targets By Language:

- **Basic:** Project, File
- **C#:** Project, File
- **C++:** Project, File
- **Java:** Project, File
- **Pascal:** Project, File
- **Python:** Project, File
- **Web:** Project, File

Configuration:

- For projects and architectures, this metric is enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show aggregated file metrics" option.
- For projects and architectures, this metric is enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show default summary metrics" option.

Code Files

API ID: CountDeclFileCode
Languages: C++
Targets: Architectures, Project

Number of code files.

Targets By Language:

- **C++:** Project

Configuration:

- For projects and architectures, this metric is enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show aggregated file metrics" option.

Code Lines

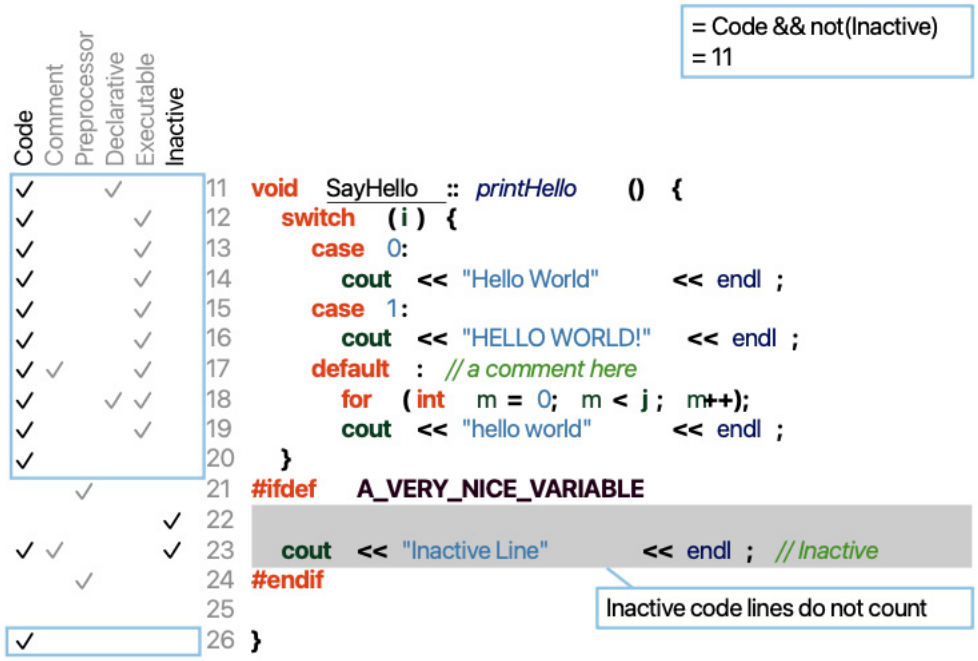
API ID: CountLineCode
Also Known As: Lines of Code (LOC), Source Lines of Code (SLOC)
Languages: Ada, Assembly, Basic, C#, C++, Fortran, Java, Jovial, Pascal, Python, Rust, VHDL, Web
Targets: Architectures, Classes, Files, Functions, Modules, Packages, Project,

Subprograms

Number of lines containing source code.

Note that a line can contain source and a comment and thus count towards multiple metrics. For classes this is the sum of CountLineCode for the member functions of the class.

Also known as Lines of Code (LOC); Source Lines of Code (SLOC).



Targets By Language:

- **Ada:** Project, File, Type, Entry, Function, Package, Procedure, Protected, Task
- **Basic:** Project, File, Method, Module, Class
- **C#:** Project, File, Class, Method
- **C++:** Project, File, Class, Struct, Union, Function
- **Fortran:** Project, File, Module, Block Data, Function, Program, Subroutine
- **Java:** Project, File, Class, Interface, Method
- **Jovial:** Project, File, Module, Subroutine
- **Pascal:** Project, File, Class, Interface, Compunit, Function, Procedure
- **Python:** Project, File, Class, Function
- **Rust:** Project, File, Function
- **VHDL:** Project, File, Procedure, Function, Process, Architecture
- **Web:** Project, File, PHP Class, PHP Interface

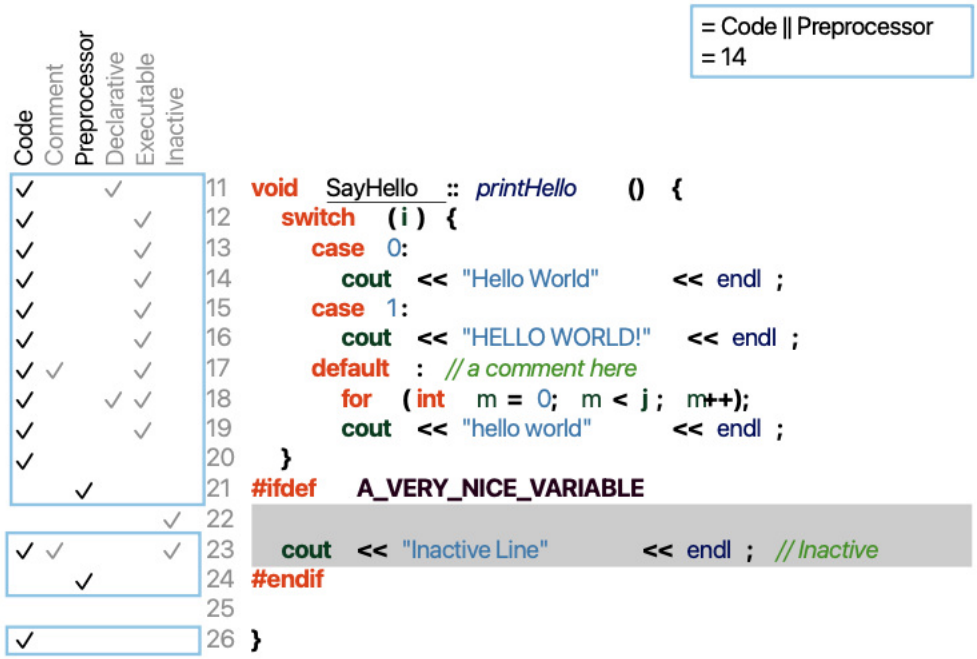
Configuration:

- This metric can be disabled from "Project Configuration/Metrics/Lines and Statements" with the "Show line count metrics" option.
- This metric can be disabled from "Project Configuration/Metrics/Lines and Statements" with the "Include inactive lines" option for languages that have inactive lines.
- For projects and architectures, this metric is enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show default summary metrics" option.

Code Lines (Includes Inactive)

API ID: CountLineCodeWithInactive
Languages: C++
Targets: Architectures, Classes, Files, Functions, Project

Number of lines containing source code, including inactive regions.



Targets By Language:

- **C++:** Project, File, Class, Struct, Union, Function

Configuration:

- This metric can be disabled from "Project Configuration/Metrics/Lines and Statements" with the "Show line count metrics" option.

- This metric can be enabled from "Project Configuration/Metrics/Lines and Statements" with the "Include inactive lines" option.

Code Lines (JavaScript)

API ID: CountLineCodeJavascript
Languages: Web
Targets: Architectures, Files, Project

Number of JavaScript lines containing source code.

Targets By Language:

- **Web:** Project, File

Configuration:

- This metric can be disabled from "Project Configuration/Metrics/Lines and Statements" with the "Show line count metrics" option.
- This metric can be enabled from "Project Configuration/Metrics/Lines and Statements" with the "Breakdown counts by web language" option.

Code Lines (PHP)

API ID: CountLineCodePhp
Languages: Web
Targets: Architectures, Classes, Files, Project

Number of PHP lines containing source code.

Targets By Language:

- **Web:** Project, File, PHP Class, PHP Interface

Configuration:

- This metric can be disabled from "Project Configuration/Metrics/Lines and Statements" with the "Show line count metrics" option.
- This metric can be enabled from "Project Configuration/Metrics/Lines and Statements" with the "Breakdown counts by web language" option.

CodeCheck Violation Density by Code Lines

API ID: CCViolDensityCode

Languages: Any
Targets: Architectures, Files, Functions, Project

The density of CodeCheck Violations by Lines of Code

This metric is calculated as Number of CodeCheck Violations divided by CountLineCode. See also CodeCheck Violation Density by Lines.

CodeCheck Violation Density by Lines

API ID: CCViolDensityLine
Languages: Any
Targets: Architectures, Files, Functions, Project

The density of CodeCheck Violations by Line

This metric is calculated as Number of CodeCheck Violations divided by CountLine. See also CodeCheck Violation Density by Code Lines.

CodeCheck Violation Types

API ID: CountCCViolType
Languages: Any
Targets: Architectures, Files, Functions, Project

The number of distinct violation types (distinct check ids)

For example, if the violation browser for a file showed these violations:

- non-void function does not return a value
- Remove unused global object `name`
- Remove unused local object dummy
- 'header.h' file not found.

The violation check ids are CPP_WARN_RETURN_TYPE, STI_UNUSED, STI_UNUSED, and UND_ERROR. UND_ERROR is an analysis error and not part of CodeCheck, so the value of this metric is 2 for CPP_WARN_RETURN_TYPE and STI_UNUSED.

See also CountAnalysisError for a count of analysis errors and CountAnalysisWarning for a count of analysis warnings.

CodeCheck Violations

API ID: CountCCViol
Languages: Any
Targets: Architectures, Files, Functions, Project

The number of CodeCheck Violations.

For example, if the violation browser for a file showed these violations:

- non-void function does not return a value
- Remove unused global object `name`
- Remove unused local object dummy
- 'header.h' file not found.

Then the value of this metric would be 3 because the fourth violation is an analysis error and this metric only counts CodeCheck violations.

See also CountAnalysisError for a count of analysis errors and CountAnalysisWarning for a count of analysis warnings.

Comment Lines

API ID: CountLineComment
Also Known As: Comment Lines of Code (CLOC)
Languages: Ada, Assembly, Basic, C#, C++, Fortran, Java, Jovial, Pascal, Python, Rust, VHDL, Web
Targets: Architectures, Classes, Files, Functions, Modules, Packages, Project, Subprograms

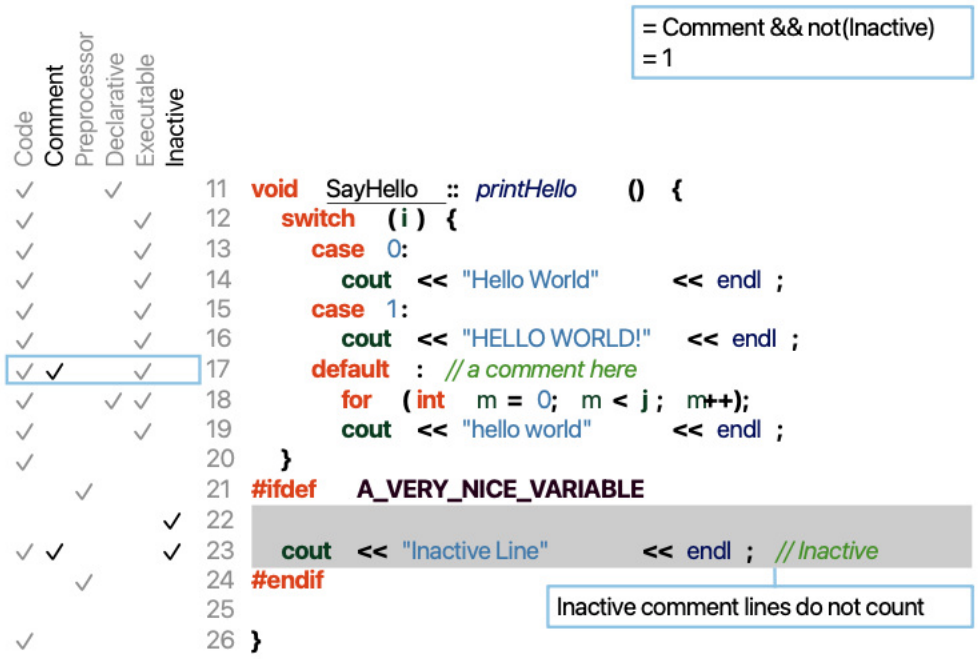
Number of lines containing comment.

This can overlap with other code counting metrics. For instance:

int j = 1; // comment

has a comment, is a source line, is an executable source line, and is a declarative source line.

Also known as Comment Lines of Code (CLOC).



Targets By Language:

- **Ada:** Project, File, Type, Entry, Function, Package, Procedure, Protected, Task
- **Basic:** Project, File, Method, Module, Class
- **C#:** Project, File, Class, Method
- **C++:** Project, File, Class, Struct, Union, Function
- **Fortran:** Project, File, Module, Block Data, Function, Program, Subroutine
- **Java:** Project, File, Class, Interface, Method
- **Jovial:** Project, File, Module, Subroutine
- **Pascal:** Project, File, Class, Interface, Compunit, Function, Procedure
- **Python:** Project, File, Class, Function
- **Rust:** Project, File, Function
- **VHDL:** Project, File, Procedure, Function, Process, Architecture
- **Web:** Project, File, PHP Class, PHP Interface

Configuration:

- This metric can be disabled from "Project Configuration/Metrics/Lines and Statements" with the "Show line count metrics" option.
- This metric can be disabled from "Project Configuration/Metrics/Lines and Statements" with the "Include inactive lines" option for languages that have inactive lines.
- For projects and architectures, this metric is enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show default summary metrics" option.

Comment Lines (HTML)

API ID: CountLineCommentHtml
Languages: Web
Targets: Architectures, Files, Project

Number of HTML lines containing comment.

Targets By Language:

- **Web:** Project, File

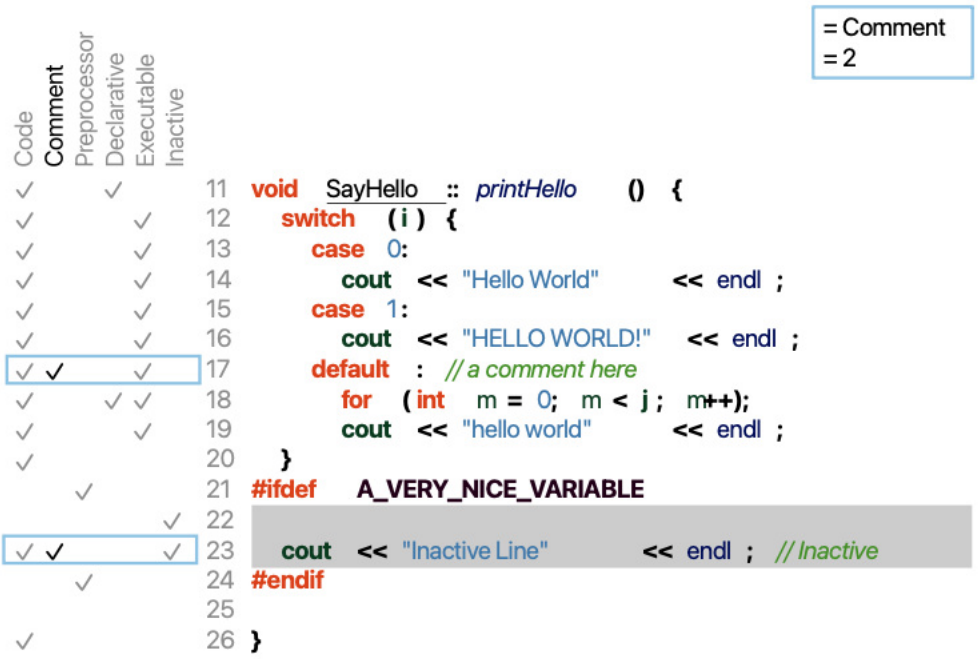
Configuration:

- This metric can be disabled from "Project Configuration/Metrics/Lines and Statements" with the "Show line count metrics" option.
- This metric can be enabled from "Project Configuration/Metrics/Lines and Statements" with the "Breakdown counts by web language" option.

Comment Lines (Includes Inactive)

API ID: CountLineCommentWithInactive
Languages: C++
Targets: Architectures, Classes, Files, Functions, Project

Number of lines containing comment, including inactive regions.



Targets By Language:

- **C++:** Project, File, Class, Struct, Union, Function

Configuration:

- This metric can be disabled from "Project Configuration/Metrics/Lines and Statements" with the "Show line count metrics" option.
- This metric can be enabled from "Project Configuration/Metrics/Lines and Statements" with the "Include inactive lines" option.

Comment Lines (JavaScript)

API ID: CountLineCommentJavascript
Languages: Web
Targets: Architectures, Files, Project

Number of JavaScript lines containing comment.

Targets By Language:

- **Web:** Project, File

Configuration:

- This metric can be disabled from "Project Configuration/Metrics/Lines and Statements" with the "Show line count metrics" option.
- This metric can be enabled from "Project Configuration/Metrics/Lines and Statements" with the "Breakdown counts by web language" option.

Comment Lines (PHP)

API ID: CountLineCommentPhp
Languages: Web
Targets: Architectures, Classes, Files, Project

Number of PHP lines containing comment.

Targets By Language:

- **Web:** Project, File, PHP Class, PHP Interface

Configuration:

- This metric can be disabled from "Project Configuration/Metrics/Lines and

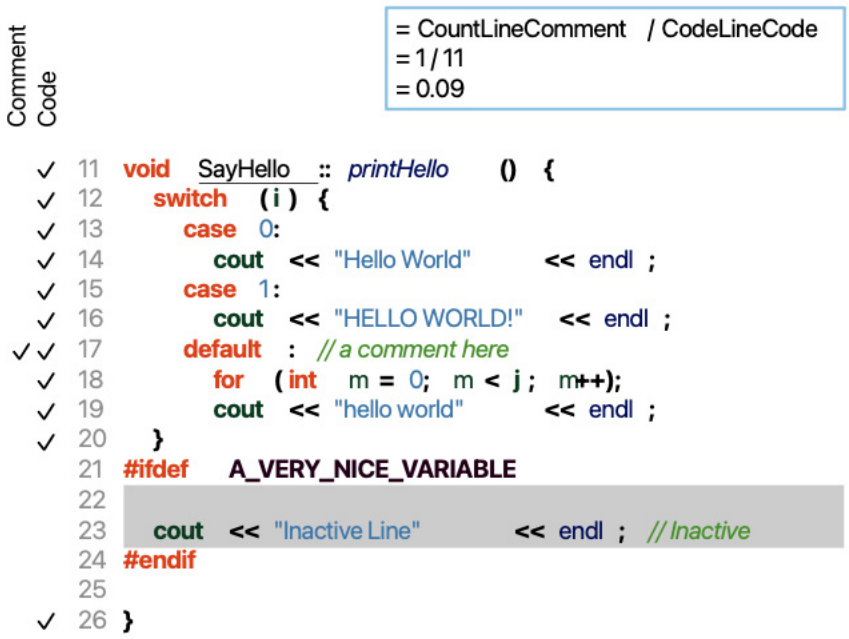
Statements" with the "Show line count metrics" option.
- This metric can be enabled from "Project Configuration/Metrics/Lines and Statements" with the "Breakdown counts by web language" option.

Comment to Code Ratio

API ID: RatioCommentToCode
Languages: Ada, Assembly, Basic, C#, C++, Fortran, Java, Jovial, Pascal, Python, Rust, VHDL, Web
Targets: Architectures, Classes, Files, Functions, Modules, Packages, Project, Subprograms

Ratio of comment lines to code lines.

Note that because some lines are both code and comment, this could easily yield percentages higher than 100.



Targets By Language:

- **Ada:** Project, File, Type, Entry, Function, Package, Procedure, Protected, Task
- **Basic:** Project, File, Method, Module, Class
- **C#:** Project, File, Class, Method
- **C++:** Project, File, Class, Struct, Union, Function
- **Fortran:** Project, File, Block Data, Function, Program, Subroutine
- **Java:** Project, File, Class, Interface, Method

- **Jovial:** Project, File, Module, Subroutine
- **Pascal:** Project, File, Class, Interface, Compunit, Function, Procedure
- **Python:** Project, File, Class, Function
- **Rust:** Project, File, Function
- **VHDL:** Project, File, Procedure, Function, Architecture
- **Web:** Project, File

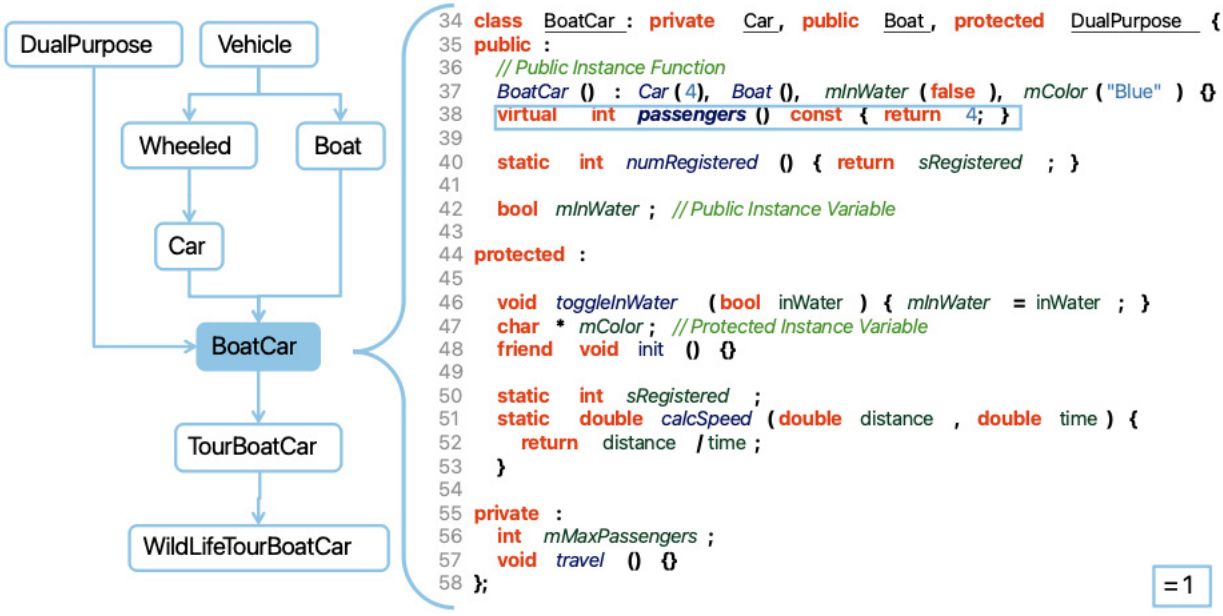
Configuration:

- This metric can be disabled from "Project Configuration/Metrics/Lines and Statements" with the "Show line count metrics" option.
- For projects and architectures, this metric is enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show default summary metrics" option.

