

# The SOLID Principles, Explained with Motivational Posters

 By **Joey deVilla** < <https://www.globalnerdy.com/author/joey/> >



**July 15, 2009** < <https://www.globalnerdy.com/2009/07/15/the-solid-principles-explained-with-motivational-posters/> >

[This article also appears in \*Canadian Developer Connection\*. < <http://blogs.msdn.com/cdndevs/archive/2009/07/15/the-solid-principles-explained-with-motivational-posters.aspx> >](#)

## SOLID Seems to be Everywhere These Days



# SOLID

Software development is not a Jenga game.

[Robert C. “Uncle Bob” Martin <](http://en.wikipedia.org/wiki/Robert_Cecil_Martin)

[http://en.wikipedia.org/wiki/Robert\\_Cecil\\_Martin>](http://en.wikipedia.org/wiki/Robert_Cecil_Martin)

**first gathered [the object-oriented design principles that would eventually go under the acronym SOLID](#)**

[≤](#)

[http://butunclebob.com/ArticleS.UncleBob.Principles](http://butunclebob.com/ArticleS.UncleBob.PrinciplesOfOOD)  
[sOfOod>](http://butunclebob.com/ArticleS.UncleBob.PrinciplesOfOOD) **almost fifteen years ago,** but the SOLID

principles have been making the rounds lately.

Before I talk about the SOLID principles, let me show you where the concept has been popping up in the past few months...

## SOLID at Conferences

At the [Microsoft TechEd conference in May < http://www.microsoft.com/events/teched2009/default.aspx>](http://www.microsoft.com/events/teched2009/default.aspx), Steve Smith gave [a presentation on applying the SOLID principles to developing applications with the ASP.NET MVC framework < http://stevesmithblog.com/blog/teched-2009-session-aftermath/>](http://stevesmithblog.com/blog/teched-2009-session-aftermath/). At around the same time, the [GeeCon < http://www.geecon.org/>](http://www.geecon.org/) conference in Krakow, Poland had a couple of presentations on SOLID: [Bruno Bossola gave a talk on the SOLID principles < http://www.slideshare.net/bbossola/geecon09-solid>](http://www.slideshare.net/bbossola/geecon09-solid) and [Lee Brandt gave a talk on spotting SOLID code “in the wild” < http://www.slideshare.net/leebrandt/solid-principles-in-the-wild>](http://www.slideshare.net/leebrandt/solid-principles-in-the-wild).

## SOLID at User Groups

A number of user groups have had recently presentations on SOLID, such as:

- [ATL ALT.NET group \(Atlanta < http://www.developerfusion.com/event/13495/atl-altnet-may-26-solid-principles-the-10-commandments-of-programming/> \)](http://www.developerfusion.com/event/13495/atl-altnet-may-26-solid-principles-the-10-commandments-of-programming/)
- [Austin .NET User Group < http://www.lostechies.com/blogs/derickbailey/arc](http://www.lostechies.com/blogs/derickbailey/arc)

[hive/2008/10/14/thanks-adnug-attendees-slides-and-code-available.aspx](http://hive/2008/10/14/thanks-adnug-attendees-slides-and-code-available.aspx)>

- [North Dalls .NET User Group](http://niddnug.net/meetings/july-meeting-derick-bailey-on-the-s-o-l-i-d-design-principle/) < <http://niddnug.net/meetings/july-meeting-derick-bailey-on-the-s-o-l-i-d-design-principle/>>
- [Pivotal Labs \(San Francisco\)](http://pivotallabs.com/users/miked/blog/articles/876-solid-object-oriented-design-sandi-metz-) < <http://pivotallabs.com/users/miked/blog/articles/876-solid-object-oriented-design-sandi-metz->>
- [Regina .NET User Group](http://geekswithblogs.net/gpro/archive/2009/04/08/regina-events.aspx) < <http://geekswithblogs.net/gpro/archive/2009/04/08/regina-events.aspx>>
- [Space Coast .NET User Group](http://stevesmithblog.com/blog/space-coast-asp-net-mvc-and-solid-principles-talk/) < <http://stevesmithblog.com/blog/space-coast-asp-net-mvc-and-solid-principles-talk/>>
- [Wisconsin .NET Users Group](http://wineta.org/DesktopDefault.aspx?tabid=58) < <http://wineta.org/DesktopDefault.aspx?tabid=58>>
- and there'll be a talk on SOLID at the [Isle of Man Dev Group](http://fuzzelogicsolutions.com/wordpress/index.php/2009/07/13/dev-g-round-3-solid-start/) < <http://fuzzelogicsolutions.com/wordpress/index.php/2009/07/13/dev-g-round-3-solid-start/>>  
tomorrow (the 17th)

## Stack Overflow on SOLID

SOLID was the topic of a recent argument that took place over a series of podcasts earlier this year:

- In *Stack Overflow* podcast 38, [Joel “Joel on Software / Stack Overflow” Spolsky](#) expressed his [doubts about the real-world usefulness of SOLID](#) <

<http://www.joelonsoftware.com/items/2009/01/31.html>> (among other things he also considers to be

stuff that [“Architecture Astronauts”](#) <

<http://www.joelonsoftware.com/articles/fog000000018.html>> like)...

