

# SOLID (object-oriented design)

---

SOLID
Principles
<ul style="list-style-type: none"><li>• Single responsibility</li><li>• Open/closed</li><li>• Liskov substitution</li><li>• Interface segregation</li><li>• Dependency inversion</li></ul>

In computer programming, **SOLID** (**S**ingle **r**esponsibility, **O**pen-closed, **L**iskov substitution, **I**nterface segregation and **D**ependency inversion) is a mnemonic acronym introduced by Michael Feathers for the "first five principles" identified by Robert C. Martin<sup>[1][2]</sup> in the early 2000s<sup>[3]</sup> that stands for five basic principles of object-oriented programming and design. The principles when applied together intend to make it more likely that a programmer will create a system that is easy to maintain and extend over time.<sup>[3]</sup> The principles of SOLID are guidelines that can be applied while working on software to remove code smells by causing the programmer to refactor the software's source code until it is both legible and extensible. It is typically used with test-driven development, and is part of an overall strategy of agile and adaptive programming.<sup>[3][4]</sup>

## Overview

Initial	Stands for (acronym)	Concept
<b>S</b>	SRP	Single responsibility principle a class should have only a single responsibility.
<b>O</b>	OCP	Open/closed principle "software entities ... should be open for extension, but closed for modification".
<b>L</b>	LSP	Liskov substitution principle "objects in a program should be replaceable with instances of their subtypes without altering the correctness of that program". See also design by contract.
<b>I</b>	ISP	Interface segregation principle "many client-specific interfaces are better than one general-purpose interface." <sup>[5]</sup>
<b>D</b>	DIP	Dependency inversion principle one should "Depend upon Abstractions. Do not depend upon concretions." <sup>[5]</sup> Dependency injection is one method of following this principle.

## References

- [1] "Principles Of OOD" (<http://butunclebob.com/ArticleS.UncleBob.PrinciplesOfOod>), Robert C. Martin ("Uncle Bob"), butunclebob.com, Last verified 2009-01-14. (Note the "first five principles", though the acronym is not used in this article.) Dates back to at least 2003.
  - [2] "Getting a SOLID start." (<http://blog.objectmentor.com/articles/2009/02/12/getting-a-solid-start>), Robert C. Martin ("Uncle Bob"), objectmentor.com. Last verified 2009-01-14.
  - [3] "SOLID Object-Oriented Design" (<http://www.confreaks.com/videos/240-goruco2009-solid-object-oriented-design>), Sandi Metz (Duke University), Talk given at the 2009 Gotham Ruby Conference in May, 2009. Last verified 2009-01-15.
  - [4] "Introducing SOLID Object-Oriented Design Principles and Microsoft Unity" ([http://nonprofits.accesscomm.ca/springseminar/nug/Introducing\\_SOLID.ppt](http://nonprofits.accesscomm.ca/springseminar/nug/Introducing_SOLID.ppt)), Uwe Schmitz, Presentation given at the Regina .NET User Group in May, 2009. Last verified 2009-01-14.
  - [5] "Design Principles and Design Patterns" ([http://www.objectmentor.com/resources/articles/Principles\\_and\\_Patterns.pdf](http://www.objectmentor.com/resources/articles/Principles_and_Patterns.pdf)), Robert C. Martin ("Uncle Bob"), objectmentor.com. Last verified 2009-01-14.
-

# Article Sources and Contributors

**SOLID (object-oriented design)** *Source:* <http://en.wikipedia.org/w/index.php?oldid=547254055> *Contributors:* Abatishchev, Angela Harms, Bastie, Cambalachero, CornellEngr2008, Cyfal, Danielflippance, Frap, Magioladitis, Michael B. Trausch, Michael Hardy, MrOllie, Nightfly85, Pablomendes, Pcap, Ptrb, Rob Bednark, Sae1962, Subversive.sound, Svick, 31 anonymous edits

## License

---

Creative Commons Attribution-Share Alike 3.0 Unported  
[//creativecommons.org/licenses/by-sa/3.0/](https://creativecommons.org/licenses/by-sa/3.0/)

---