Const Methods

API ID: CountDeclMethodConst
Languages: C++
Targets: Architectures, Classes, Project

Number of local const methods.



Targets By Language:

- **C++:** Project, Class, Struct, Union

Configuration:

- For projects and architectures, this metric must also be enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show aggregated class metrics" option. Enabling this option can be time consuming.
- This metric can be disabled from "Project Configuration/Metrics/Object Orientated" with the "Show method and variable count metrics" option.

Coupled Classes

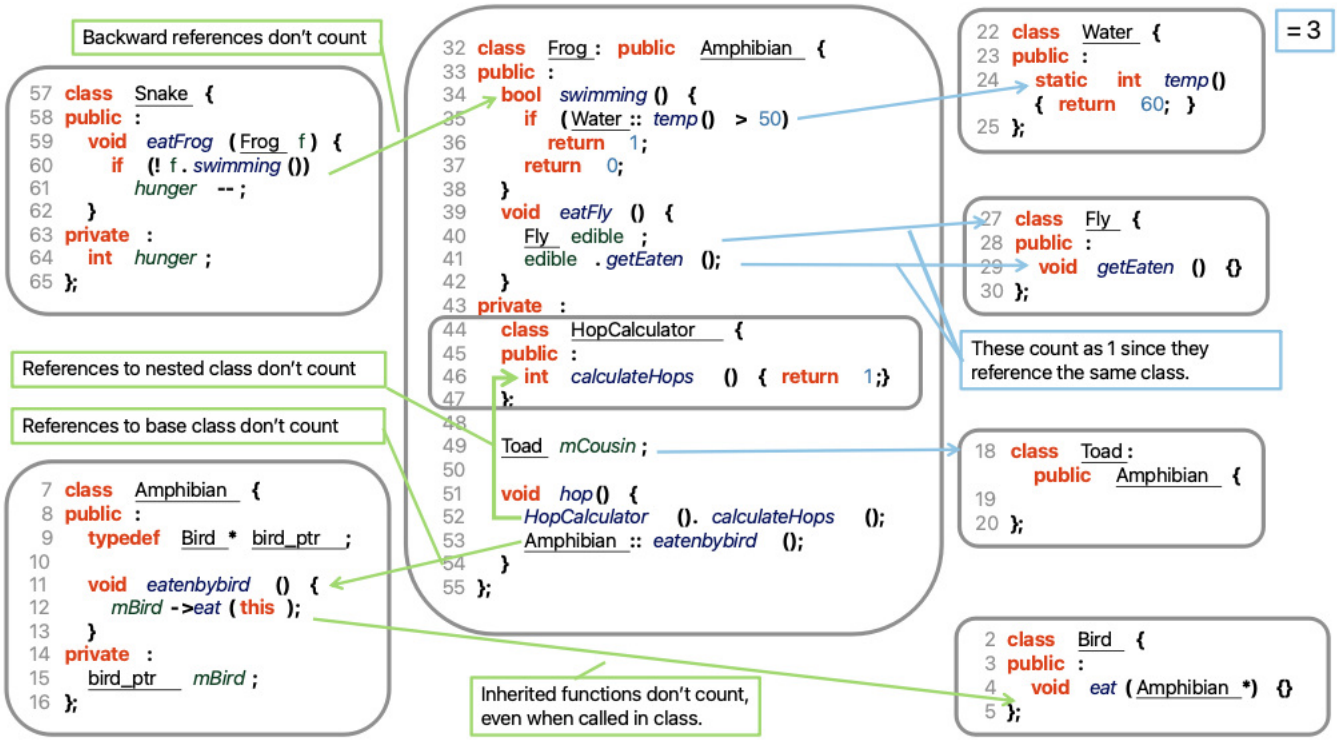
API ID: CountClassCoupled
Also Known As: Coupling Between Objects (CBO)
Languages: Basic, C#, C++, Java, Pascal, Python
Targets: Classes

Number of other classes coupled to.

The Coupling Between Object Classes (CBO) measure for a class is a count of the number of other classes to which it is coupled. Base classes and nested classes are not counted. Class A is coupled to class B if class A uses a type, data, or member from class B. This metric is also referred to as Efferent Coupling (Ce). Any number of couplings to a given class counts as 1 towards the metric total.

Chidamber & Kemerer suggest that:
1) Excessive coupling between object classes is detrimental to modular design and prevents reuse.
2) Inter-object class couples should be kept to a minimum.
3) The higher the inter-object class coupling, the more rigorous testing needs to be.

Also known as Chidamber & Kemerer - Coupling Between Objects (CBO).



Targets By Language:

- **Basic:** Class, Struct
- **C#:** Class, Struct
- **C++:** Class, Struct, Union
- **Java:** Class, Interface
- **Pascal:** Class, Interface
- **Python:** Class

Configuration:

- This metric can be disabled from "Project Configuration/Metrics/Object Orientated" with the "Show coupling and cohesion metrics" option.

Coupled Classes Modified

API ID: CountClassCoupledModified
Languages: Basic, C#, Java, Pascal, Python
Targets: Classes

Number of other non-standard classes coupled to.

Targets By Language:

- **Basic:** Class, Struct
- **C#:** Class, Struct
- **Java:** Class, Interface

- **Pascal:** Class, Interface
- **Python:** Class

Configuration:

- This metric can be disabled from "Project Configuration/Metrics/Object Orientated" with the "Show coupling and cohesion metrics" option.

Coupled Packages

API ID: CountPackageCoupled
Languages: Ada
Targets: Packages

Number of other packages coupled to.

Targets By Language:

- **Ada:** Package

Configuration:

- This metric can be disabled from "Project Configuration/Metrics/Object Orientated" with the "Show coupling and cohesion metrics" option.

Cyclomatic Complexity

API ID: Cyclomatic
Also Known As: Cyclomatic Complexity (CC)
Languages: Ada, Assembly, Basic, C#, C++, Fortran, Java, Jovial, Pascal, Python, Rust, VHDL, Web
Targets: Files, Functions, Modules, Packages, Subprograms

McCabe Cyclomatic Complexity, the number of decision points + 1.

McCabe Cyclomatic complexity as per the original NIST paper on the subject. The cyclomatic complexity of any structured program with only one entrance point and one exit point is equal to the number of decision points contained in that program plus one. Understand counts the keywords for decision points (FOR, WHILE, etc) and then adds 1. For a switch statement, each 'case' is counted as 1. For languages with macros, the expanded macro text is also included in the calculation.

Also known as McCabe - Cyclomatic Complexity (CC).

Targets By Language:

- **Ada:** Type, Entry, Function, Package, Procedure, Protected, Task
- **Basic:** Method
- **C#:** Method
- **C++:** Function
- **Fortran:** Module, Block Data, Function, Program, Subroutine
- **Java:** Method
- **Jovial:** Subroutine
- **Pascal:** Compunit, Function, Procedure
- **Python:** File, Function
- **Rust:** Function
- **VHDL:** Procedure, Function, Process
- **Web:** File

Configuration:

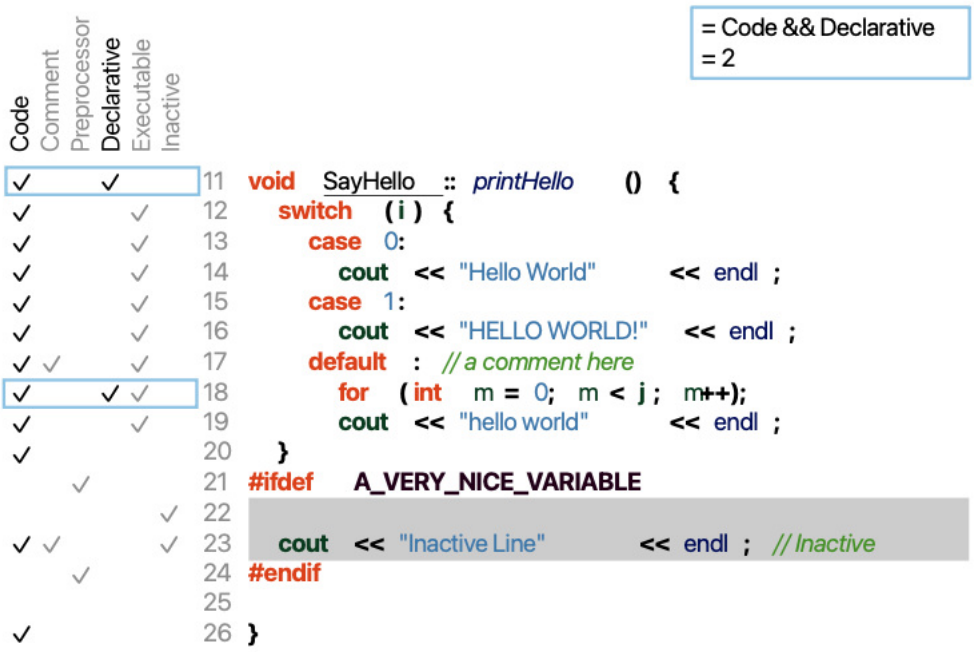
- The cyclomatic metric used is configured from "Project Configuration/ Metrics/Complexity". This metric results from having "Count each decision in a multi-way-decision structure" on and having "Count logical conjuctions (OR) and logical and (AND)" off. Checking "Show all available cyclomatic complexity metric variants" will also enable the metric.

Declarative Code Lines

API ID: CountLineCodeDecl
Languages: Ada, Assembly, C#, C++, Fortran, Java, Pascal, Python

Targets: Architectures, Classes, Files, Functions, Modules, Packages, Project, Subprograms

Number of lines containing declarative source code. Note that a line can be declarative and executable - "int i =0;" for instance.



Targets By Language:

- **Ada:** Project, File, Type, Entry, Function, Package, Procedure, Protected, Task
- **C#:** Project, File, Class, Method
- **C++:** Project, File, Class, Struct, Union, Function
- **Fortran:** Project, File, Module, Block Data, Function, Program, Subroutine
- **Java:** Project, File, Class, Interface, Method
- **Pascal:** Project, File, Class, Interface, Compunit, Function, Procedure
- **Python:** Project, File, Class, Function

Configuration:

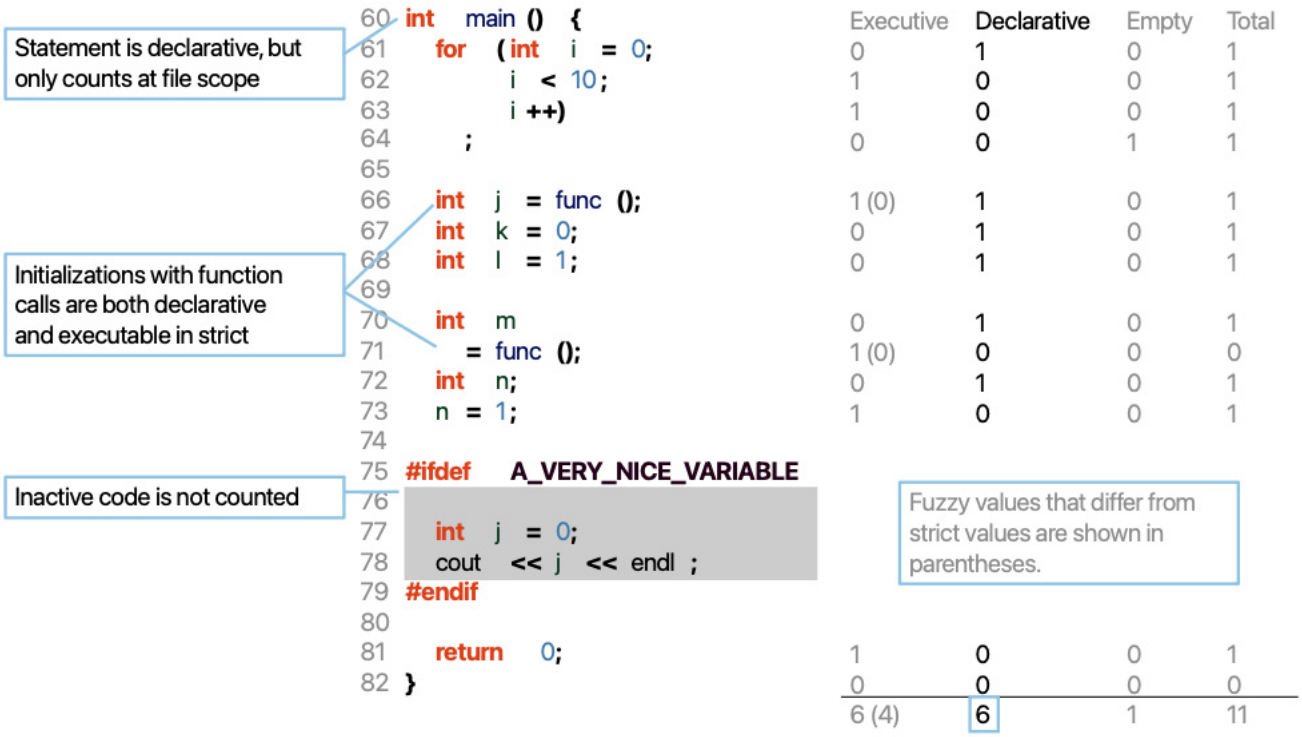
- This metric can be disabled from "Project Configuration/Metrics/Lines and Statements" with the "Show line count metrics" option.

Declarative Statements

API ID: CountStmtDecl
Languages: Ada, Assembly, Basic, C#, C++, Fortran, Java, Jovial, Pascal, Python,

VHDL, Web
Targets: Architectures, Classes, Files, Functions, Modules, Packages, Project, Subprograms

Number of declarative statements.



Targets By Language:

- **Ada:** Project, File, Type, Entry, Function, Package, Procedure, Protected, Task
- **Basic:** Project, File, Method, Module, Class
- **C#:** Project, File, Class, Method
- **C++:** Project, File, Class, Struct, Union, Function
- **Fortran:** Project, File, Module, Block Data, Function, Program, Subroutine
- **Java:** Project, File, Class, Interface, Method
- **Jovial:** Project, File, Module, Subroutine
- **Pascal:** Project, File, Class, Interface, Compunit, Function, Procedure
- **Python:** Project, File, Class, Function
- **VHDL:** Project, File
- **Web:** Project, File, PHP Class, PHP Interface

Configuration:

- This metric can be disabled from "Project Configuration/Metrics/Lines and Statements" with the "Show statement count metrics" option.
- For projects and architectures, this metric is enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show default

summary metrics" option.

Declarative Statements (Javascript)

API ID: CountStmtDeclJavascript
Languages: Web
Targets: Architectures, Files, Project

Number of JavaScript declarative statements.

Targets By Language:

- **Web:** Project, File

Configuration:

- This metric can be disabled from "Project Configuration/Metrics/Lines and Statements" with the "Show statement count metrics" option.
- This metric can be enabled from "Project Configuration/Metrics/Lines and Statements" with the "Breakdown counts by web language" option.

Declarative Statements (PHP)

API ID: CountStmtDeclPhp
Languages: Web
Targets: Architectures, Classes, Files, Project

Number of PHP declarative statements.

Targets By Language:

- **Web:** Project, File, PHP Class, PHP Interface

Configuration:

- This metric can be disabled from "Project Configuration/Metrics/Lines and Statements" with the "Show statement count metrics" option.
- This metric can be enabled from "Project Configuration/Metrics/Lines and Statements" with the "Breakdown counts by web language" option.

Default Methods

API ID: CountDeclMethodDefault
Languages: Java

Targets: Architectures, Classes, Files, Project

Number of local default methods.

Targets By Language:

- **Java:** Project, File, Class, Interface

Configuration:

- For projects and architectures, this metric must also be enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show aggregated class metrics" option. Enabling this option can be time consuming.
- This metric can be disabled from "Project Configuration/Metrics/Object Orientated" with the "Show method and variable count metrics" option.

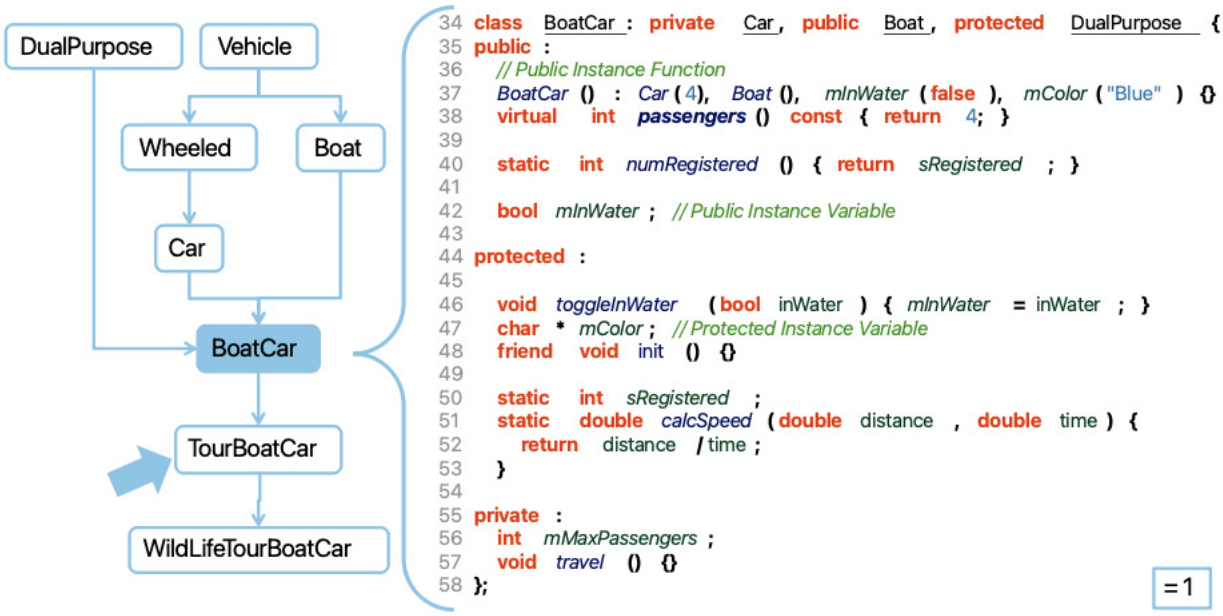
Derived Classes

API ID: CountClassDerived
Also Known As: Number of Children (NOC)
Languages: Basic, C#, C++, Java, Pascal, Python, Web
Targets: Classes

Number of immediate subclasses.

(i.e. the number of classes one level down the inheritance tree from this class).

Also known as Chidamber & Kemerer - Number of Children (NOC).



Targets By Language:

- **Basic:** Class, Struct
- **C#:** Class, Struct
- **C++:** Class, Struct, Union
- **Java:** Class, Interface
- **Pascal:** Class, Interface
- **Python:** Class
- **Web:** PHP Class, PHP Interface

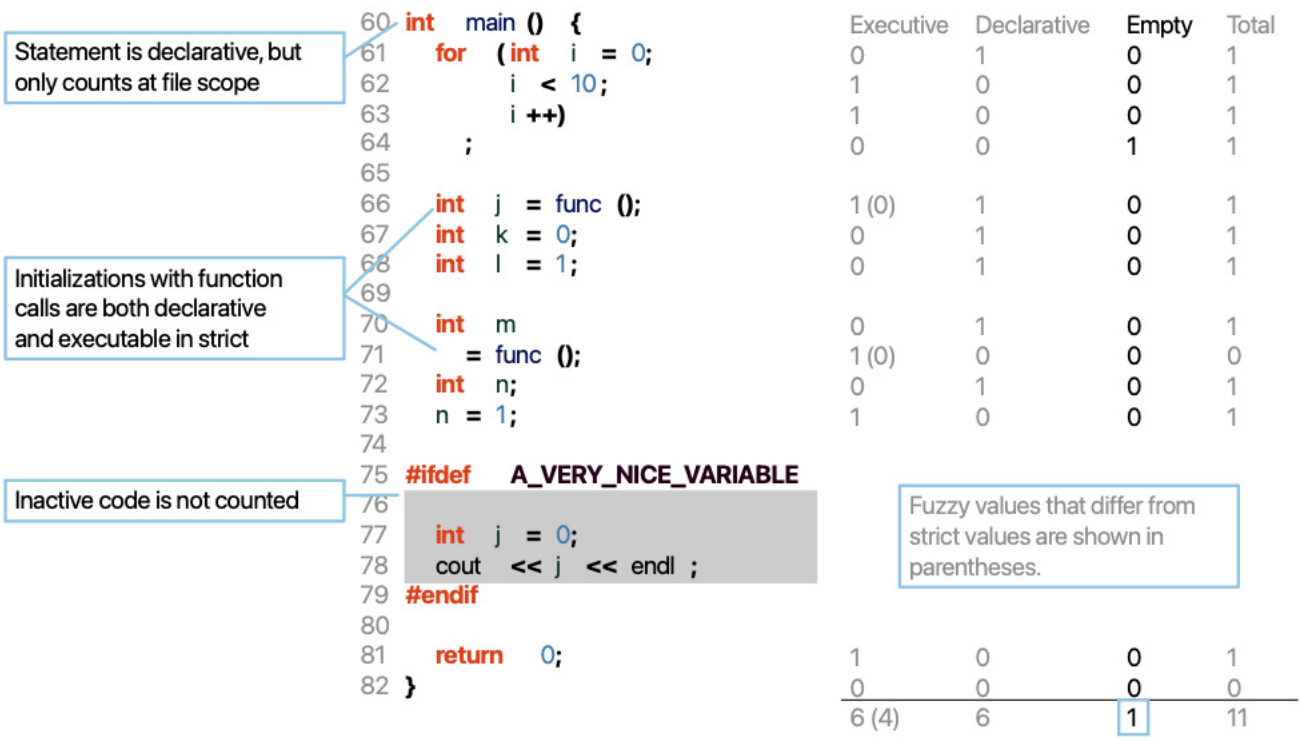
Configuration:

- This metric can be disabled from "Project Configuration/Metrics/Object Orientated" with the "Show inheritance metrics" option.