- [So naturally “Uncle Bob” responded... <](#)  
<http://blog.objectmentor.com/articles/2009/02/06/on-open-letter-to-joel-spolsky-and-jeff-atwood>>
- and then [Joel and Jeff had him as a guest of Stack Overflow podcast 41 <](#)  
<https://stackoverflow.fogbugz.com/default.asp?W29030>>
- [the SOLID scuffle was covered at InfoQ, <](#)  
<http://www.infoq.com/news/2009/02/spolsky-vs-uncle-bob>>
- and after that, [he was a guest on Hanselminutes podcast 150. <](#)  
<http://hanselminutes.com/default.aspx?showID=168>>

## SOLID in Blogs

SOLID has also been spotted in a number of blogs:

- [Dave Tchepek blogged about SOLID in January <](#)  
<http://www.davesquared.net/2009/01/introduction-to-solid-principles-of-oo.html>>
- [“Uncle Bob” himself posted an article on getting started with SOLID in February <](#)

<http://blog.objectmentor.com/articles/2009/02/12/getting-a-solid-start>>

- [LosTechies.com](http://www.lostechies.com/blogs/chad_myers/archive/2008/03/07/pablo-s-topic-of-the-month-march-solid-principles.aspx) made SOLID their Topic of the Month in March < [http://www.lostechies.com/blogs/chad\\_myers/archive/2008/03/07/pablo-s-topic-of-the-month-march-solid-principles.aspx](http://www.lostechies.com/blogs/chad_myers/archive/2008/03/07/pablo-s-topic-of-the-month-march-solid-principles.aspx)>
- [Sandi Metz](http://sandimetz.com/2009/03/solid-design-principles-dependency.html) blogged about SOLID in March as well < <http://sandimetz.com/2009/03/solid-design-principles-dependency.html>>
- [Steve Bohlen](http://unhandled-exceptions.com/blog/index.php/2009/07/13/why-i-dont-listen-to-the-stackoverflow-podcast-any-more/) seems to have sworn off [Stack Overflow's](http://unhandled-exceptions.com/blog/index.php/2009/07/13/why-i-dont-listen-to-the-stackoverflow-podcast-any-more/) podcast as a result of the SOLID scuffle < <http://unhandled-exceptions.com/blog/index.php/2009/07/13/why-i-dont-listen-to-the-stackoverflow-podcast-any-more/>>

**Will There be a Presentation on SOLID at TechDays Canada 2009?**

*Mmmmmmmmaybe...*

**And Now, The Principles and Posters**

[The folks at LosTechies.com](http://www.lostechies.com/blogs/derickbailey/archive/2009/02/11/solid-development-principles-in-motivational-pictures.aspx) have created a series of Creative Commons-licenced posters that illustrate the SOLID principles. < <http://www.lostechies.com/blogs/derickbailey/archive/2009/02/11/solid-development-principles-in-motivational-pictures.aspx>> I took the posters,

cleaned up the typography a little and posted them below under the same Creative Commons licence. Below each poster is an explanation of the corresponding SOLID principle. Enjoy!

## Single Responsibility Principle



## Single Responsibility Principle

Just because you *can* doesn't mean you *should*.

< [https://www.globalnerdy.com/wordpress/wp-content/uploads/2009/07/single\\_responsibility\\_principle.jpg](https://www.globalnerdy.com/wordpress/wp-content/uploads/2009/07/single_responsibility_principle.jpg) >

The “S” in SOLID is for ***Single Responsibility Principle***, which states that every object should have a single responsibility and that all of its services should be aligned with that responsibility. “Responsibility” is defined as “a reason to change”, and [Wikipedia does a](#)

[pretty good job of explaining it <http://en.wikipedia.org/wiki/Single\\_responsibility\\_principle>](http://en.wikipedia.org/wiki/Single_responsibility_principle) :

As an example, consider a module that compiles and prints a report. Such a module can be changed for two reasons. First, the content of the report can change. Second, the format of the report can change. These two things change for very different causes; one substantive, and one cosmetic. The single responsibility principle says that these two aspects of the problem are really two separate responsibilities, and should therefore be in separate classes or modules. It would be a bad design to couple two things that change for different reasons at different times.



