

## Re: Static vs. Dynamic typing

**Source:** <http://coding.derkeiler.com/Archive/General/comp.object/2004-09/0503.html>

---

**From:** Christopher Barber (*cbarber\_at\_curl.com*)

**Date:** 09/10/04

Date: 10 Sep 2004 11:01:52 -0400

"Ilja Preuß" <it@iljapreuss.de> writes:

> *Cristiano Sadun wrote:*

>

>> *It just came to me that STC can be seen as a test in itself – in TDD*

>> *sense: it can be executed before the functionality is written or*

>> *available, then fixing the code makes it pass the test. "execution"*

>> *is the set of type– check steps of the compiler.*

>>

>> *For example, trying to invoke an unexisting method in a class (this*

>> *example works better for OO rather than procedural, since methods are*

>> *part of the type definition) results in a test (compilation) failure;*

>> *or not having a return statement of the proper type; etc.*

>

> *A nice example. When using a dynamically typed language, the same tests*

> *fails, too, because of the same problem (object does not understand*

> *message) – just at runtime instead of compile time. So in this specific*

> *case, the compile time check doesn't seem to buy us anything, as far as I*

> *can tell...*

Without static checking this will only fail if the call is actually executed during testing. All it takes for the static check to fail is for the code to be compiled. Without static checks it is more important that your tests cover every line of code and more thoroughly as well. If you consider the static declarations themselves to be tests, then using them really shouldn't be any worse than writing tests.