Empty Statements

API ID: CountStmtEmpty
Languages: C++
Targets: Architectures, Classes, Files, Functions, Project

Number of empty statements.



Targets By Language:

- **C++:** Project, File, Class, Struct, Union, Function

Configuration:

- This metric can be disabled from "Project Configuration/Metrics/Lines and Statements" with the "Show statement count metrics" option.

Essential Complexity

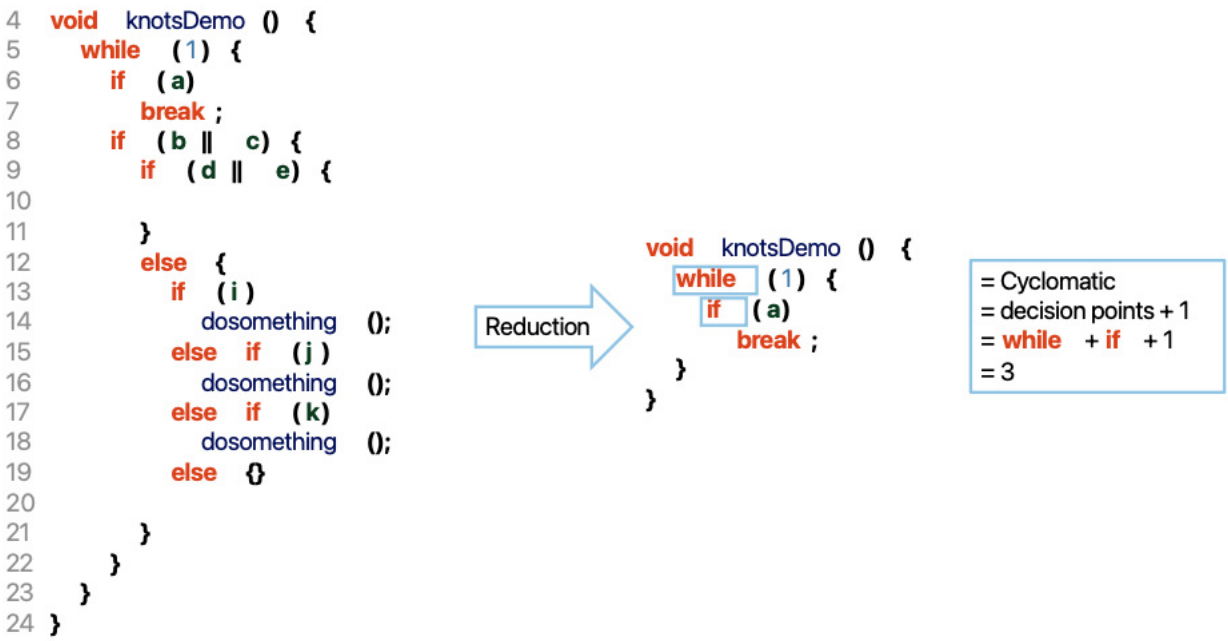
API ID: Essential
Also Known As: ev(G)
Languages: Ada, Basic, C#, C++, Fortran, Java, Jovial, Pascal, Python, Rust, Web
Targets: Files, Functions, Modules, Packages, Subprograms

The number of decision points + 1 after control graph reduction.

Essential complexity is the cyclomatic complexity after iteratively replacing all well structured control structures with a single statement. Structures such as if-then-else and while loops are considered well structured. Understand calculates the essential complexity by removing all the structured subgraphs from the control graph and then calculating the complexity. A graph that has only the regular single entry/single exit loops or branches will be reducible to a graph with complexity one. Any branches into or out of a loop or decision will make the graph non-reducible and will have Essential Complexity > 2. (You never get 2 since a graph with complexity 2

is always reducible to a graph with complexity 1)

Also known as McCabe - ev(G).



Targets By Language:

- **Ada:** Type, Entry, Function, Package, Procedure, Protected, Task
- **Basic:** Method
- **C#:** Method
- **C++:** Function
- **Fortran:** Module, Block Data, Function, Program, Subroutine
- **Java:** Method
- **Jovial:** Subroutine
- **Pascal:** Compunit, Function, Procedure
- **Python:** File, Function
- **Rust:** File, Function
- **Web:** File

Configuration:

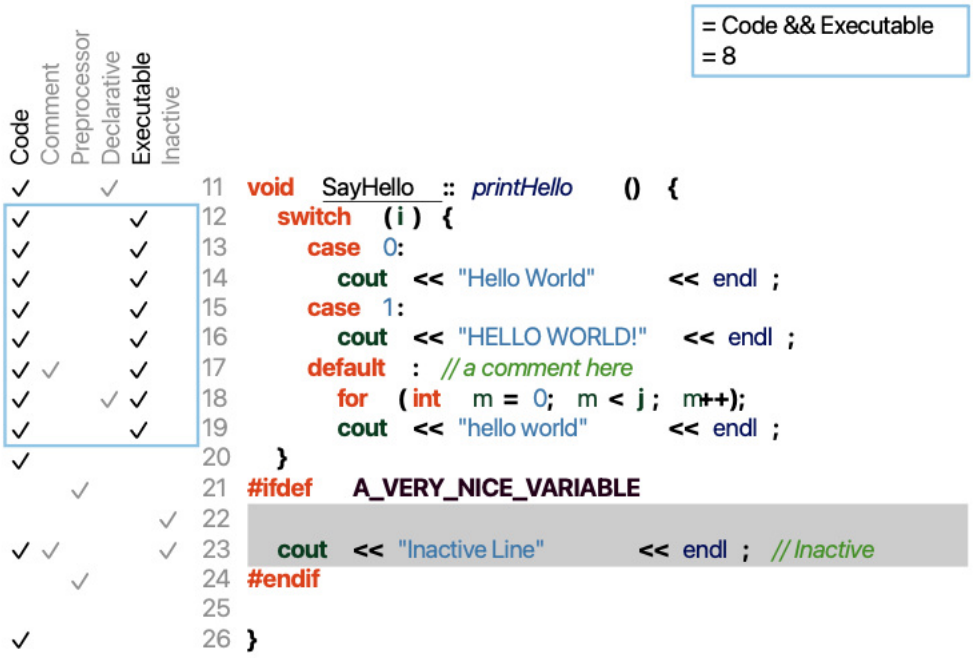
- This metric can be enabled from "Project Configuration/Metrics/Complexity" with the "Essential" option.

Executable Code Lines

API ID: CountLineCodeExe

Languages: Ada, Assembly, C#, C++, Fortran, Java, Pascal, Python
Targets: Architectures, Classes, Files, Functions, Modules, Packages, Project, Subprograms

Number of lines containing executable source code.



Targets By Language:

- **Ada:** Project, File, Type, Entry, Function, Package, Procedure, Protected, Task
- **C#:** Project, File, Class, Method
- **C++:** Project, File, Class, Struct, Union, Function
- **Fortran:** Project, File, Module, Block Data, Function, Program, Subroutine
- **Java:** Project, File, Class, Interface, Method
- **Pascal:** Project, File, Class, Interface, Compunit, Function, Procedure
- **Python:** Project, File, Class, Function

Configuration:

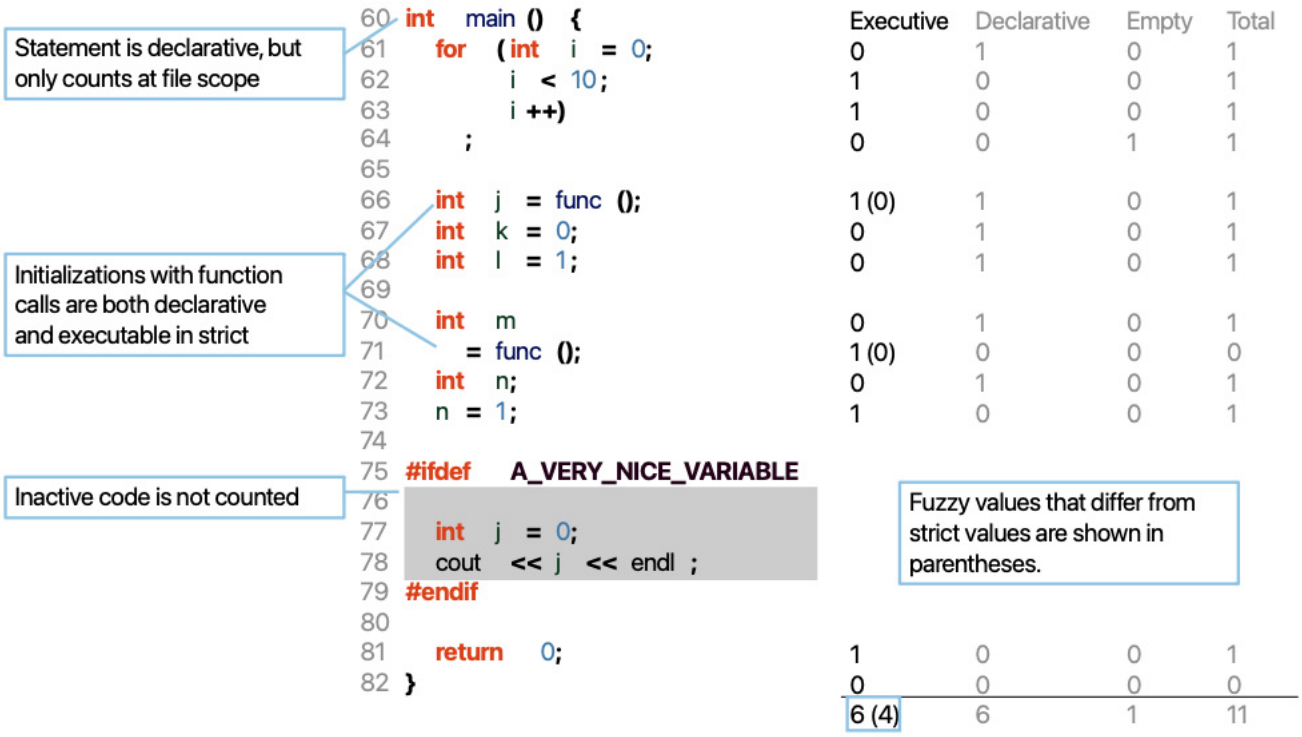
- This metric can be disabled from "Project Configuration/Metrics/Lines and Statements" with the "Show line count metrics" option.

Executable Statements

API ID: CountStmtExe
Languages: Ada, Assembly, Basic, C#, C++, Fortran, Java, Jovial, Pascal, Python,

VHDL, Web
Targets: Architectures, Classes, Files, Functions, Modules, Packages, Project, Subprograms

Number of executable statements.



Targets By Language:

- **Ada:** Project, File, Type, Entry, Function, Package, Procedure, Protected, Task
- **Basic:** Project, File, Method, Module, Class
- **C#:** Project, File, Class, Method
- **C++:** Project, File, Class, Struct, Union, Function
- **Fortran:** Project, File, Module, Block Data, Function, Program, Subroutine
- **Java:** Project, File, Class, Interface, Method
- **Jovial:** Project, File, Subroutine
- **Pascal:** Project, File, Class, Interface, Compunit, Function, Procedure
- **Python:** Project, File, Class, Function
- **VHDL:** Project, File
- **Web:** Project, File, PHP Class, PHP Interface

Configuration:

- This metric can be disabled from "Project Configuration/Metrics/Lines and Statements" with the "Show statement count metrics" option.
- For projects and architectures, this metric is enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show default

summary metrics" option.

Executable Statements (JavaScript)

API ID: CountStmtExeJavascript
Languages: Web
Targets: Architectures, Files, Project

Number of JavaScript executable statements.

Targets By Language:

- **Web:** Project, File

Configuration:

- This metric can be disabled from "Project Configuration/Metrics/Lines and Statements" with the "Show statement count metrics" option.
- This metric can be enabled from "Project Configuration/Metrics/Lines and Statements" with the "Breakdown counts by web language" option.

Executable Statements (PHP)

API ID: CountStmtExePhp
Languages: Web
Targets: Architectures, Classes, Files, Project

Number of PHP executable statements.

Targets By Language:

- **Web:** Project, File, PHP Class, PHP Interface

Configuration:

- This metric can be disabled from "Project Configuration/Metrics/Lines and Statements" with the "Show statement count metrics" option.
- This metric can be enabled from "Project Configuration/Metrics/Lines and Statements" with the "Breakdown counts by web language" option.

Executable Units

API ID: CountDeclExecutableUnit
Languages: Ada, Basic, C#, Fortran, Java, Pascal, Python, Web

Targets: Architectures, Files, Project

Number of program units with executable code.

Targets By Language:

- **Ada:** Project, File
- **Basic:** Project, File
- **C#:** Project, File
- **Fortran:** Project, File
- **Java:** Project, File
- **Pascal:** Project, File
- **Python:** Project, File
- **Web:** Project, File

Configuration:

- For projects and architectures, this metric is enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show aggregated file metrics" option.

Files

API ID: CountDeclFile
Languages: Ada, Assembly, Basic, C#, C++, Fortran, Java, Jovial, Pascal, Python, Rust, VHDL, Web
Targets: Architectures, Project

Number of files.

Targets By Language:

- **Ada:** Project
- **Basic:** Project
- **C#:** Project
- **C++:** Project
- **Fortran:** Project
- **Java:** Project
- **Jovial:** Project
- **Pascal:** Project
- **Python:** Project
- **Rust:** Project
- **VHDL:** Project
- **Web:** Project

Configuration:

- For projects and architectures, this metric is enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show aggregated file metrics" option.
- For projects and architectures, this metric is enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show default summary metrics" option.

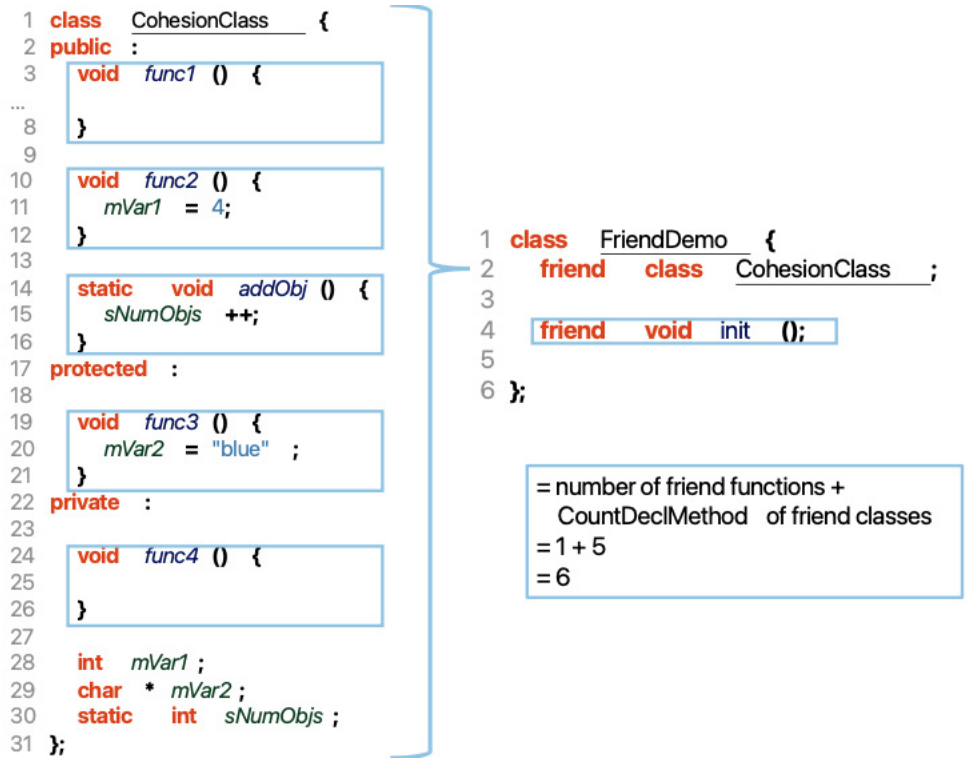
Friend Methods

API ID: CountDeclMethodFriend
Also Known As: Number of Friend Methods (NFM), Number of Friends (NF)
Languages: C++
Targets: Classes

Number of local (not inherited) friend methods.

The number of friend functions plus the CountDeclMethod of friend classes.

Also known as Lorenz & Kidd - Number of Friends (NF); Number of Friend Methods (NFM).



Targets By Language:

- **C++:** Class, Struct, Union

Configuration:

- This metric can be disabled from "Project Configuration/Metrics/Object Orientated" with the "Show method and variable count metrics" option.

Functions

API ID: CountDeclFunction
Languages: Assembly, C#, C++, Java, Python, Rust, Web
Targets: Architectures, Files, Project

Number of functions.

Targets By Language:

- **C#:** Project, File
- **C++:** Project, File
- **Java:** Project, File
- **Python:** Project, File
- **Rust:** Project, File
- **Web:** Project, File

Configuration:

- For projects and architectures, this metric is enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show aggregated file metrics" option.
- For projects and architectures, this metric is enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show default summary metrics" option.

Header Files

API ID: CountDeclFileHeader
Languages: C++
Targets: Architectures, Project

Number of header files.

Targets By Language:

- **C++:** Project

Configuration:

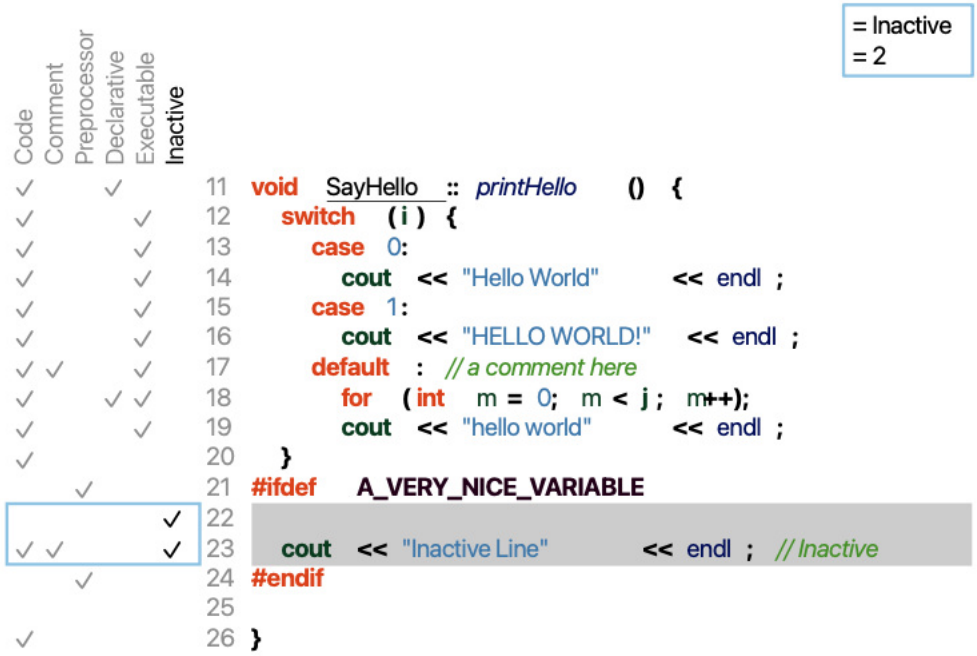
- For projects and architectures, this metric is enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show aggregated file metrics" option.

Inactive Lines

API ID: CountLineInactive
Languages: C#, C++, Pascal
Targets: Architectures, Classes, Files, Functions, Modules, Project, Subprograms

Number of inactive lines.

This is the number of lines that are inactive from the view of the preprocessor. In other words, they are on the FALSE side of a #if or #ifdef preprocessor directive.



Targets By Language:

- **C#:** Project, File, Class, Method
- **C++:** Project, File, Class, Struct, Union, Function
- **Pascal:** Project, File, Class, Interface, Compunit, Function, Procedure

Configuration:

- This metric can be disabled from "Project Configuration/Metrics/Lines and Statements" with the "Show line count metrics" option.
- For projects and architectures, this metric is enabled from "Project

Configuration/Metrics/Projects and Architectures" with the "Show default summary metrics" option.

Inputs

API ID: CountInput
Also Known As: FANIN (Infomational fan-in)
Languages: C#, C++, Fortran, Java
Targets: Functions, Subprograms

Number of calling subprograms plus global variables read.

The number of inputs a function uses plus the number of unique subprograms calling the function. Inputs include parameters and global variables that are used in the function, so Functions calledby + Parameters read + Global Variables read. Recursive function calls and local variables that are not class static variables are not included. Of the two general approaches to calculating FANIN (informational versus structural) ours is the informational approach.

