# Coding Standards

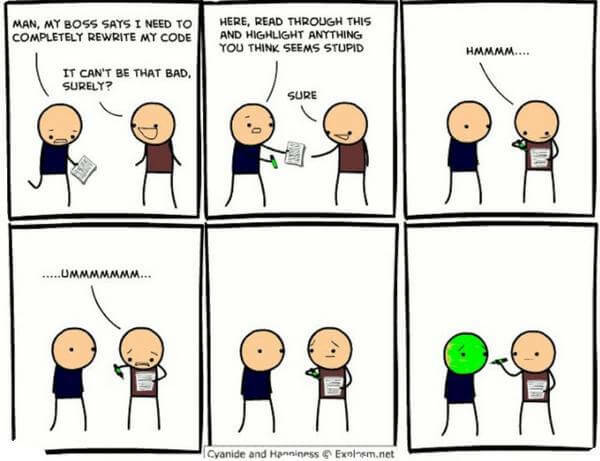
Coding standards are a set of guidelines for programmers to follow when writing their software. These standards can cover several aspects such as naming conventions, indentation standards, code organization, commenting and handling exceptions. The aim of coding standards is to increase code readability, maintainability and consistency.

A cartoon of two people at a desk

AI-generated content may be incorrect.

Use of coding standards are a good way to ensure good practises when coding such as:

* **Consistency:** Coding standards ensure uniformity across codebases and teams, making it easier for developers to read, understand, and maintain code.
* **Readability:** Well-defined standards enhance code readability, reducing errors and improving collaboration.
* **Error Prevention:**Consistent practices help catch common mistakes early, preventing bugs and improving code quality.
* **Scalability:** Adhering to standards ensures code can scale without becoming unwieldy or unmanageable.
* **Cross-Team Collaboration:**Shared standards facilitate collaboration among developers, even in large teams.
* **Code Reviews:**Standards provide clear criteria for code reviews, leading to more effective feedback.
* **Efficient Maintenance:**Following standards simplifies debugging, refactoring, and maintenance tasks.



Standards can be defined at many different levels, from personal preference, organisation standards and open coding standards. Open coding standards are publicly available allowing for community feedback and improvement. Examples of public coding standards include:

* Microsoft   
  https://learn.microsoft.com/en-us/dotnet/csharp/fundamentals/coding-style/coding-conventions#
* SEI CERT https://wiki.sei.cmu.edu/confluence/display/c/SEI+CERT+C+Coding+Standard

In contrast, closed organisation standards are only enforced within the organisation.

The standards required by an organisation vary from place to place, but can include guidelines covering the following areas:

* **Languages** to be used – does the organisation only use a few specific languages, or specific languages for specific tasks. Each language can have its own variant of guidelines. javascript vs typescript, powershell or batch files.
* **Language** **constraints** – restricts the use of language features. For example no goto statements, use of datatypes for specific purposes, error handling methods, use of var instead of strongly typed variables.
* **Commenting** – Coding standards can define how comments are to be used. Are they required to be formatted in a specific way, what information is to be included.
* **Layout** of code blocks – Use of white space or tabs for indentation, the requirements of brace alignment etc.
* **Naming conventions** – How routines and variables should be named including capitalisation schemes (eg CamelCase, pascalCase, Snake\_Case or Kebab-Case).
* **Logging** – Logs are an essential tool for debugging and coding standards can help by defining the methods of logging to be used and how often. Where will logs be stored? (flat files or a database), how long will they be retained for and what information will be logged.

Commonly programming involves the use of Refactoring, where working code is re-written without changing its external behaviour. This can be done for clarity, better implementation of coding standards, and to improve efficiency or performance. For example, a search routine could be changed to use a more efficient algorithm without having to change any other code.

Automated tools can warn programmers when they do not follow coding standards, or can even format code to follow them. For example, in Visual Studio IntelliSense and error messages are used to warn developers:

