

# Clean Code Development (CCD)

**Grades**  
**1 Red**  
**Principles**  
Don't repeat yourself (DRY)  
Keep it simple, stupid (KISS)  
Beware of optimizations  
Favour composition over inheritance (FCoI)  
**Practices**  
Use a version control system  
Refactoring patterns, rename and extract method  
Leave the code cleaner than when you found it  
Daily reflection  
Always look for the root cause of a problem

**2 Orange**  
**Principles**  
Single Level of Abstraction (SLA)  
Single responsibility principle (SRP)  
Separation of concerns (SoC)  
Source code convention  
**Practices**  
Issue tracking  
Automatic integration tests  
Reviews

**3 Yellow**  
**Principles**  
Information hiding principle  
Principle of least astonishment  
Liskov substitution principle (LSP)  
Interface segregation principle (ISP)  
Dependency inversion principle (DIP)  
**Practices**  
Automatic unit tests  
Mockups  
Code coverage analysis  
Complex refactoring

**4 Green**  
**Principles**  
Open closed principle (OCP)  
    An implementation should be open for extension but closed for modification  
Tell, don't ask  
    Tell methods what to do instead of asking for their internal state (information hiding)  
    Promotes cohesion and loose coupling  
Law of Demeter  
    Units should have limited knowledge about other units except for closely related units  
**Practices**  
Continuous integration (CI)  
Static code analysis (metrics)  
Inversion of control container (IoC)

**5 Blue**  
**Principles**  
Separation of design and implementation  
    Design breaks down software into components, defines dependencies and contracts  
Implementation reflects design  
    Ideally logical structures in architecture manifest as physical as possible  
You aren't gonna need it (YAGNI)  
    Implement functionality only when it is actually needed  
    When in doubt, decide against the effort  
**Practices**  
Continuous delivery (CD)  
Iterative development  
Component Orientation  
Test-driven development (TDD)  
  
Source: [clean-code-developer.de](https://clean-code-developer.de)

**Additional rules**  
  
**General rules**  
Follow standard conventions  
Be consistent  
Do not override safeties  
  
**Naming**  
Use descriptive and unambiguous variable names  
Make meaningful distinctions  
Use pronounceable names  
Use searchable names  
  
**Functions**  
**Should be...**  
    doing one thing  
    relatively small  
**Should have...**  
    descriptive names  
    as few arguments as possible  
    no side effects  
    explanatory variables  
  
**Comments**  
Try to explain yourself in code, not in comment  
Use as explanation of intent  
Use as clarification of code  
Use to inform or warn the reader  
Don't be redundant  
Don't add obvious noise  
Remove code instead of commenting it out  
  
Sources:  
[cheatography.com/costemaxime/cheat-sheets/summary-of-clean-code-by-robert-c-martin](https://cheatography.com/costemaxime/cheat-sheets/summary-of-clean-code-by-robert-c-martin)  
[gist.github.com/wojtekl/73c6914cc446146b8b533c0988cf8d29](https://gist.github.com/wojtekl/73c6914cc446146b8b533c0988cf8d29)