Also known as FANIN (Infomational fan-in).

```
3  int  in  = 1;
4  int  out  = 1;
5
6  int  inOutFunc  (int  in1 , int  in2 , int  *inout1  , int  &inout2  , int * out1 , int  & out2 ) {
7    out  = in  + in1  + in2  + *inout1  + inout2  ;
8
9    *inout1  = in1 ;
10   inout2  = in2 ;
11
12   *out1  = in1 ;
13   out2  = in2 ;
14
15   in1  = somefunc  ();
16   in2  = 2;
17
18   int  randomint  = 3;
19   in1  = randomint  ;
20
21   return  4;
22 }
...
24 void  callingfunc  () {
25   int  a, b, c, d;
26   int  myval  = inOutFunc  (1, 2,& a, b,& c, d);
```

= functions called -by + parameters used + globals used
= 1 + 4 + 1
= 6

Entity	Counts?	Comment
in	Yes	Use line 7
out	No	Not used
in1	Yes	Use line 7, Use line 9, Use line 12
in2	Yes	Use line 7, Use line 10, Use line 13
inout1	Yes	Use line 7
inout2	Yes	Use line 7
out1	No	Not used
out2	No	Not used
randomint	No	Not a parameter, global, or class static variable
calledbyfunc	Yes	Line 26

Targets By Language:

- **C#:** Method
- **C++:** Function

- **Fortran:** Function, Program, Subroutine
- **Java:** Method

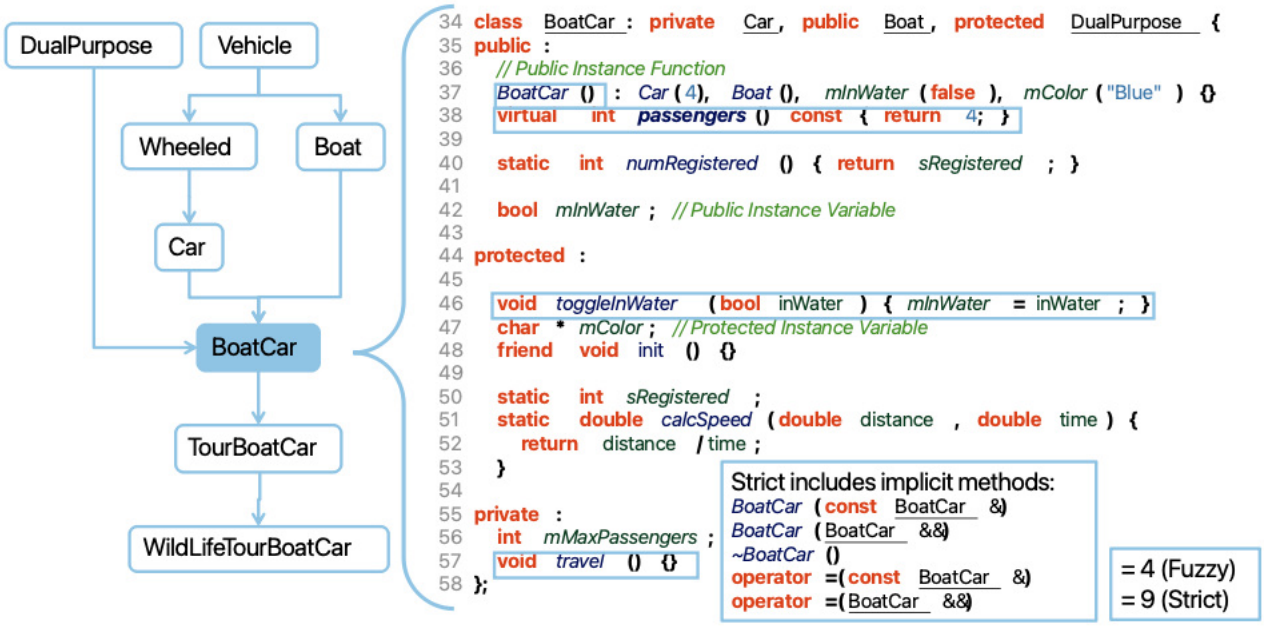
Instance Methods

API ID: CountDeclInstanceMethod
Also Known As: Number of Instance Methods (NIM)
Languages: Basic, C#, C++, Java, Pascal, Python, Web
Targets: Architectures, Classes, Files, Project

Number of instance methods.

Methods defined in a class that are only accessable through an object of that class.

Also known as Number of Instance Methods (NIM).



Targets By Language:

- **Basic:** Project, Class, Struct
- **C#:** Project, Class, Struct
- **C++:** Project, Class, Struct, Union
- **Java:** Project, File, Class, Interface
- **Pascal:** Project, Class, Interface
- **Python:** Project, Class
- **Web:** Project, PHP Class, PHP Interface

Configuration:

- For projects and architectures, this metric must also be enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show aggregated class metrics" option. Enabling this option can be time consuming.
- This metric can be disabled from "Project Configuration/Metrics/Object Orientated" with the "Show method and variable count metrics" option.

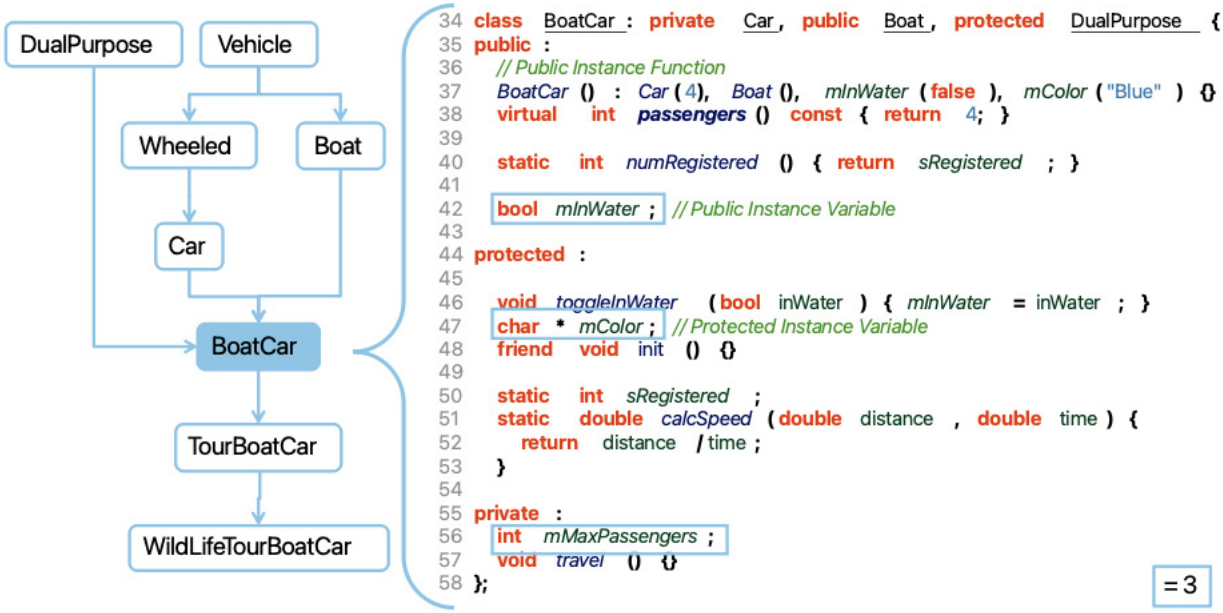
Instance Variables

API ID: CountDeclInstanceVariable
Also Known As: Number of Instance Variables (NIV)
Languages: Basic, C#, C++, Java, Pascal, Python, Web
Targets: Architectures, Classes, Files, Project

Number of instance variables.

Variables defined in a class that are only accessible through an object of that class.

Also known as Number of Instance Variables (NIV).



Targets By Language:

- **Basic:** Project, Class, Struct
- **C#:** Project, Class, Struct
- **C++:** Project, Class, Struct, Union

- **Java:** Project, File, Class, Interface
- **Pascal:** Project, Class, Interface
- **Python:** Project, Class
- **Web:** Project, PHP Class, PHP Interface

Configuration:

- For projects and architectures, this metric must also be enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show aggregated class metrics" option. Enabling this option can be time consuming.
- This metric can be disabled from "Project Configuration/Metrics/Object Orientated" with the "Show method and variable count metrics" option.

Internal Instance Variables

API ID: CountDeclInstanceVariableInternal
Languages: C#
Targets: Architectures, Classes, Project

Number of internal instance variables.

Targets By Language:

- **C#:** Project, Class, Struct

Configuration:

- For projects and architectures, this metric must also be enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show aggregated class metrics" option. Enabling this option can be time consuming.
- This metric can be disabled from "Project Configuration/Metrics/Object Orientated" with the "Show method and variable count metrics" option.

Internal Methods

API ID: CountDeclMethodInternal
Languages: C#
Targets: Architectures, Classes, Project

Number of local internal methods.

Targets By Language:

- **C#:** Project, Class, Struct

Configuration:

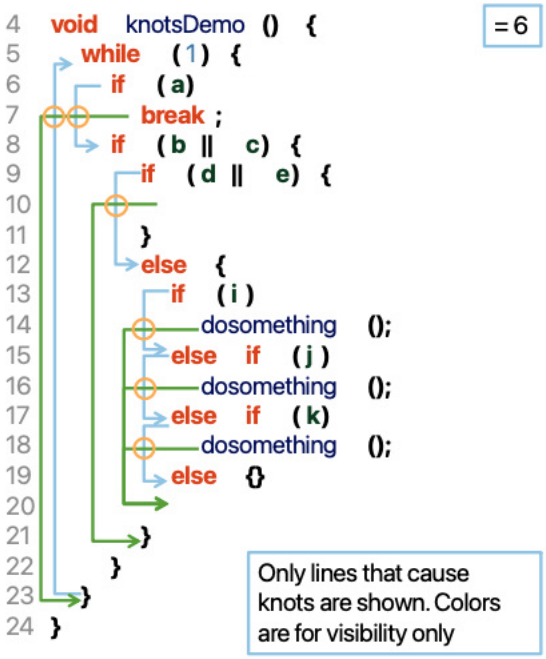
- For projects and architectures, this metric must also be enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show aggregated class metrics" option. Enabling this option can be time consuming.
- This metric can be disabled from "Project Configuration/Metrics/Object Orientated" with the "Show method and variable count metrics" option.

Knots

API ID: Knots
Languages: Ada, C#, C++, Java, Rust
Targets: Functions, Packages, Subprograms

Measure of overlapping jumps.

If a piece of code has arrowed lines indicating where every jump in the flow of control occurs, a knot is defined as where two such lines cross each other. The number of knots is proportional to the complexity of the control flow.



Targets By Language:

- **Ada:** Type, Entry, Function, Package, Procedure, Protected, Task
- **C#:** Method
- **C++:** Function

- **Java:** Method
- **Rust:** Function

Configuration:

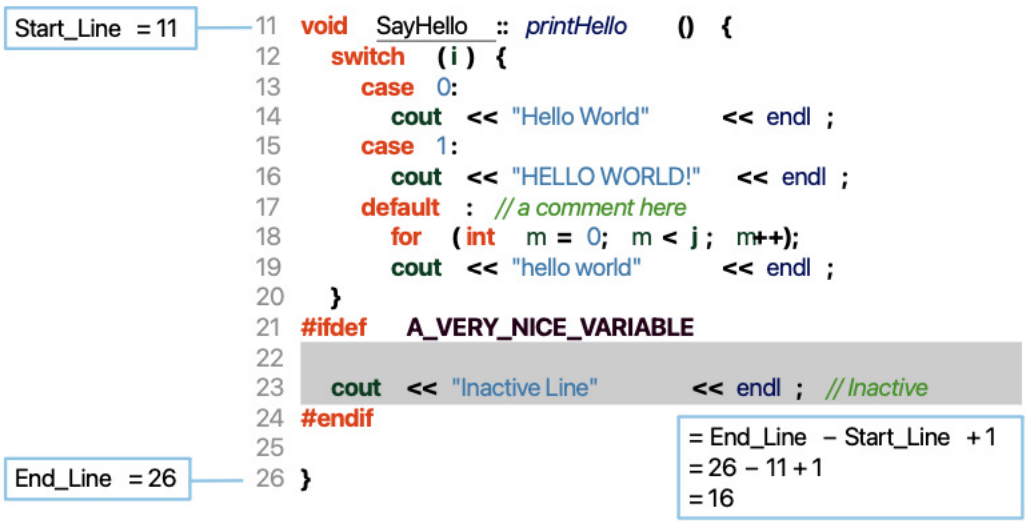
- This metric can be enabled from "Project Configuration/Metrics/Complexity" with the "Knots" option.

Lines

API ID: CountLine
Also Known As: Number of Lines (NL)
Languages: Ada, Assembly, Basic, C#, C++, Fortran, Java, Jovial, Pascal, Python, Rust, VHDL, Web
Targets: Architectures, Classes, Files, Functions, Modules, Packages, Project, Subprograms

Number of physical lines.

Also known as Number of Lines (NL).



Targets By Language:

- **Ada:** Project, File, Type, Entry, Function, Package, Procedure, Protected, Task
- **Basic:** Project, File, Method, Module, Class
- **C#:** Project, File, Class, Method

- **C++:** Project, File, Class, Struct, Union, Function
- **Fortran:** Project, File, Module, Block Data, Function, Program, Subroutine
- **Java:** Project, File, Class, Interface, Method
- **Jovial:** Project, File, Module, Subroutine
- **Pascal:** Project, File, Class, Interface, Compunit, Function, Procedure
- **Python:** Project, File, Class, Function
- **Rust:** Project, File, Function
- **VHDL:** Project, File, Procedure, Function, Process, Architecture
- **Web:** Project, File, PHP Class, PHP Interface

Configuration:

- This metric can be disabled from "Project Configuration/Metrics/Lines and Statements" with the "Show line count metrics" option.
- For projects and architectures, this metric is enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show default summary metrics" option.

Lines (HTML)

API ID: CountLineHtml
Languages: Web
Targets: Architectures, Files, Project

Number of all HTML lines.

Targets By Language:

- **Web:** Project, File

Configuration:

- This metric can be disabled from "Project Configuration/Metrics/Lines and Statements" with the "Show line count metrics" option.
- This metric can be enabled from "Project Configuration/Metrics/Lines and Statements" with the "Breakdown counts by web language" option.

Lines (JavaScript)

API ID: CountLineJavascript
Languages: Web
Targets: Architectures, Files, Project

Number of all JavaScript lines.

Targets By Language:

- **Web:** Project, File

Configuration:

- This metric can be disabled from "Project Configuration/Metrics/Lines and Statements" with the "Show line count metrics" option.
- This metric can be enabled from "Project Configuration/Metrics/Lines and Statements" with the "Breakdown counts by web language" option.

Lines (PHP)

API ID: CountLinePhp
Languages: Web
Targets: Architectures, Classes, Files, Project

Number of all PHP lines.

Targets By Language:

- **Web:** Project, File, PHP Class, PHP Interface

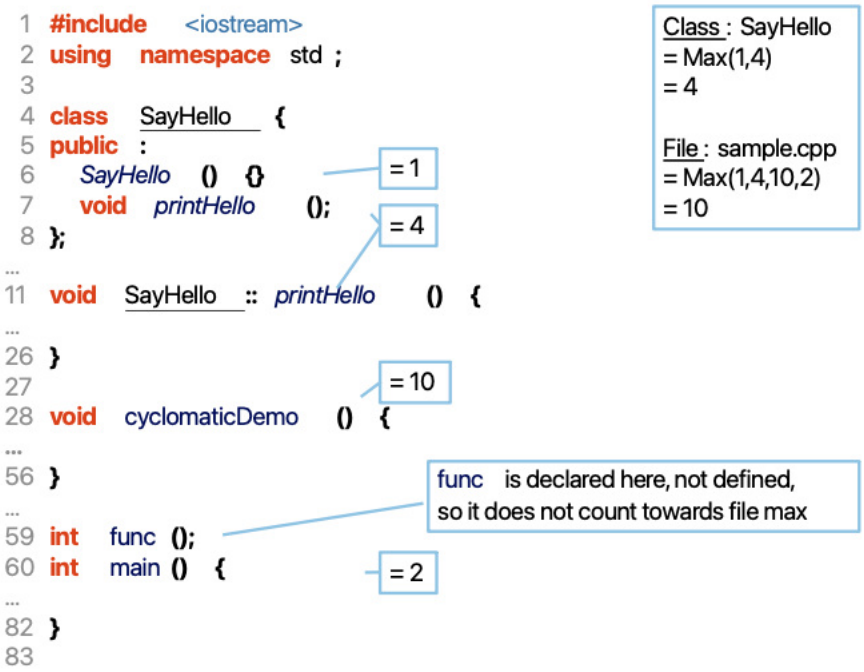
Configuration:

- This metric can be disabled from "Project Configuration/Metrics/Lines and Statements" with the "Show line count metrics" option.
- This metric can be enabled from "Project Configuration/Metrics/Lines and Statements" with the "Breakdown counts by web language" option.

Max Cyclomatic Complexity

API ID: MaxCyclomatic
Languages: Ada, Basic, C#, C++, Fortran, Java, Jovial, Pascal, Python, Rust, Web
Targets: Architectures, Classes, Files, Modules, Packages, Project

Maximum cyclomatic complexity of all nested functions or methods.



Targets By Language:

- **Ada:** Project, File, Package
- **Basic:** Project, File, Module, Class, Struct
- **C#:** Project, File, Class, Struct
- **C++:** Project, File, Class, Struct, Union
- **Fortran:** Project, File
- **Java:** Project, File, Class, Interface
- **Jovial:** Project, File
- **Pascal:** Project, File, Class, Interface, Compunit
- **Python:** Project, File, Class
- **Rust:** Project, File
- **Web:** Project, File, PHP Class, PHP Interface

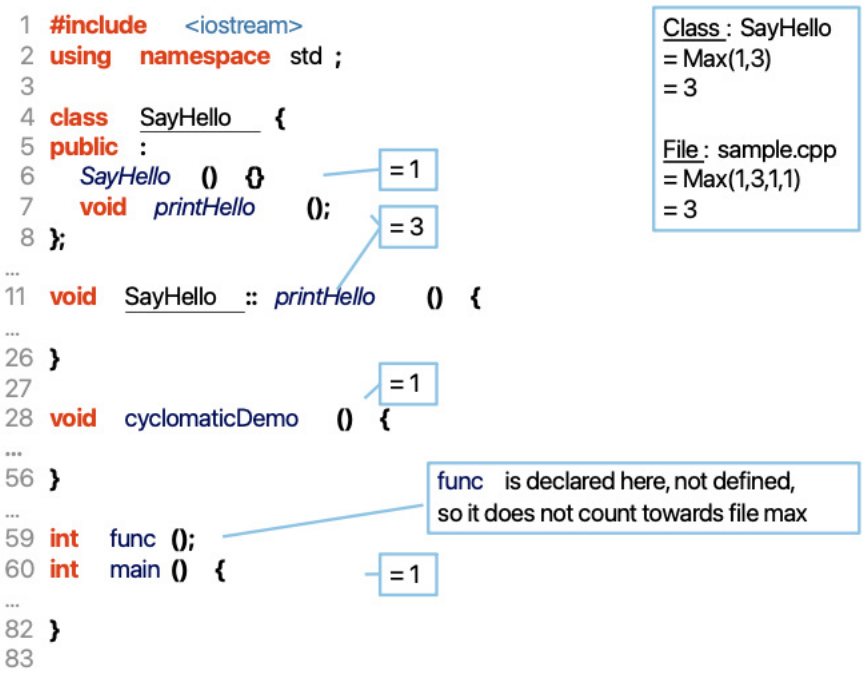
Configuration:

- For projects and architectures, this metric must also be enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show aggregated function metrics" option.
- The cyclomatic metric used is configured from "Project Configuration/Metrics/Complexity". This metric results from having "Count each decision in a multi-way-decision structure" on and having "Count logical conjuctions (OR) and logical and (AND)" off. Checking "Show all available cyclomatic complexity metric variants" will also enable the metric.

Max Essential Complexity

API ID: MaxEssential
Languages: Ada, Basic, C#, C++, Fortran, Java, Jovial, Pascal, Python, Rust, Web
Targets: Architectures, Classes, Files, Modules, Packages, Project

Maximum essential complexity of all nested functions or methods.



Targets By Language:

- **Ada:** Project, File, Package
- **Basic:** Project, File, Module, Class, Struct
- **C#:** Project, File, Class, Struct
- **C++:** Project, File, Class, Struct, Union
- **Fortran:** Project, File
- **Java:** Project, File, Class, Interface
- **Jovial:** Project, File
- **Pascal:** Project, File, Class, Interface, Compunit
- **Python:** Project, File, Class
- **Rust:** Project, File
- **Web:** Project, File, PHP Class, PHP Interface

Configuration:

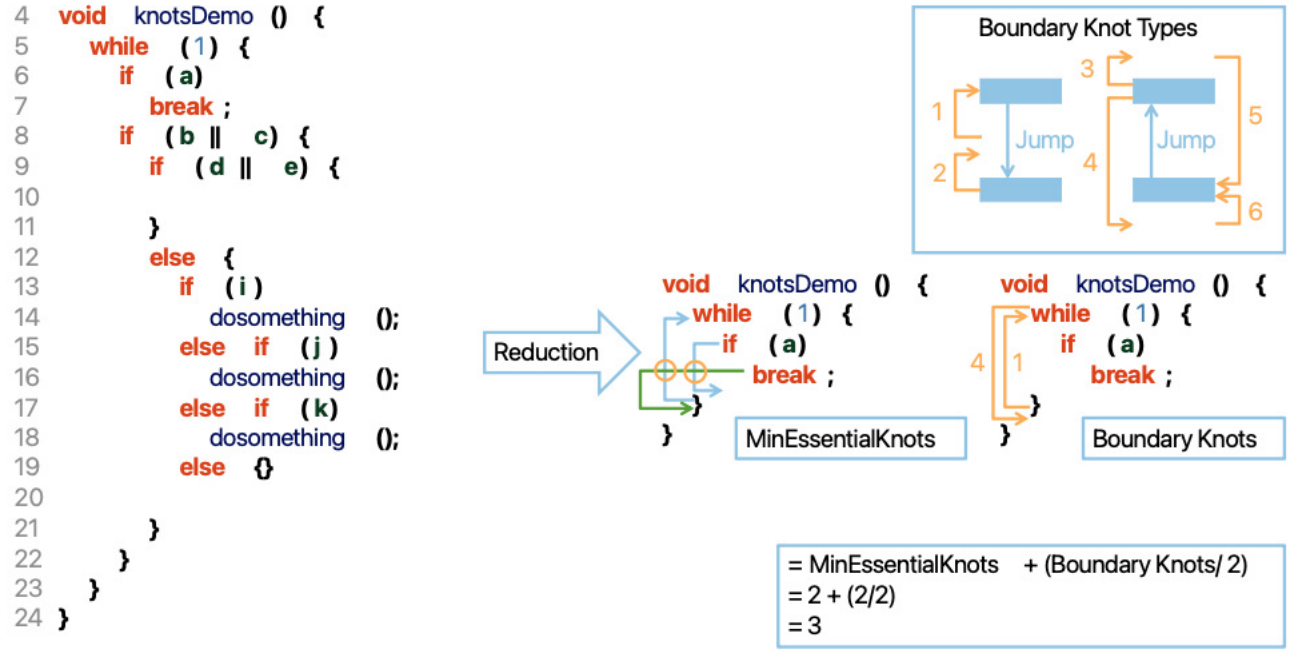
- For projects and architectures, this metric must also be enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show aggregated function metrics" option.
- This metric can be enabled from "Project Configuration/Metrics/Complexity"

with the "Essential" option.

Max Essential Knots

API ID: MaxEssentialKnots
Languages: Ada, C#, C++, Java, Rust
Targets: Functions, Packages, Subprograms

Maximum Knots after structured programming constructs have been removed.