## Open-Closed Principle



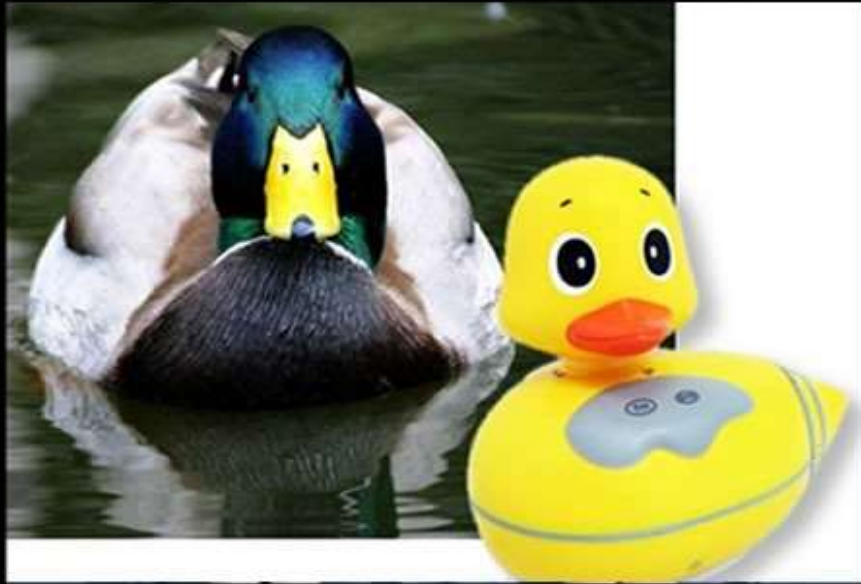
## Open-Closed Principle

Open-chest surgery isn't needed when putting on a coat.

[https://www.globalnerdy.com/wordpress/wp-content/uploads/2009/07/openclosed\\_principle.jpg](https://www.globalnerdy.com/wordpress/wp-content/uploads/2009/07/openclosed_principle.jpg)

The “O” in **SOLID** is for **Open-Closed Principle**, which states that software entities – such as classes, modules, functions and so on – should be open for extension but closed for modification. The idea is that it's often better to make changes to things like classes by adding to or building on top of them (using mechanisms like subclassing or polymorphism) rather than modifying their code.

## Liskov Substitution Principle



## Liskov Substitution Principle

If it looks like a duck and quacks like a duck but needs batteries, you probably have the wrong abstraction.

[https://www.globalnerdy.com/wordpress/wp-content/uploads/2009/07/liskov\\_substitution\\_principle.jpg](https://www.globalnerdy.com/wordpress/wp-content/uploads/2009/07/liskov_substitution_principle.jpg)

The “L” in SOLID is for ***Liskov Substitution Principle***, which states that subclasses should be substitutable for the classes from which they were derived. For example, if MySubclass is a subclass of MyClass, you should be able to replace MyClass with MySubclass without bugging up the program.

## Interface Segregation Principle



## Interface Segregation Principle

You want me to plug this in *where*?

< [https://www.globalnerdy.com/wordpress/wp-content/uploads/2009/07/interface\\_segregation\\_principle.jpg](https://www.globalnerdy.com/wordpress/wp-content/uploads/2009/07/interface_segregation_principle.jpg) >

### The “I” in SOLID is for **Interface Segregation**

**Principle**, which states that clients should not be forced to depend on methods they don’t use. If a class exposes so many members that those members can be broken down into groups that serve different clients that don’t use members from the other groups, you should think about exposing those member groups as separate interfaces.



## Dependency Inversion Principle

Would you solder a lamp directly to the electrical wiring in a wall?

< [https://www.globalnerdy.com/wordpress/wp-content/uploads/2009/07/dependency\\_inversion\\_principle.jpg](https://www.globalnerdy.com/wordpress/wp-content/uploads/2009/07/dependency_inversion_principle.jpg) >

The “D” in **SOLID** is for ***Dependency Inversion Principle***, which states that high-level modules shouldn’t depend on low-level modules, but both should depend on shared abstractions. In addition, abstractions should not depend on details – instead, details should depend on abstractions.

### Bonus Reading Material on SOLID

In addition to the *LosTechies.com* posters, they’ve also produced a book covering the SOLID principles in detail. Go and download [\*\*\*Pablo’s Solid Software\*\*\*](#)

[Development <](#)

[http://cdn.cloudfiles.mosso.com/c82752/pablos\\_solid\\_ebook.pdf](http://cdn.cloudfiles.mosso.com/c82752/pablos_solid_ebook.pdf) [1.4 MB PDF] now!

---

© 2024 Global Nerdy < <https://www.globalnerdy.com/> >

To the top ↑