

### Principles

<b>Single Responsibility</b>	A class should have only one reason to change
<b>Open / Close principle</b>	Software entities (classes, modules, functions, etc.) should be open for extension but closed for modification
<b>Liskov Substitution Principle</b>	Subtypes must be substitutable for their base types
<b>Interface Segregation Principle</b>	Clients should not be forced to depend on methods they do not use.
<b>Dependency Inversion Principle</b>	High-level modules should not depend on low-level modules. Both should depend on abstractions. Abstractions should not depend upon details. Details should depend upon abstractions
<b>Command Query Separation</b>	A method should either be a command or a query
<b>Do not Repeat Yourself (DRY)</b>	Every piece of knowledge must have a single, unambiguous, authoritative representation within a system

### General

<b>Boy Scout Rule</b>	Leave the campground cleaner than you found it
<b>Keep it Simple, Stupid (KISS)</b>	Simpler is always better. Reduce complexity as much as possible
<b>Positive Conditionals</b>	Positive conditionals are easier to read than negative conditionals
<b>Don't Be Arbitrary</b>	



By Galad  
[cheatography.com/galad/](https://cheatography.com/galad/)

Not published yet.  
Last updated 3rd February, 2015.  
Page 1 of 2.

Sponsored by **Readability-Score.com**  
Measure your website readability!  
<https://readability-score.com>

### Design

#### Prefer Polymorphism To If/Else or Switch/Case

#### Prefer composition over inheritance

**Prefer Dedicated Value Objects to Primitive Types** Instead of passing primitive types like strings and integers, use dedicated primitive types: e.g. *Uri* instead of *string*

**Vertical Separation** Variables and methods should be defined close to where they are used

**Encapsulate Conditionals** `if (this.ShouldBeDeleted(timer))` is preferable to `if (timer.HasExpired && !timer.IsRecurrent)`

**(-) Method with Too Many Arguments** Prefer fewer arguments. Maybe functionality can be outsourced to a dedicated class that holds the information in fields

**(-) Method with Out/Ref Arguments** Prevent usage. Return complex object holding all values, split into several methods

**(-) Obscured Intent** Too dense algorithms that lose all expressiveness

**(-) Magic Numbers / Strings** Replace Magic Numbers and Strings with named constants to give them a meaningful name when meaning cannot be derived from the value itself

**(-) Dead code / dead comment** Delete unused things. You can find them in your version control system



By Galad  
[cheatography.com/galad/](https://cheatography.com/galad/)

Not published yet.  
Last updated 3rd February, 2015.  
Page 2 of 2.

Sponsored by **Readability-Score.com**  
Measure your website readability!  
<https://readability-score.com>