Targets By Language:

- **Ada:** Type, Entry, Function, Package, Procedure, Protected, Task
- **C#:** Method
- **C++:** Function
- **Java:** Method
- **Rust:** Function

Configuration:

- This metric can be enabled from "Project Configuration/Metrics/Complexity" with the "Knots" option.

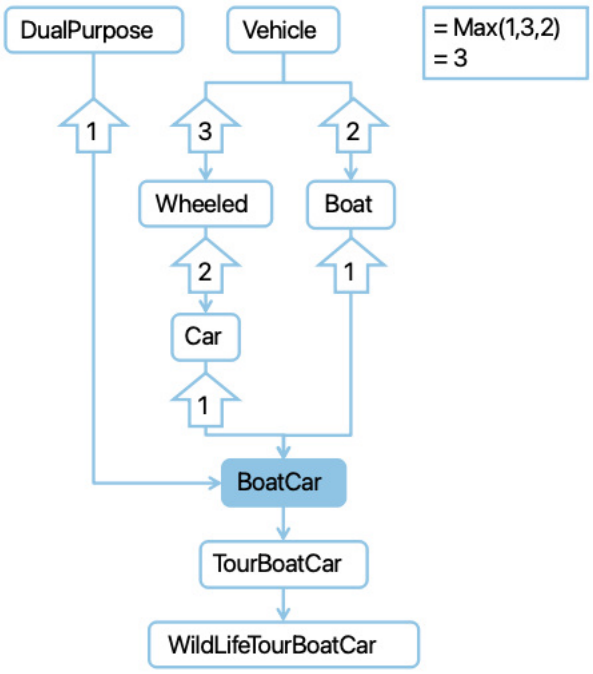
Max Inheritance Tree

API ID: MaxInheritanceTree
Also Known As: Depth of Inheritance Tree (DIT)
Languages: C#, C++, Java, Pascal, Python, Web
Targets: Classes

Maximum depth of class in inheritance tree.

The depth of a class within the inheritance hierarchy is the maximum number of nodes from the class node to the root of the inheritance tree. The root node has a DIT of 0. The deeper within the hierarchy, the more methods the class can inherit, increasing its complexity

Also known as Chidamber & Kemerer - Depth of Inheritance Tree (DIT).



Targets By Language:

- **C#:** Class, Struct
- **C++:** Class, Struct, Union
- **Java:** Class, Interface
- **Pascal:** Class, Interface
- **Python:** Class
- **Web:** PHP Class, PHP Interface

Configuration:

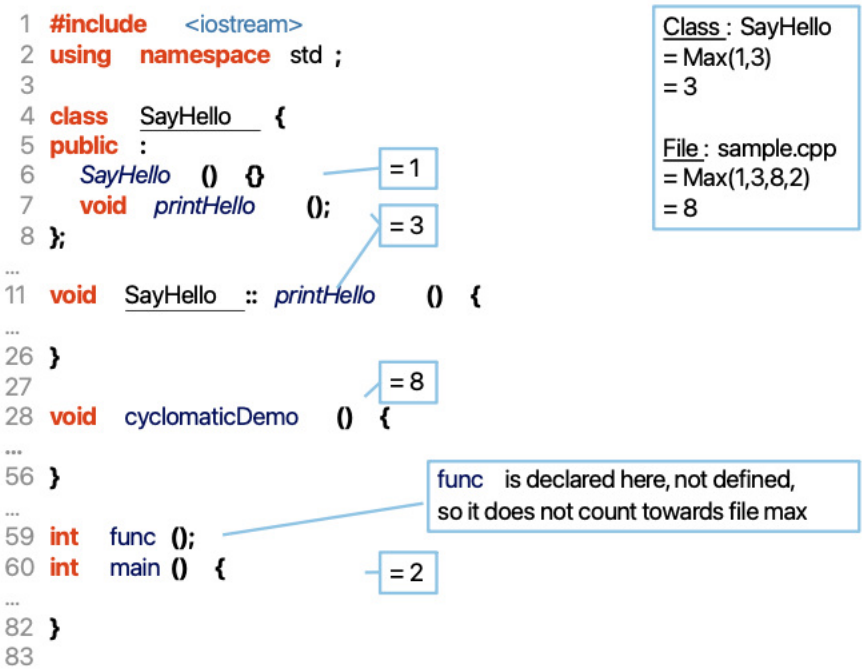
- This metric can be disabled from "Project Configuration/Metrics/Object

Orientated" with the "Show inheritance metrics" option.

Max Modified Cyclomatic Compexity

API ID: MaxCyclomaticModified
Languages: Ada, Basic, C#, C++, Fortran, Java, Jovial, Pascal, Python, Rust, Web
Targets: Architectures, Classes, Files, Modules, Packages, Project

Maximum modified cyclomatic complexity of nested functions or methods.



Targets By Language:

- **Ada:** Project, File, Package
- **Basic:** Project, File, Module, Class, Struct
- **C#:** Project, File, Class, Struct
- **C++:** Project, File, Class, Struct, Union
- **Fortran:** Project, File
- **Java:** Project, File, Class, Interface
- **Jovial:** Project, File
- **Pascal:** Project, File, Class, Interface, Compunit
- **Python:** Project, File, Class
- **Rust:** Project, File
- **Web:** Project, File, PHP Class, PHP Interface

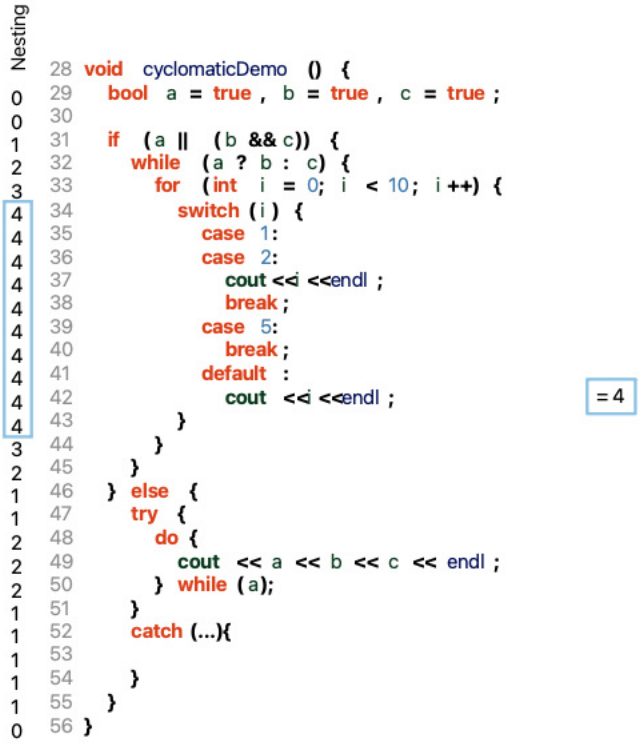
Configuration:

- For projects and architectures, this metric must also be enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show aggregated function metrics" option.
- The cyclomatic metric used is configured from "Project Configuration/Metrics/Complexity". This metric results from having "Count each decision in a multi-way-decision structure" off and having "Count logical conjuctions (OR) and logical and (AND)" off. Checking "Show all available cyclomatic complexity metric variants" will also enable the metric.

Max Nesting

API ID: MaxNesting
Languages: Ada, Basic, C#, C++, Fortran, Java, Jovial, Pascal, Python, Rust, Web
Targets: Architectures, Classes, Files, Functions, Modules, Packages, Project, Subprograms

Maximum nesting level of control constructs (if, while, for, switch, etc.) in the function.



Targets By Language:

- **Ada:** Project, File, Type, Entry, Function, Package, Procedure, Protected, Task
- **Basic:** Project, File, Method, Module, Class, Struct
- **C#:** Project, File, Class, Struct, Method
- **C++:** Project, File, Class, Struct, Union, Function

- **Fortran:** Project, File, Module, Block Data, Function, Program, Subroutine
- **Java:** Project, File, Class, Interface, Method
- **Jovial:** Project, File, Subroutine
- **Pascal:** Project, File, Class, Interface, Compunit, Function, Procedure
- **Python:** Project, File, Class, Function
- **Rust:** Project, File, Function
- **Web:** Project, File, PHP Class, PHP Interface

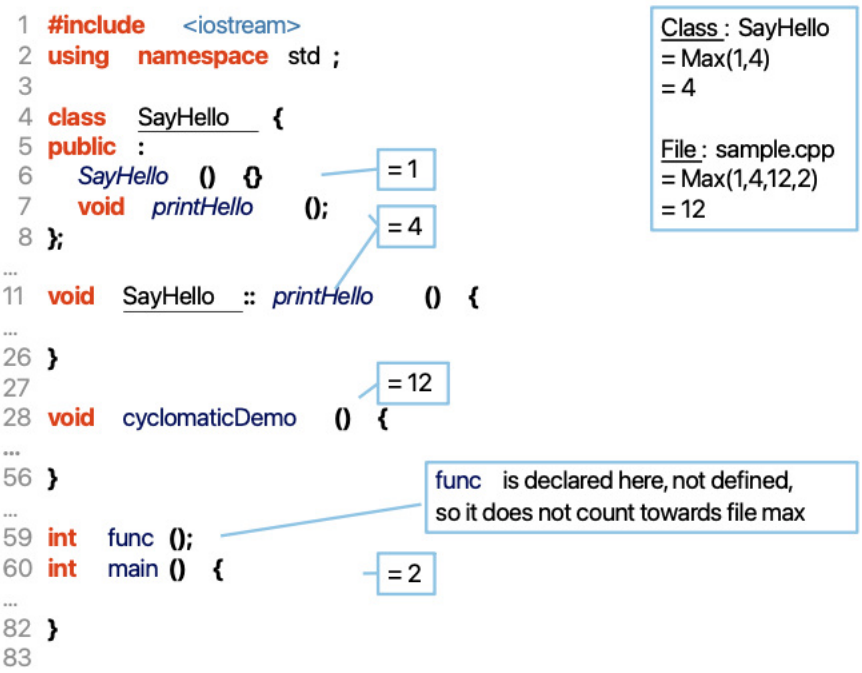
Configuration:

- For projects and architectures, this metric must also be enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show aggregated function metrics" option.
- This metric can be disabled from "Project Configuration/Metrics/Complexity" with the "Maximum Nesting" option.

Max Strict Cyclomatic Complexity

API ID: MaxCyclomaticStrict
Languages: Ada, C#, C++, Fortran, Java, Jovial, Pascal, Python, Rust, Web
Targets: Architectures, Classes, Files, Modules, Packages, Project

Maximum strict cyclomatic complexity of nested functions or methods.



Targets By Language:

- **Ada:** Project, File, Package
- **C#:** Project, File, Class, Struct
- **C++:** Project, File, Class, Struct, Union
- **Fortran:** Project, File
- **Java:** Project, File, Class, Interface
- **Jovial:** Project, File
- **Pascal:** Project, File, Class, Interface, Compunit
- **Python:** Project, File, Class
- **Rust:** Project, File
- **Web:** Project, File, PHP Class, PHP Interface

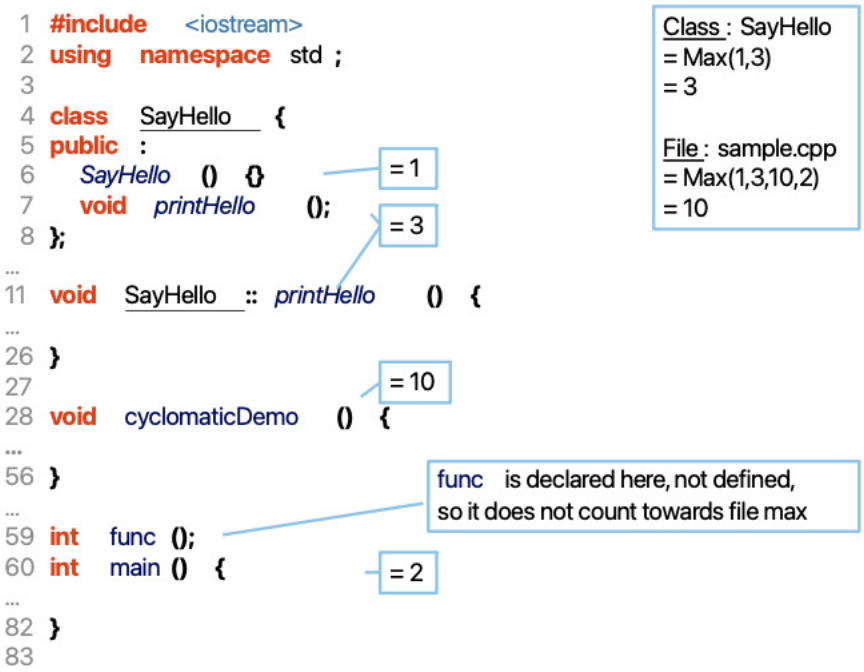
Configuration:

- For projects and architectures, this metric must also be enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show aggregated function metrics" option.
- The cyclomatic metric used is configured from "Project Configuration/Metrics/Complexity". This metric results from having "Count each decision in a multi-way-decision structure" on and having "Count logical conjuctions (OR) and logical and (AND)" on. Checking "Show all available cyclomatic complexity metric variants" will also enable the metric.

Max Strict Modified Cyclomatic Complexity

API ID: MaxCyclomaticStrictModified
Languages: Ada, Basic, C#, C++, Fortran, Java, Jovial, Pascal, Python, Rust, Web
Targets: Architectures, Classes, Files, Modules, Packages, Project

Maximum strict modified cyclomatic complexity of nested functions or methods.



Targets By Language:

- **Ada:** Project, File, Package
- **Basic:** Project, File, Module, Class, Struct
- **C#:** Project, File, Class, Struct
- **C++:** Project, File, Class, Struct, Union
- **Fortran:** Project, File
- **Java:** Project, File, Class, Interface
- **Jovial:** Project, File
- **Pascal:** Project, File, Class, Interface, Compunit
- **Python:** Project, File, Class
- **Rust:** Project, File
- **Web:** Project, File, PHP Class, PHP Interface

Configuration:

- For projects and architectures, this metric must also be enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show aggregated function metrics" option.
- The cyclomatic metric used is configured from "Project Configuration/Metrics/Complexity". This metric results from having "Count each decision in a multi-way-decision structure" off and having "Count logical conjuctions (OR) and logical and (AND)" on. Checking "Show all available cyclomatic complexity metric variants" will also enable the metric.

Max Strict Modified Essential Complexity

API ID: MaxEssentialStrictModified
Languages: Ada
Targets: Architectures, Files, Packages, Project

Maximum strict modified essential complexity of all nested functions or methods.

Targets By Language:

- **Ada:** Project, File, Package

Configuration:

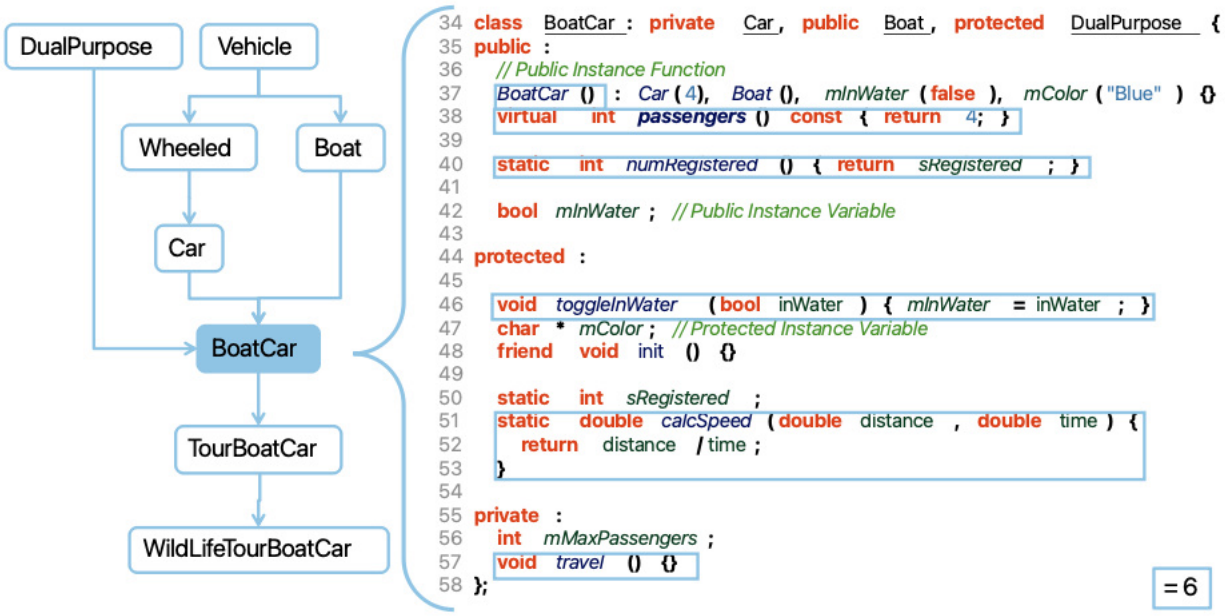
- For projects and architectures, this metric must also be enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show aggregated function metrics" option.
- This metric can be enabled from "Project Configuration/Metrics/Complexity" with the "Essential" option.

Methods

API ID: CountDeclMethod
Also Known As: Weighted Methods per Class (WMC)
Languages: Basic, C#, C++, Java, Pascal, Python, Web
Targets: Architectures, Classes, Files, Modules, Project

Number of local (not inherited) methods.

Also known as Chidamber & Kemerer - Weighted Methods per Class (WMC).



Targets By Language:

- **Basic:** Project, Module, Class, Struct
- **C#:** Project, Class, Struct
- **C++:** Project, Class, Struct, Union
- **Java:** Project, File, Class, Interface
- **Pascal:** Project, Class, Interface
- **Python:** Project, Class
- **Web:** Project, PHP Class, PHP Interface

Configuration:

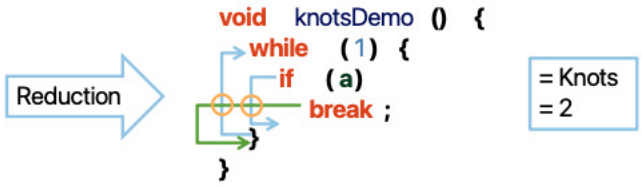
- For projects and architectures, this metric must also be enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show aggregated class metrics" option. Enabling this option can be time consuming.
- This metric can be disabled from "Project Configuration/Metrics/Object Orientated" with the "Show method and variable count metrics" option.

Min Essential Knots

API ID: MinEssentialKnots
Languages: Ada, C#, C++, Java, Rust
Targets: Functions, Packages, Subprograms

Minimum Knots after structured programming constructs have been removed.

```
4  void  knotsDemo ()  {
5    while  (1)  {
6      if  (a)
7        break ;
8      if  (b ||  c)  {
9        if  (d ||  e)  {
10
11       }
12       else  {
13         if  (i)
14           dosomething  ();
15         else  if  (j)
16           dosomething  ();
17         else  if  (k)
18           dosomething  ();
19         else  {}
20
21       }
22     }
23   }
24 }
```



Targets By Language:

- **Ada:** Type, Entry, Function, Package, Procedure, Protected, Task
- **C#:** Method
- **C++:** Function
- **Java:** Method
- **Rust:** Function

Configuration:

- This metric can be enabled from "Project Configuration/Metrics/Complexity" with the "Knots" option.

Modified Cyclomatic Complexity

API ID: CyclomaticModified
Also Known As: Modified Cyclomatic Complexity (CC3)
Languages: Ada, Assembly, Basic, C#, C++, Fortran, Java, Jovial, Pascal, Python, Rust, VHDL, Web
Targets: Files, Functions, Modules, Packages, Subprograms

Modified McCabe Cyclomatic complexity.

The Cyclomatic Complexity except that each decision in a multi-decision structure (switch in C/Java, Case in Ada, computed Goto and arithmetic if in FORTRAN)

statement is not counted and instead the entire multi-way decision structure counts as 1.

Also known as McCabe - Modified Cyclomatic Complexity (CC3).

Targets By Language:

- **Ada:** Type, Entry, Function, Package, Procedure, Protected, Task
- **Basic:** Method
- **C#:** Method
- **C++:** Function
- **Fortran:** Module, Block Data, Function, Program, Subroutine
- **Java:** Method
- **Jovial:** Subroutine
- **Pascal:** Compunit, Function, Procedure
- **Python:** File, Function
- **Rust:** Function
- **VHDL:** Procedure, Function, Process
- **Web:** File

Configuration:

- The cyclomatic metric used is configured from "Project Configuration/ Metrics/Complexity". This metric results from having "Count each decision in a multi-way-decision structure" off and having "Count logical conjuctions (OR) and logical and (AND)" off. Checking "Show all available cyclomatic complexity metric variants" will also enable the metric.

Modules

API ID: CountDeclModule
Languages: Fortran, Jovial, Pascal
Targets: Architectures, Files, Functions, Modules, Project, Subprograms

Number of modules.

Targets By Language:

- **Fortran:** Project, File, Module, Block Data, Function, Program, Subroutine
- **Jovial:** Project, File
- **Pascal:** Project, File

Configuration:

- For projects and architectures, this metric is enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show aggregated file metrics" option.

New Lines

API ID: CountLineNew
Languages: Ada, Assembly, Basic, C#, C++, Fortran, Java, Jovial, Pascal, Python, Rust, VHDL, Web
Targets: Classes, Files, Functions, Modules, Packages, Subprograms

Number of new lines

Outputs

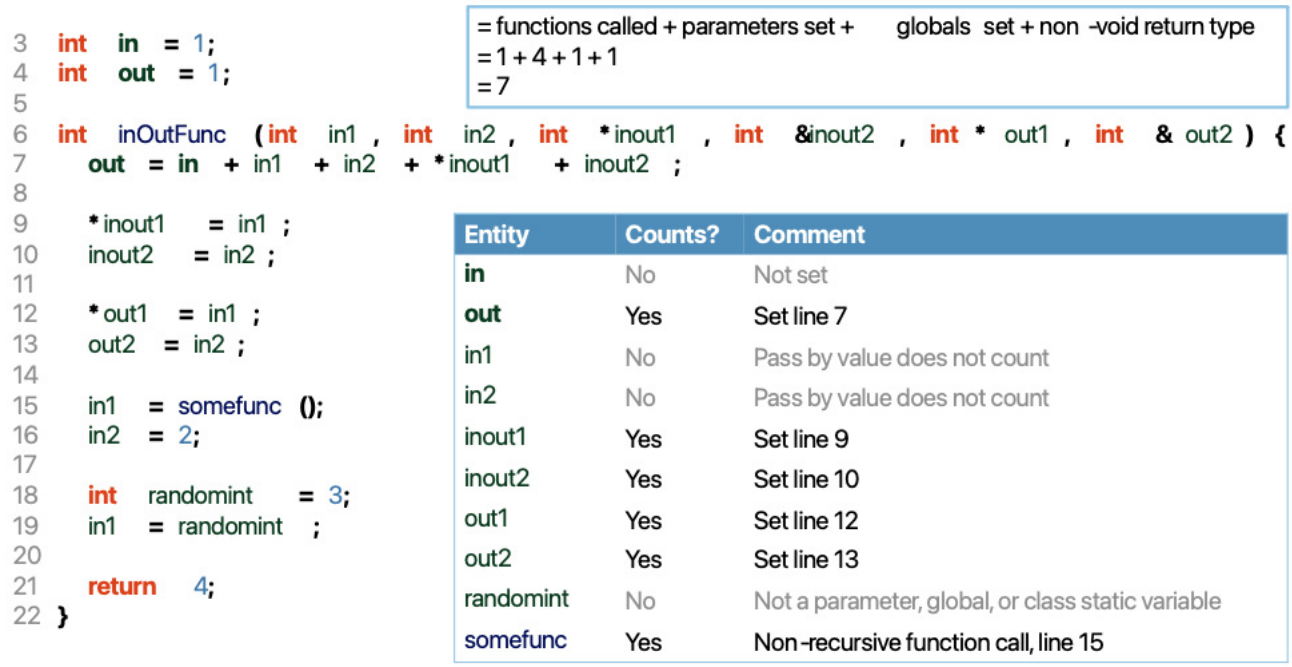
API ID: CountOutput
Also Known As: FANOUT (Infomational fan-out)
Languages: C#, C++, Fortran, Java
Targets: Functions, Subprograms

Number of called subprograms plus global variables set.

The number of outputs that are SET. These can be parameters or global variables. So Functions calls + Parameters set/modify + Global Variables set/modify. A non-void return value adds one to the count. Recursive function calls, local variables that are not class static variables, and parameters that are pass by value are not included. Of the two general approachs to calculating FANOUT (informational versus

structural) ours is the informational approach.

Also known as FANOUT (Infomational fan-out).



```
3  int  in  = 1;
4  int  out  = 1;
5
6  int  inOutFunc  (int  in1 , int  in2 , int  *inout1  , int  &inout2  , int * out1 , int  & out2 ) {
7     out  = in  + in1  + in2  + *inout1  + inout2  ;
8
9     *inout1  = in1 ;
10    inout2  = in2 ;
11
12    *out1  = in1 ;
13    out2  = in2 ;
14
15    in1  = somefunc  ();
16    in2  = 2;
17
18    int  randomint  = 3;
19    in1  = randomint  ;
20
21    return  4;
22 }
```

= functions called + parameters set + globals set + non -void return type
= 1 + 4 + 1 + 1
= 7

Entity	Counts?	Comment
in	No	Not set
out	Yes	Set line 7
in1	No	Pass by value does not count
in2	No	Pass by value does not count
inout1	Yes	Set line 9
inout2	Yes	Set line 10
out1	Yes	Set line 12
out2	Yes	Set line 13
randomint	No	Not a parameter, global, or class static variable
somefunc	Yes	Non-recursive function call, line 15

Targets By Language:

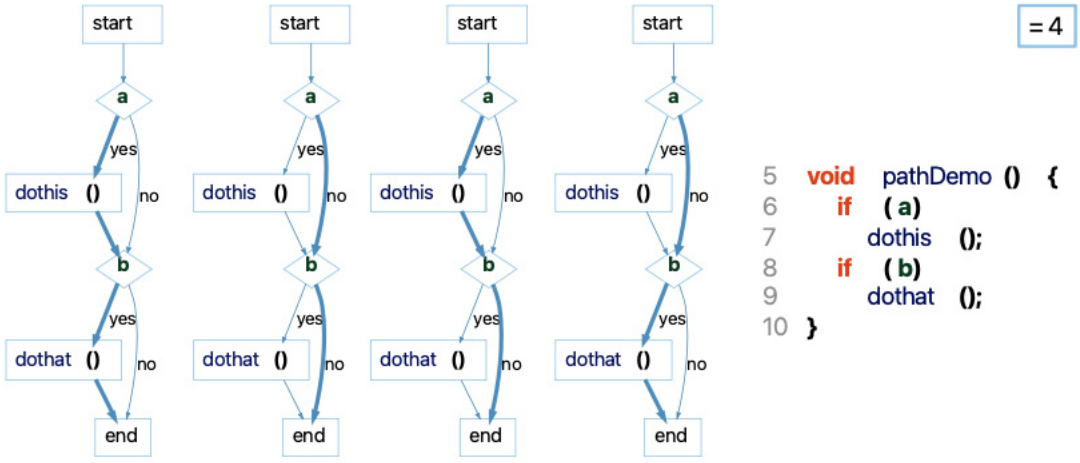
- **C#:** Method
- **C++:** Function
- **Fortran:** Function, Program, Subroutine
- **Java:** Method

Paths

API ID: CountPath
Also Known As: NPATH
Languages: Ada, Basic, C#, C++, Fortran, Java, Jovial, Pascal, Python, Rust, Web
Targets: Files, Functions, Modules, Packages, Subprograms

Number of unique paths through a body of code, not counting abnormal exits or gotos.

Also known as NPATH.



Targets By Language:

- **Ada:** Type, Entry, Function, Package, Procedure, Protected, Task
- **Basic:** Method
- **C#:** Method
- **C++:** Function
- **Fortran:** Module, Block Data, Function, Program, Subroutine
- **Java:** Method
- **Jovial:** Subroutine
- **Pascal:** Compunit, Function, Procedure
- **Python:** File, Function
- **Rust:** File, Function
- **Web:** File

Configuration:

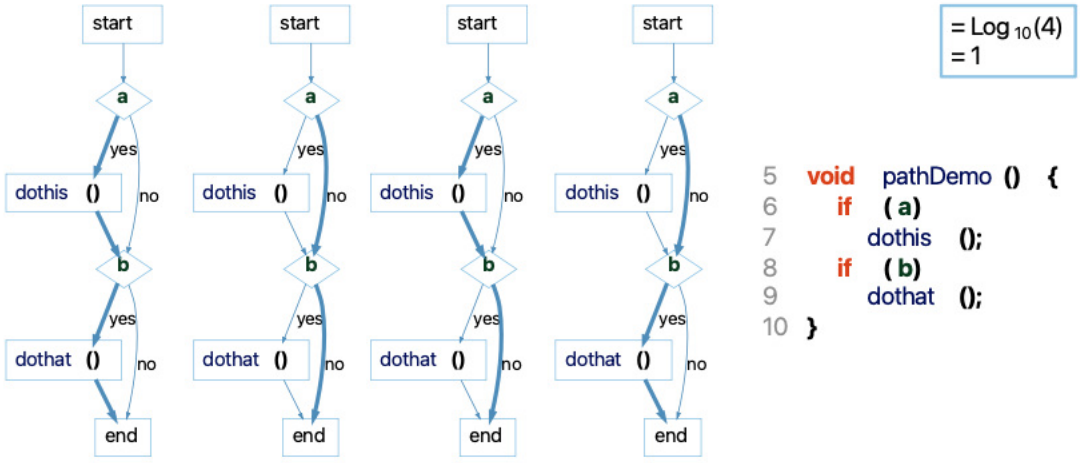
- This metric can be enabled from "Project Configuration/Metrics/Complexity" with the "Paths" option.

Paths Log(x)

API ID: CountPathLog
Languages: Ada, Basic, C#, C++, Fortran, Java, Jovial, Pascal, Python, Rust, Web
Targets: Files, Functions, Modules, Packages, Subprograms

The base 10 logarithm Log(x) of the number of unique paths though a body of code,

not counting abnormal exits or gotos, truncated to an integer value.



Targets By Language:

- **Ada:** Type, Entry, Function, Package, Procedure, Protected, Task
- **Basic:** Method
- **C#:** Method
- **C++:** Function
- **Fortran:** Module, Block Data, Function, Program, Subroutine
- **Java:** Method
- **Jovial:** Subroutine
- **Pascal:** Compunit, Function, Procedure
- **Python:** File, Function
- **Rust:** File, Function
- **Web:** File

Configuration:

- This metric can be enabled from "Project Configuration/Metrics/Complexity" with the "Paths" option.

Percent Changed

API ID: PercentChanged
Languages: Ada, Assembly, Basic, C#, C++, Fortran, Java, Jovial, Pascal, Python, Rust, VHDL, Web

Targets: Classes, Files, Functions, Modules, Packages, Subprograms

(Added + Removed + 2*Changed) / (CurrentLines + PreviousLines)

Percent Lack Of Cohesion

API ID: PercentLackOfCohesion
Also Known As: LCOM2, Lack of Cohesion in Methods (LCOM/LOCM)
Languages: Ada, Basic, C#, C++, Java, Pascal
Targets: Classes, Packages

100% minus the average cohesion for package entities.

100% minus average cohesion for class data members. Calculates what percentage of class methods use a given class instance variable. To calculate, average percentages for all of that class's instance variables and subtract from 100%. A lower percentage means higher cohesion between class data and methods.

Also known as Chidamber & Kemerer - Lack of Cohesion in Methods (LCOM/LOCM); LCOM2.

```
1  class CohesionClass {
2  public :
3    void func1 () {
4      for (int i = 0; i < mVar1; i ++) {
5        mVar2 = nullptr ;
6      }
7      mVar1 = 3;
8    }
9
10   void func2 () {
11     mVar1 = 4;
12   }
13
14   static void addObj () {
15     sNumObjs ++;
16   }
17 protected :
18
19   void func3 () {
20     mVar2 = "blue" ;
21   }
22 private :
23
24   void func4 () {
25
26   }
27
28   int mVar1 ;
29   char * mVar2;
30   static int sNumObjs;
31 };
```

	mVar1	mVar2	sNumObjs
func1 ()	✓	✓	
func2 ()	✓		
addObj ()			✓
func3 ()		✓	
func4 ()			

# Functions Using Variable:	2	2	1
Divided By Total Functions (5):	0.4	0.4	0.2
Averaged Together:	0.3333		
Subtract from 1:	0.6667		
To Percent:	67%		

Targets By Language:

- **Ada:** Package
- **Basic:** Class, Struct

- **C#:** Class, Struct
- **C++:** Class, Struct, Union
- **Java:** Class, Interface
- **Pascal:** Class, Interface

Configuration:

- This metric can be disabled from "Project Configuration/Metrics/Object Orientated" with the "Show coupling and cohesion metrics" option.

Percent Lack Of Cohesion Modified

API ID: PercentLackOfCohesionModified
Languages: Basic, C#, Java, Pascal
Targets: Classes

100% minus the average cohesion for class data members, modified for accessor methods.

Same as PercentLackOfCohesion but does not penalize the use of accessor methods within a class to set/read variables.

Targets By Language:

- **Basic:** Class, Struct
- **C#:** Class, Struct
- **Java:** Class, Interface
- **Pascal:** Class, Interface

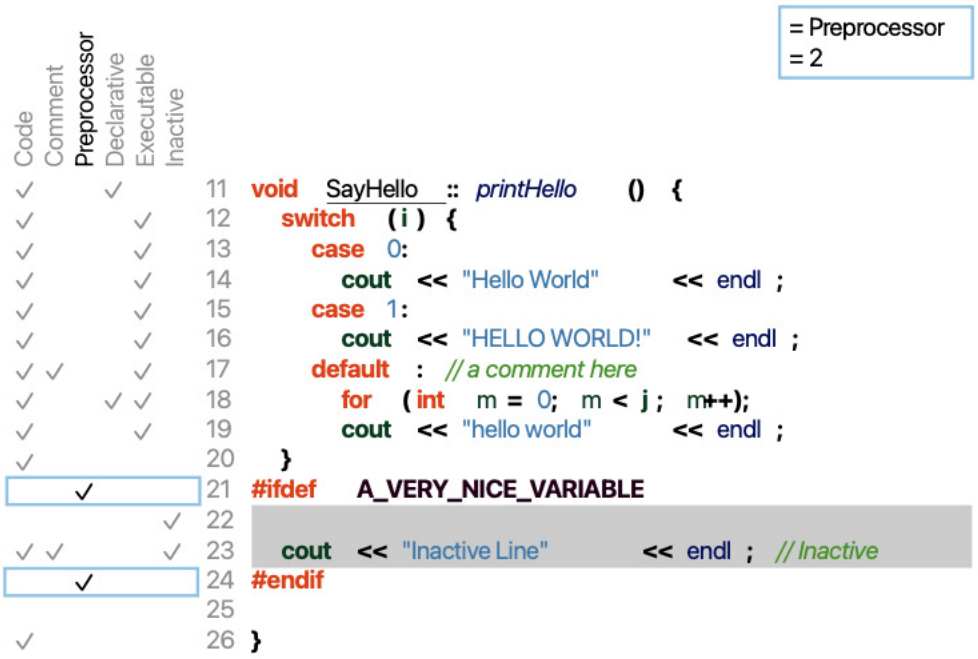
Configuration:

- This metric can be disabled from "Project Configuration/Metrics/Object Orientated" with the "Show coupling and cohesion metrics" option.

Preprocessor Lines

API ID: CountLinePreprocessor
Languages: Ada, C#, C++
Targets: Architectures, Classes, Files, Functions, Packages, Project, Subprograms

Number of preprocessor lines.



Targets By Language:

- **Ada:** Project, File, Type, Entry, Function, Package, Procedure, Protected, Task
- **C#:** Project, File, Class, Method
- **C++:** Project, File, Class, Struct, Union, Function

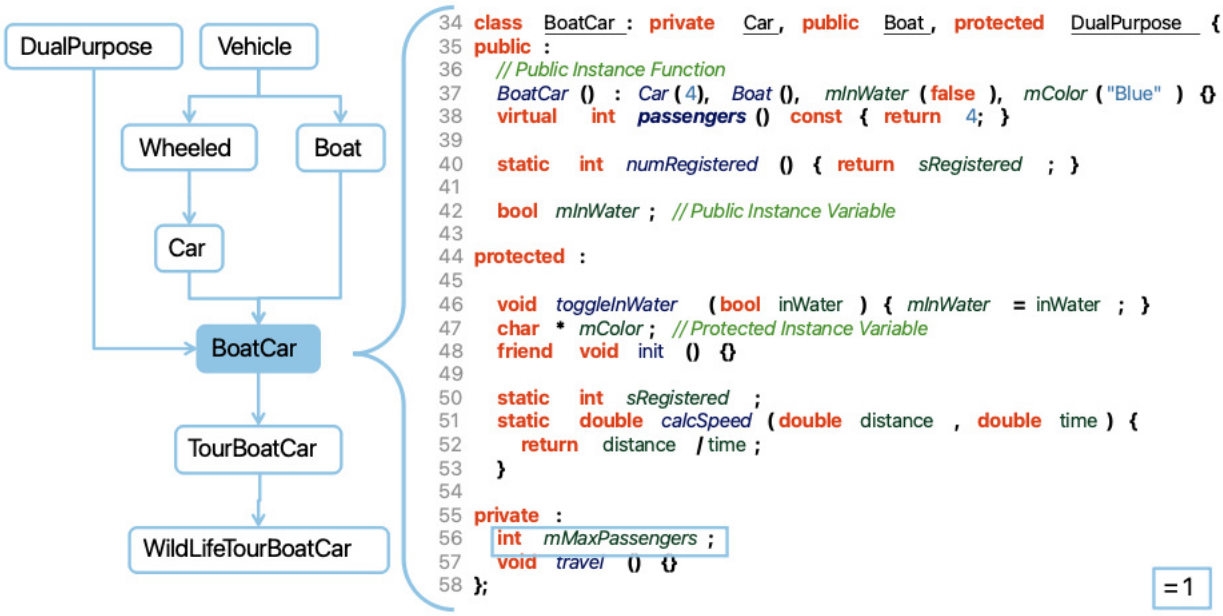
Configuration:

- This metric can be disabled from "Project Configuration/Metrics/Lines and Statements" with the "Show line count metrics" option.
- For projects and architectures, this metric is enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show default summary metrics" option.

Private Instance Variables

API ID: CountDeclInstanceVariablePrivate
Languages: C#, C++, Web
Targets: Architectures, Classes, Project

Number of private instance variables.



Targets By Language:

- **C#:** Project, Class, Struct
- **C++:** Project, Class, Struct, Union
- **Web:** Project, PHP Class, PHP Interface

Configuration:

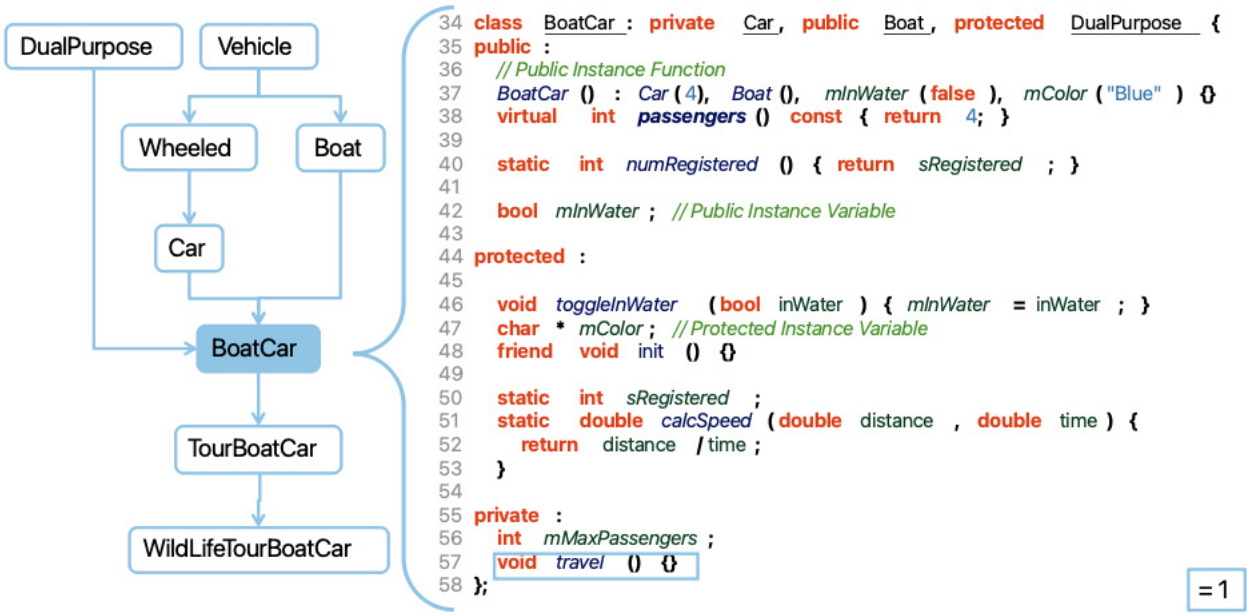
- For projects and architectures, this metric must also be enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show aggregated class metrics" option. Enabling this option can be time consuming.
- This metric can be disabled from "Project Configuration/Metrics/Object Orientated" with the "Show method and variable count metrics" option.

Private Methods

API ID: CountDeclMethodPrivate
Also Known As: Number Private Methods (NPRM)
Languages: Basic, C#, C++, Java, Pascal, Web
Targets: Architectures, Classes, Files, Modules, Project

Number of local (not inherited) private methods.

Also known as Number Private Methods (NPRM).



Targets By Language:

- **Basic:** Project, Module, Class, Struct
- **C#:** Project, Class, Struct
- **C++:** Project, Class, Struct, Union
- **Java:** Project, File, Class, Interface
- **Pascal:** Project, Class, Interface
- **Web:** Project, PHP Class, PHP Interface

Configuration:

- For projects and architectures, this metric must also be enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show aggregated class metrics" option. Enabling this option can be time consuming.
- This metric can be disabled from "Project Configuration/Metrics/Object Orientated" with the "Show method and variable count metrics" option.

Program Units

API ID: CountDeclProgUnit
Languages: Fortran
Targets: Architectures, Files, Project

Number of non-nested modules, block data units, and subprograms.

Targets By Language:

- **Fortran:** Project, File

Configuration:

- For projects and architectures, this metric is enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show aggregated file metrics" option.

Properties

API ID: CountDeclProperty
Languages: C#, Pascal
Targets: Architectures, Classes, Project

Number of properties.

Targets By Language:

- **C#:** Project, Class, Struct
- **Pascal:** Project, Class, Interface

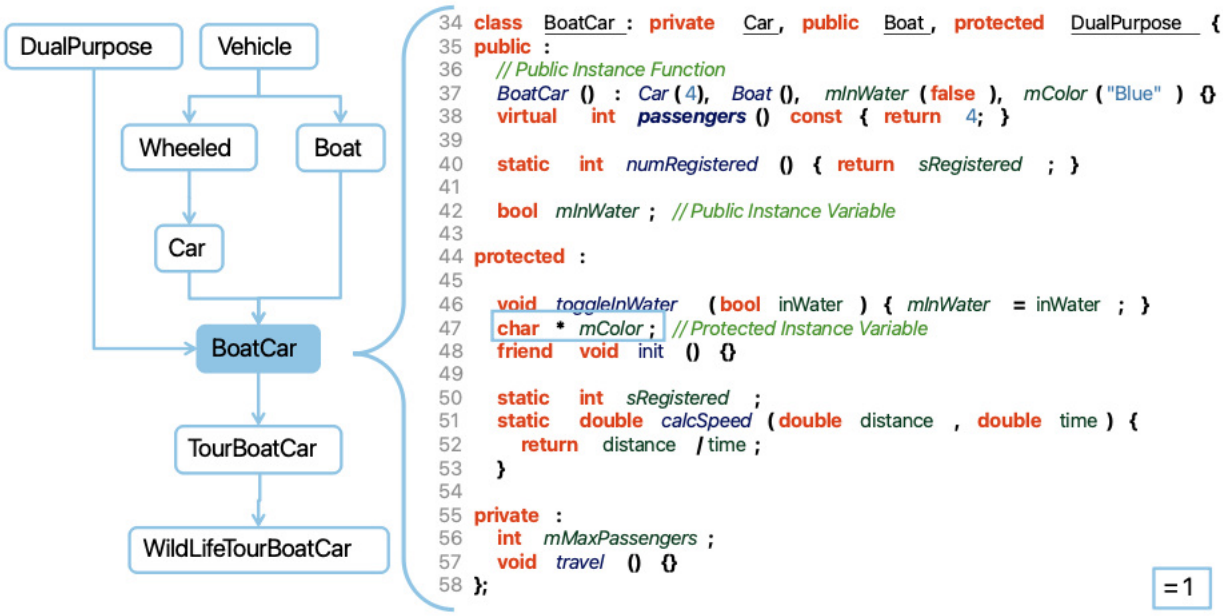
Configuration:

- For projects and architectures, this metric must also be enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show aggregated class metrics" option. Enabling this option can be time consuming.
- This metric can be disabled from "Project Configuration/Metrics/Object Orientated" with the "Show method and variable count metrics" option.

Protected Instance Variables

API ID: CountDeclInstanceVariableProtected
Languages: C#, C++, Web
Targets: Architectures, Classes, Project

Number of protected instance variables.



Targets By Language:

- **C#:** Project, Class, Struct
- **C++:** Project, Class, Struct, Union
- **Web:** Project, PHP Class, PHP Interface

Configuration:

- For projects and architectures, this metric must also be enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show aggregated class metrics" option. Enabling this option can be time consuming.
- This metric can be disabled from "Project Configuration/Metrics/Object Orientated" with the "Show method and variable count metrics" option.

Protected Internal Instance Variables

API ID: CountDeclInstanceVariableProtectedInternal
Languages: C#
Targets: Architectures, Classes, Project

Number of protected internal instance variables.

Targets By Language:

- **C#:** Project, Class, Struct

Configuration:

- For projects and architectures, this metric must also be enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show aggregated class metrics" option. Enabling this option can be time consuming.
- This metric can be disabled from "Project Configuration/Metrics/Object Orientated" with the "Show method and variable count metrics" option.

Protected Internal Methods

API ID: CountDeclMethodProtectedInternal
Languages: C#
Targets: Architectures, Classes, Project

Number of local protected internal methods.

Targets By Language:

- **C#:** Project, Class, Struct

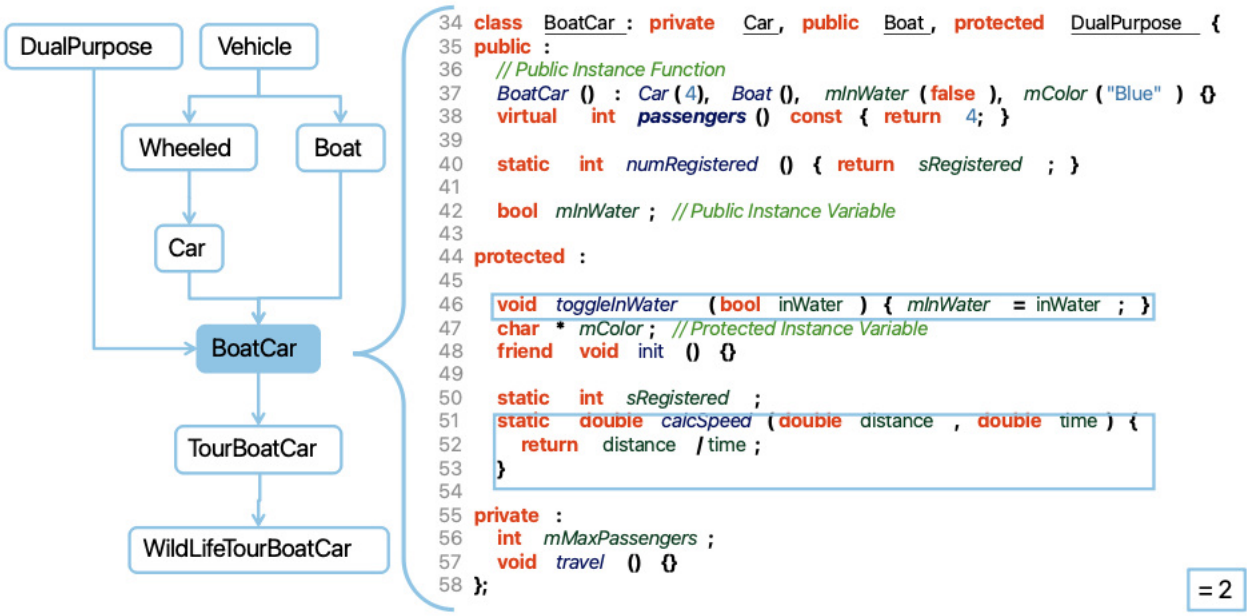
Configuration:

- For projects and architectures, this metric must also be enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show aggregated class metrics" option. Enabling this option can be time consuming.
- This metric can be disabled from "Project Configuration/Metrics/Object Orientated" with the "Show method and variable count metrics" option.

Protected Methods

API ID: CountDeclMethodProtected
Languages: Basic, C#, C++, Java, Pascal, Web
Targets: Architectures, Classes, Files, Modules, Project

Number of local protected methods.



Targets By Language:

- **Basic:** Project, Module, Class, Struct
- **C#:** Project, Class, Struct
- **C++:** Project, Class, Struct, Union
- **Java:** Project, File, Class, Interface
- **Pascal:** Project, Class, Interface
- **Web:** Project, PHP Class, PHP Interface

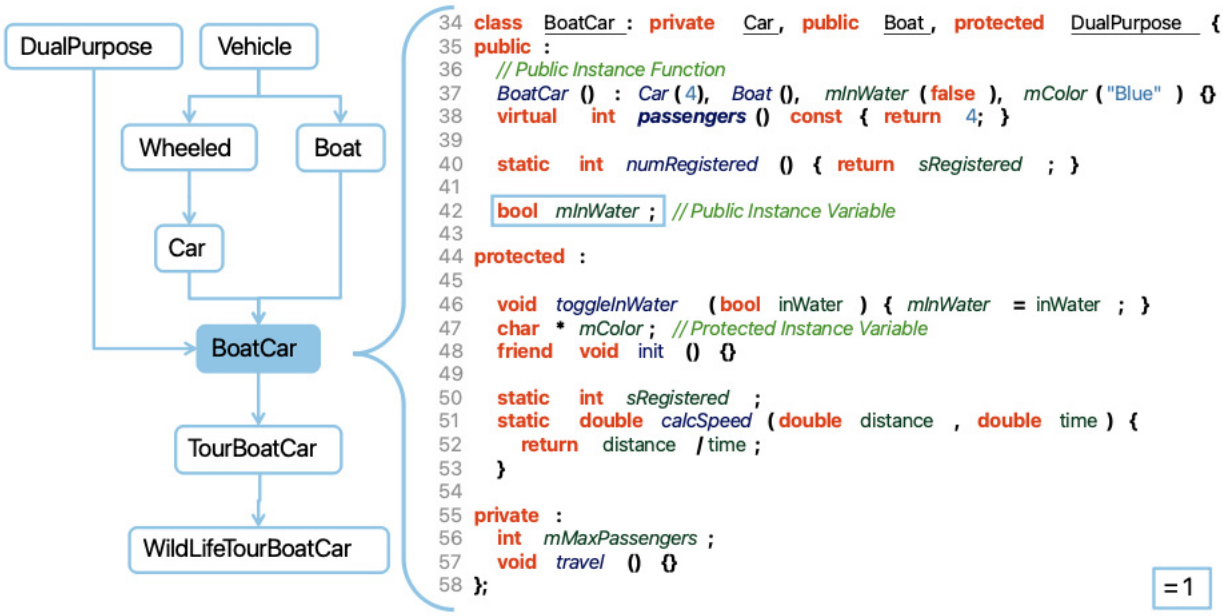
Configuration:

- For projects and architectures, this metric must also be enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show aggregated class metrics" option. Enabling this option can be time consuming.
- This metric can be disabled from "Project Configuration/Metrics/Object Orientated" with the "Show method and variable count metrics" option.

Public Instance Variables

API ID: CountDeclInstanceVariablePublic
Languages: C#, C++, Web
Targets: Architectures, Classes, Project

Number of public instance variables.



Targets By Language:

- **C#:** Project, Class, Struct
- **C++:** Project, Class, Struct, Union
- **Web:** Project, PHP Class, PHP Interface

Configuration:

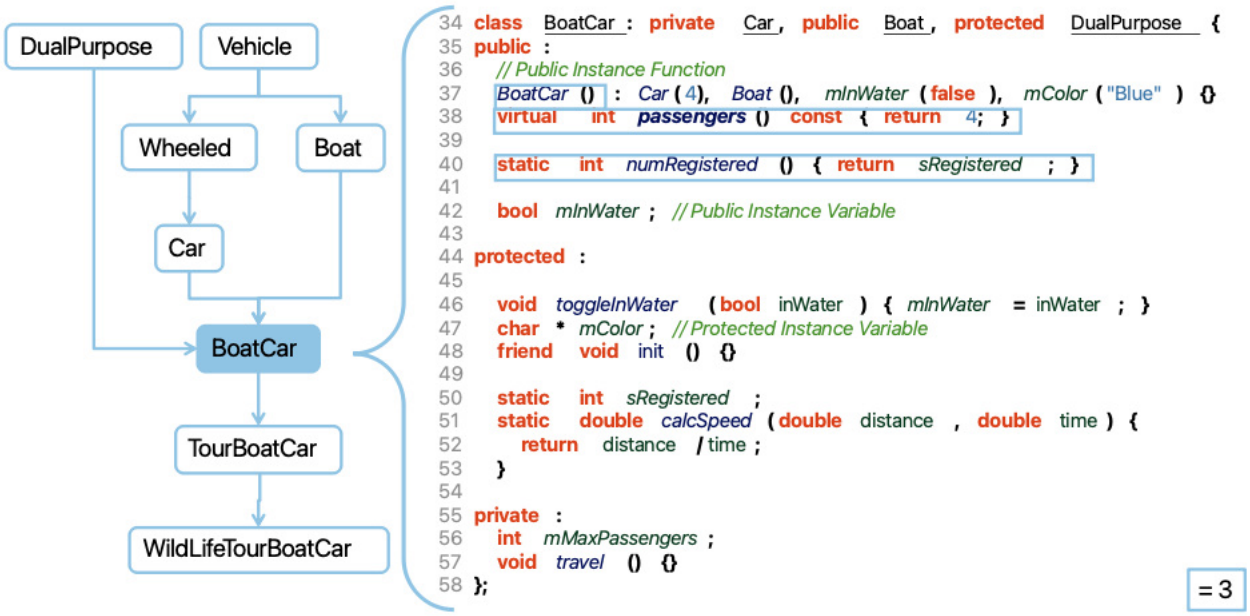
- For projects and architectures, this metric must also be enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show aggregated class metrics" option. Enabling this option can be time consuming.
- This metric can be disabled from "Project Configuration/Metrics/Object Orientated" with the "Show method and variable count metrics" option.

Public Methods

API ID: CountDeclMethodPublic
Also Known As: Number of Public Methods (PM, NPM)
Languages: Basic, C#, C++, Java, Pascal, Web
Targets: Architectures, Classes, Files, Modules, Project

Number of local (not inherited) public methods.

Also known as Lorenz & Kidd - Number of Public Methods (PM, NPM).



Targets By Language:

- **Basic:** Project, Module, Class, Struct
- **C#:** Project, Class, Struct
- **C++:** Project, Class, Struct, Union
- **Java:** Project, File, Class, Interface
- **Pascal:** Project, Class, Interface
- **Web:** Project, PHP Class, PHP Interface

Configuration:

- For projects and architectures, this metric must also be enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show aggregated class metrics" option. Enabling this option can be time consuming.
- This metric can be disabled from "Project Configuration/Metrics/Object Orientated" with the "Show method and variable count metrics" option.

Removed Lines

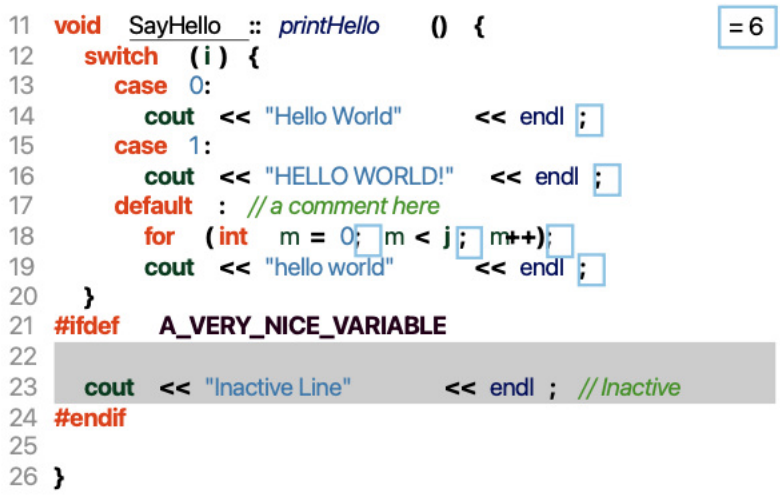
API ID: CountLineRemoved
Languages: Ada, Assembly, Basic, C#, C++, Fortran, Java, Jovial, Pascal, Python, Rust, VHDL, Web
Targets: Classes, Files, Functions, Modules, Packages, Subprograms

Number of removed lines

Semicolons

API ID: CountSemicolon
Languages: Ada, C#, C++, Java
Targets: Architectures, Classes, Files, Functions, Packages, Project, Subprograms

Number of semicolons.



Targets By Language:

- **Ada:** Project, File, Type, Entry, Function, Package, Procedure, Protected, Task
- **C#:** Project, File, Class, Method
- **C++:** Project, File, Function
- **Java:** Project, File, Class, Interface, Method

Configuration:

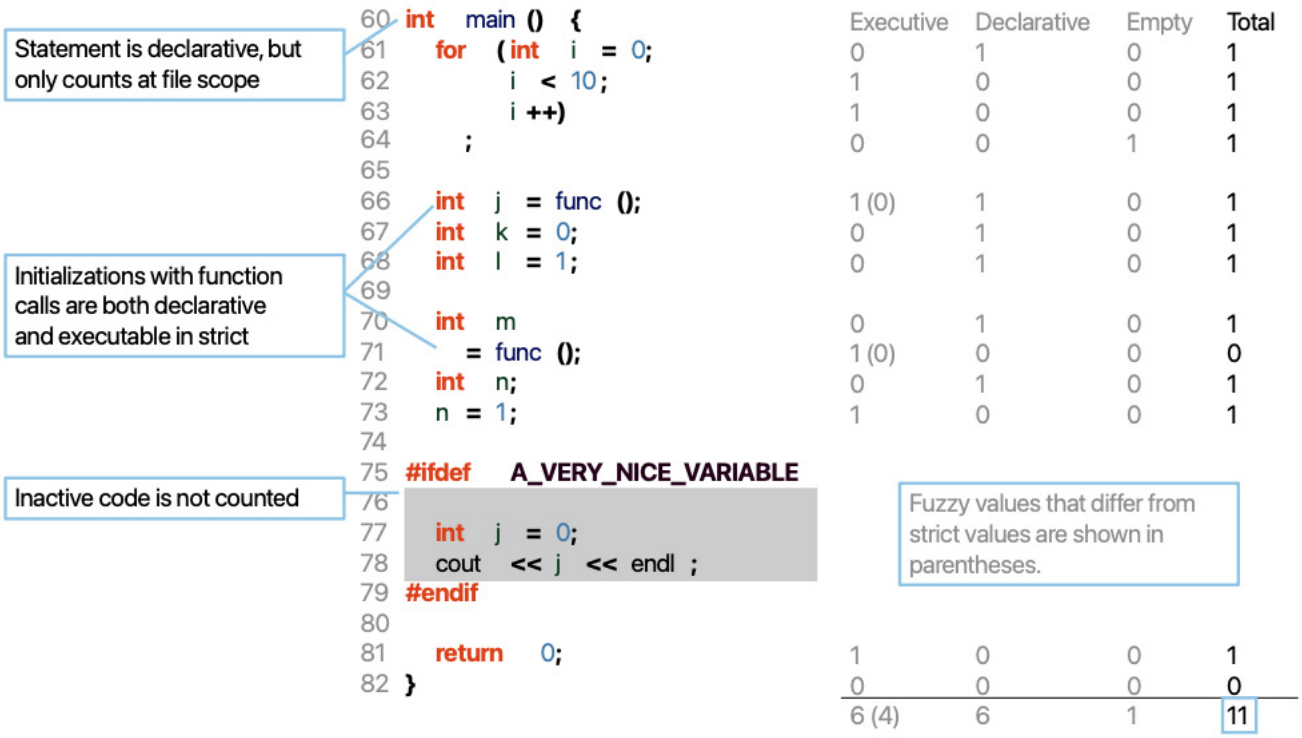
- This metric can be disabled from "Project Configuration/Metrics/Lines and Statements" with the "Show statement count metrics" option.

Statements

API ID: CountStmt
Languages: Ada, Assembly, Basic, C#, C++, Fortran, Java, Jovial, Pascal, Python, Rust, VHDL, Web

Targets: Architectures, Classes, Files, Functions, Modules, Packages, Project, Subprograms

Number of statements.



Targets By Language:

- **Ada:** Project, File, Type, Entry, Function, Package, Procedure, Protected, Task
- **Basic:** Project, File, Method, Module, Class
- **C#:** Project, File, Class, Method
- **C++:** Project, File, Class, Struct, Union, Function
- **Fortran:** Project, File, Block Data, Function, Program, Subroutine
- **Java:** Project, File, Class, Interface, Method
- **Jovial:** Project, File, Module, Subroutine
- **Pascal:** Project, File, Class, Interface, Compunit, Function, Procedure
- **Python:** Project, File, Class, Function
- **Rust:** Project, File, Function
- **VHDL:** Project, File
- **Web:** Project, File, PHP Class, PHP Interface

Configuration:

- This metric can be disabled from "Project Configuration/Metrics/Lines and Statements" with the "Show statement count metrics" option.

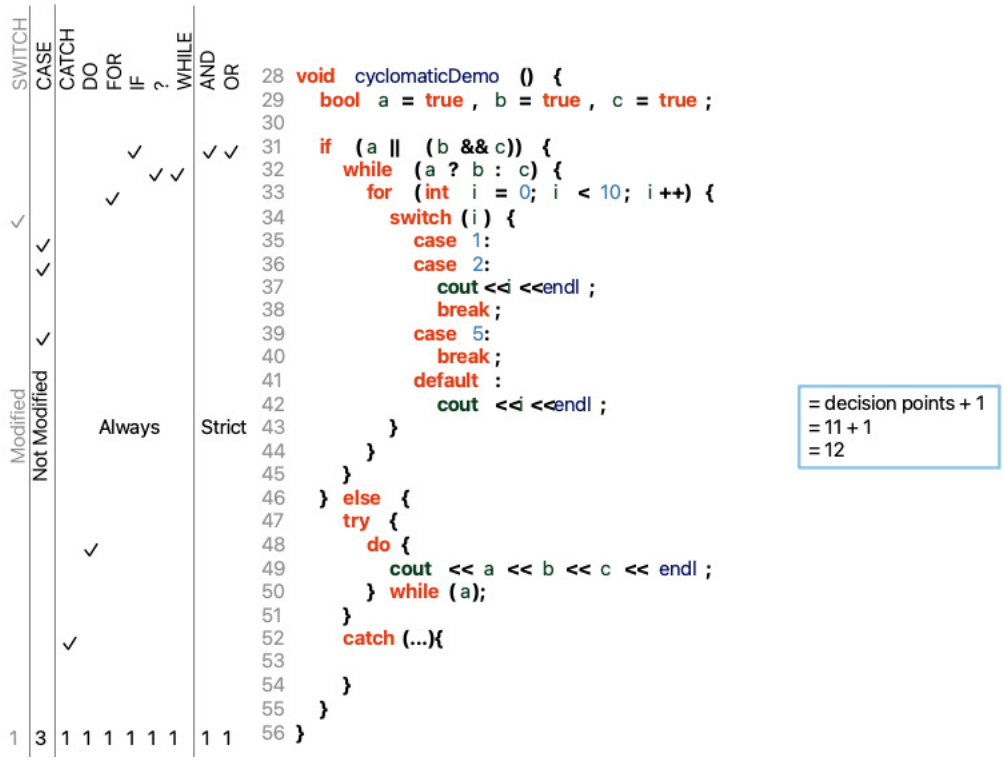
Strict Cyclomatic Complexity

API ID: CyclomaticStrict
Also Known As: Strict Cyclomatic Complexity (CC2)
Languages: Ada, Assembly, Basic, C#, C++, Fortran, Java, Jovial, Pascal, Python, Rust, VHDL, Web
Targets: Files, Functions, Modules, Packages, Subprograms

Strict McCabe Cyclomatic Complexity.

The Cyclomatic Complexity with logical conjunction and logical and in conditional expressions also adding 1 to the complexity for each of their occurrences. i.e. The statement 'if (a && b || c)' would have a cyclomatic complexity of 1 but a strict cyclomatic complexity of 3

Also known as McCabe - Strict Cyclomatic Complexity (CC2).



Targets By Language:

- **Ada:** Type, Entry, Function, Package, Procedure, Protected, Task
- **Basic:** Method
- **C#:** Method
- **C++:** Function
- **Fortran:** Module, Block Data, Function, Program, Subroutine
- **Java:** Method

- **Jovial:** Subroutine
- **Pascal:** Compunit, Function, Procedure
- **Python:** File, Function
- **Rust:** Function
- **VHDL:** Procedure, Function, Process
- **Web:** File

Configuration:

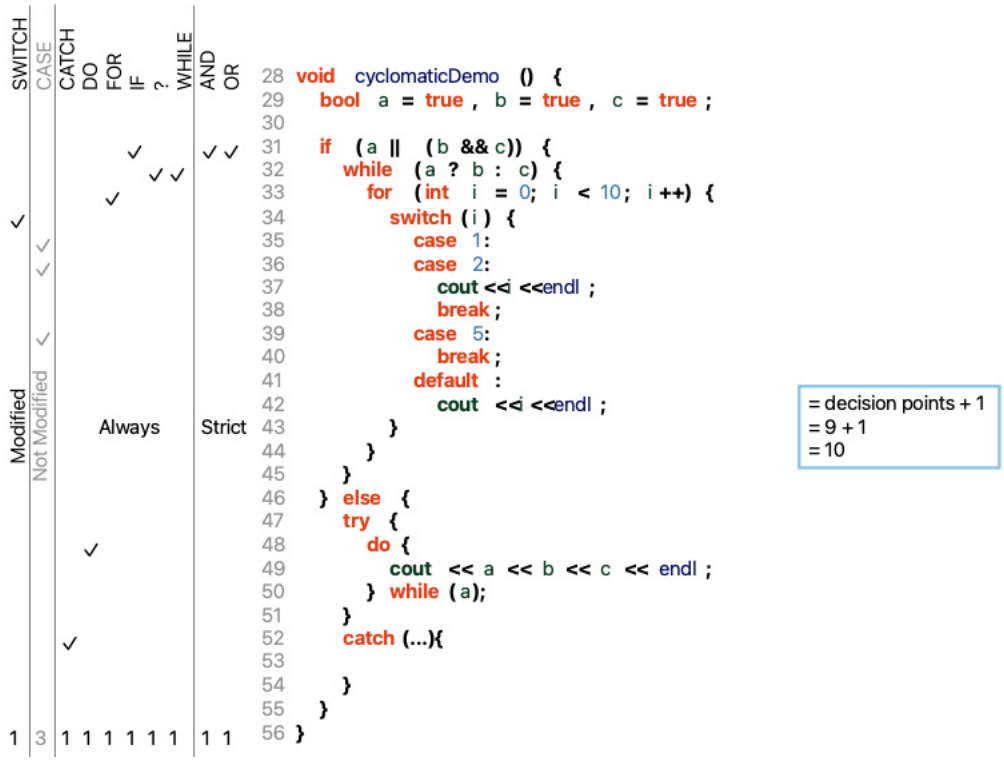
- The cyclomatic metric used is configured from "Project Configuration/ Metrics/Complexity". This metric results from having "Count each decision in a multi-way-decision structure" on and having "Count logical conjuctions (OR) and logical and (AND)" on. Checking "Show all available cyclomatic complexity metric variants" will also enable the metric.

Strict Modified Cyclomatic Complexity

API ID: CyclomaticStrictModified
Languages: Ada, Basic, C#, C++, Fortran, Java, Jovial, Pascal, Python, Rust, Web
Targets: Files, Functions, Modules, Packages, Subprograms

Cyclomatic Complexity with the following conditions.

(1) Logical operators (AND, OR) in conditional expressions add one (1) to the complexity for each occurrence.
(2) In multi-decision structures (switch in C/Java, Case in Ada, computed Goto, and arithmetic if in FORTRAN) the decision points are not counted, instead, the entire multi-way decision structure counts as 1



Targets By Language:

- **Ada:** Type, Entry, Function, Package, Procedure, Protected, Task
- **Basic:** Method
- **C#:** Method
- **C++:** Function
- **Fortran:** Module, Block Data, Function, Program, Subroutine
- **Java:** Method
- **Jovial:** Subroutine
- **Pascal:** Compunit, Function, Procedure
- **Python:** File, Function
- **Rust:** Function
- **Web:** File

Configuration:

- The cyclomatic metric used is configured from "Project Configuration/ Metrics/Complexity". This metric results from having "Count each decision in a multi-way-decision structure" off and having "Count logical conjuctions (OR) and logical and (AND)" on. Checking "Show all available cyclomatic complexity metric variants" will also enable the metric.

Strict Modified Essential Complexity

API ID: EssentialStrictModified
Languages: Ada
Targets: Packages, Subprograms

The cyclomatic complexity with short circuit operators (and then/or else) as unstructured but only adds one for all structured paths through case statements after graph reduction.

Targets By Language:

- **Ada:** Type, Entry, Function, Package, Procedure, Protected, Task

Configuration:

- This metric can be enabled from "Project Configuration/Metrics/Complexity" with the "Essential" option.

Strict Private Methods

API ID: CountDeclMethodStrictPrivate
Languages: Pascal
Targets: Architectures, Classes, Project

Number of local strict private methods.

Targets By Language:

- **Pascal:** Project, Class, Interface

Configuration:

- For projects and architectures, this metric must also be enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show aggregated class metrics" option. Enabling this option can be time consuming.
- This metric can be disabled from "Project Configuration/Metrics/Object Orientated" with the "Show method and variable count metrics" option.

Strict Published Methods

API ID: CountDeclMethodStrictPublished
Languages: Pascal
Targets: Architectures, Classes, Project

Number of local strict published methods.

Targets By Language:

- **Pascal:** Project, Class, Interface

Configuration:

- For projects and architectures, this metric must also be enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show aggregated class metrics" option. Enabling this option can be time consuming.
- This metric can be disabled from "Project Configuration/Metrics/Object Orientated" with the "Show method and variable count metrics" option.

Subprograms

API ID: CountDeclSubprogram
Languages: Ada, Basic, Fortran, Jovial, Pascal
Targets: Architectures, Files, Functions, Modules, Packages, Project, Subprograms

Number of subprograms.

Targets By Language:

- **Ada:** Project, File, Package
- **Basic:** Project, File
- **Fortran:** Project, File, Module, Block Data, Function, Program, Subroutine
- **Jovial:** Project, File, Module, Subroutine
- **Pascal:** Project, File, Compunit, Function, Procedure

Configuration:

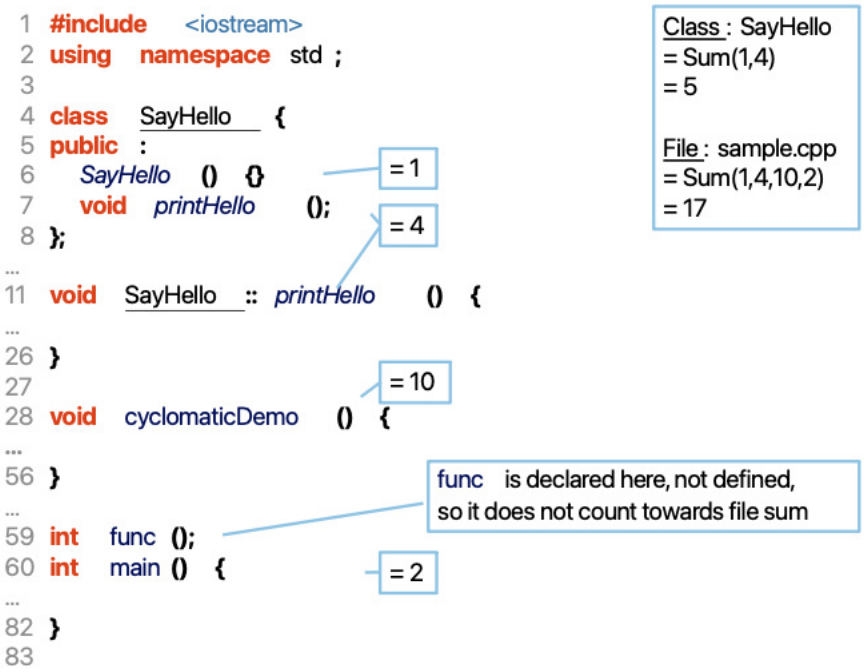
- For projects and architectures, this metric is enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show aggregated file metrics" option.
- For projects and architectures, this metric is enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show default summary metrics" option.

Sum Cyclomatic Complexity

API ID: SumCyclomatic
Also Known As: Weighted Methods per Class (WMC)
Languages: Ada, Basic, C#, C++, Fortran, Java, Jovial, Pascal, Python, Rust, Web
Targets: Architectures, Classes, Files, Functions, Modules, Packages, Project, Subprograms

Sum of cyclomatic complexity of all nested functions or methods.

Also known as Chidamber & Kemerer - Weighted Methods per Class (WMC).



Targets By Language:

- **Ada:** Project, File, Type, Entry, Function, Package, Procedure, Protected, Task
- **Basic:** Project, File, Method, Module, Class, Struct
- **C#:** Project, File, Class, Struct, Method
- **C++:** Project, File, Class, Struct, Union
- **Fortran:** Project, File, Module, Function, Program, Subroutine
- **Java:** Project, File, Class, Interface, Method
- **Jovial:** Project, File
- **Pascal:** Project, File, Class, Interface, Compunit, Function, Procedure
- **Python:** Project, File, Class
- **Rust:** Project, File
- **Web:** Project, File, PHP Class, PHP Interface

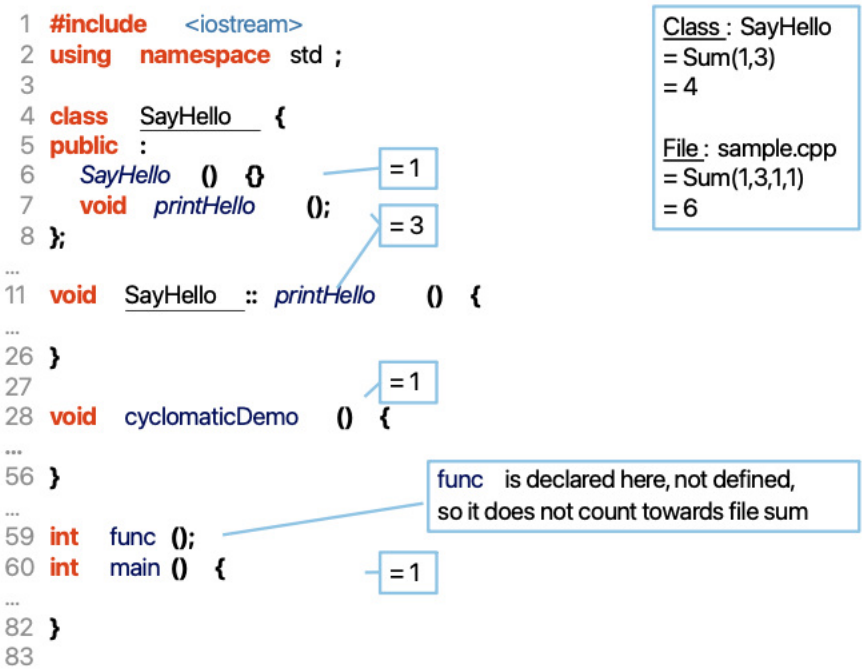
Configuration:

- For projects and architectures, this metric must also be enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show aggregated function metrics" option.
- The cyclomatic metric used is configured from "Project Configuration/Metrics/Complexity". This metric results from having "Count each decision in a multi-way-decision structure" on and having "Count logical conjuctions (OR) and logical and (AND)" off. Checking "Show all available cyclomatic complexity metric variants" will also enable the metric.

Sum Essential Complexity

API ID: SumEssential
Languages: Ada, Basic, C#, C++, Fortran, Java, Jovial, Pascal, Python, Rust, Web
Targets: Architectures, Classes, Files, Functions, Modules, Packages, Project, Subprograms

Sum of essential complexity of all nested functions or methods.



Targets By Language:

- **Ada:** Project, File, Type, Entry, Function, Package, Procedure, Protected, Task
- **Basic:** Project, File, Method, Module, Class, Struct
- **C#:** Project, File, Class, Struct, Method
- **C++:** Project, File, Class, Struct, Union
- **Fortran:** Project, File, Module, Function, Program, Subroutine
- **Java:** Project, File, Class, Interface, Method
- **Jovial:** Project, File
- **Pascal:** Project, File, Class, Interface, Compunit, Function, Procedure
- **Python:** Project, File, Class
- **Rust:** Project, File
- **Web:** Project, File, PHP Class, PHP Interface

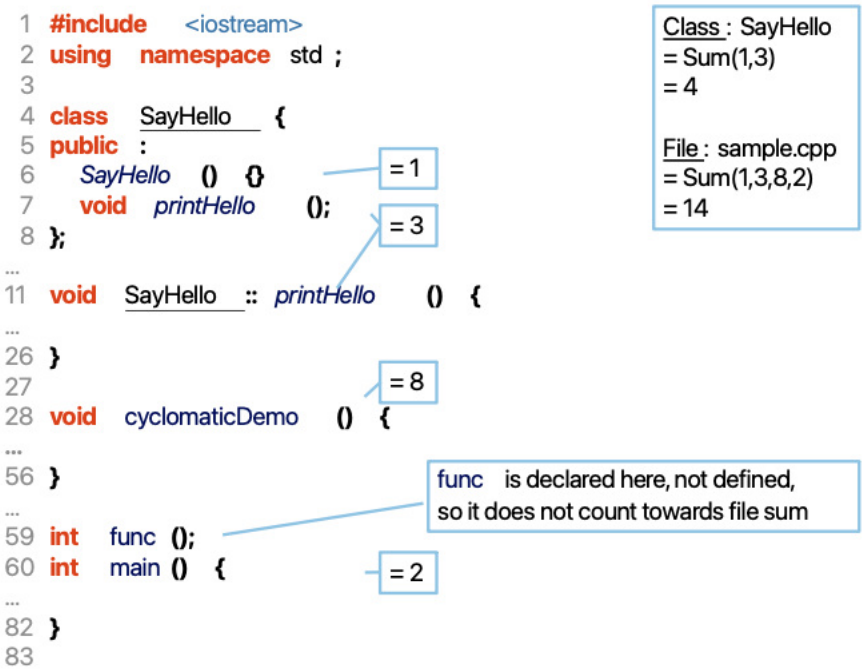
Configuration:

- For projects and architectures, this metric must also be enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show aggregated function metrics" option.
- This metric can be enabled from "Project Configuration/Metrics/Complexity" with the "Essential" option.

Sum Modified Cyclomatic Complexity

API ID: SumCyclomaticModified
Languages: Ada, Basic, C#, C++, Fortran, Java, Jovial, Pascal, Python, Rust, Web
Targets: Architectures, Classes, Files, Functions, Modules, Packages, Project, Subprograms

Sum of modified cyclomatic complexity of all nested functions or methods.



Targets By Language:

- **Ada:** Project, File, Type, Entry, Function, Package, Procedure, Protected, Task
- **Basic:** Project, File, Method, Module, Class, Struct
- **C#:** Project, File, Class, Struct, Method
- **C++:** Project, File, Class, Struct, Union
- **Fortran:** Project, File, Module, Function, Program, Subroutine
- **Java:** Project, File, Class, Interface, Method
- **Jovial:** Project, File
- **Pascal:** Project, File, Class, Interface, Compunit, Function, Procedure

- **Python:** Project, File, Class
- **Rust:** Project, File
- **Web:** Project, File, PHP Class, PHP Interface

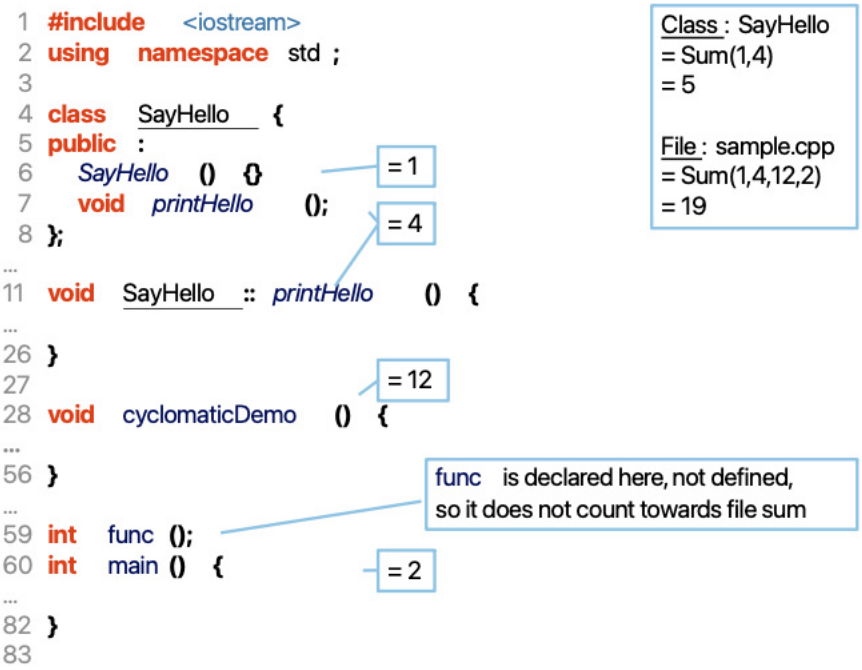
Configuration:

- For projects and architectures, this metric must also be enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show aggregated function metrics" option.
- The cyclomatic metric used is configured from "Project Configuration/Metrics/Complexity". This metric results from having "Count each decision in a multi-way-decision structure" off and having "Count logical conjuctions (OR) and logical and (AND)" off. Checking "Show all available cyclomatic complexity metric variants" will also enable the metric.

Sum Strict Cyclomatic Complexity

API ID: SumCyclomaticStrict
Languages: Ada, Basic, C#, C++, Fortran, Java, Jovial, Pascal, Python, Rust, Web
Targets: Architectures, Classes, Files, Functions, Modules, Packages, Project, Subprograms

Sum of strict cyclomatic complexity of all nested functions or methods.



Targets By Language:

- **Ada:** Project, File, Type, Entry, Function, Package, Procedure, Protected, Task
- **Basic:** Project, File, Method, Module, Class, Struct
- **C#:** Project, File, Class, Struct, Method
- **C++:** Project, File, Class, Struct, Union
- **Fortran:** Project, File, Module, Function, Program, Subroutine
- **Java:** Project, File, Class, Interface, Method
- **Jovial:** Project, File
- **Pascal:** Project, File, Class, Interface, Compunit, Function, Procedure
- **Python:** Project, File, Class
- **Rust:** Project, File
- **Web:** Project, File, PHP Class, PHP Interface

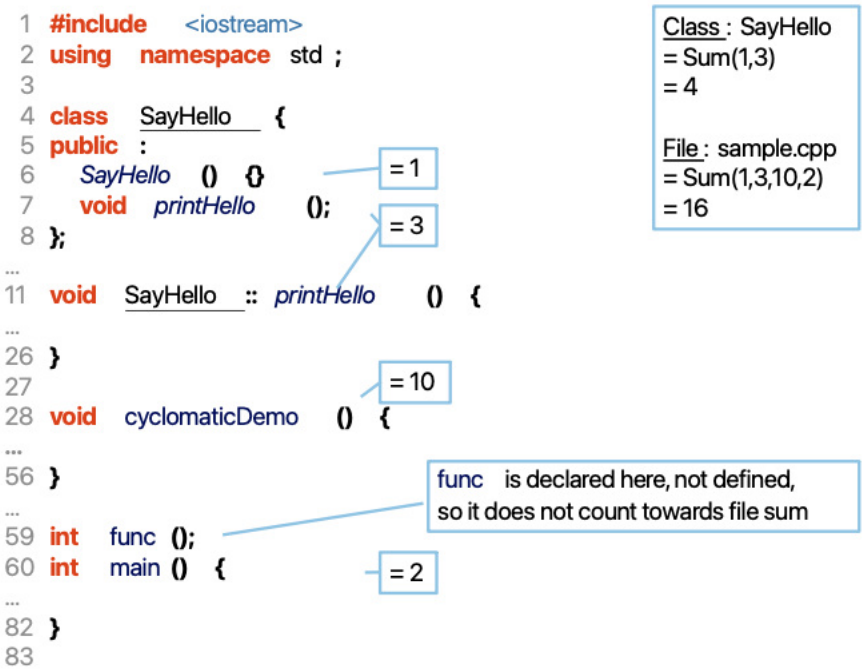
Configuration:

- For projects and architectures, this metric must also be enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show aggregated function metrics" option.
- The cyclomatic metric used is configured from "Project Configuration/Metrics/Complexity". This metric results from having "Count each decision in a multi-way-decision structure" on and having "Count logical conjuctions (OR) and logical and (AND)" on. Checking "Show all available cyclomatic complexity metric variants" will also enable the metric.

Sum Strict Modified Cyclomatic Complexity

API ID: SumCyclomaticStrictModified
Languages: Ada, Basic, C#, C++, Fortran, Java, Jovial, Pascal, Python, Rust, Web
Targets: Architectures, Classes, Files, Functions, Modules, Packages, Project, Subprograms

Sum of strict modified cyclomatic complexity of all nested functions or methods.



Targets By Language:

- **Ada:** Project, File, Type, Entry, Function, Package, Procedure, Protected, Task
- **Basic:** Project, File, Method, Module, Class, Struct
- **C#:** Project, File, Class, Struct, Method
- **C++:** Project, File, Class, Struct, Union
- **Fortran:** Project, File, Module, Function, Program, Subroutine
- **Java:** Project, File, Class, Interface, Method
- **Jovial:** Project, File
- **Pascal:** Project, File, Class, Interface, Compunit, Function, Procedure
- **Python:** Project, File, Class
- **Rust:** Project, File
- **Web:** Project, File, PHP Class, PHP Interface

Configuration:

- For projects and architectures, this metric must also be enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show aggregated function metrics" option.
- The cyclomatic metric used is configured from "Project Configuration/Metrics/Complexity". This metric results from having "Count each decision in a multi-way-decision structure" off and having "Count logical conjuctions (OR) and logical and (AND)" on. Checking "Show all available cyclomatic complexity metric variants" will also enable the metric.

Sum Strict Modified Essential Complexity

API ID: SumEssentialStrictModified
Languages: Ada
Targets: Architectures, Files, Packages, Project

Sum of strict modified essential complexity of all nested functions or methods.

Targets By Language:

- **Ada:** Project, File, Package

Configuration:

- For projects and architectures, this metric must also be enabled from "Project Configuration/Metrics/Projects and Architectures" with the "Show aggregated function metrics" option.
- This metric can be enabled from "Project Configuration/Metrics/Complexity" with the "Essential" option.