

Cobra Programming Language

[Downloads](#)[Documentation](#)[How To](#)[Samples](#)[Discussion](#)[Wiki](#)[Contact](#)

Why Cobra?

(This article was translated to the [Serbo-Croatian](#) language by [WHGeeks](#).)

There are plenty of object-oriented programming languages in existence including C#, Python and Ruby. So why use Cobra? This document addresses that question.

Right now, if you want software contracts in your language, how can you get them? The answer is to use Eiffel or D. What if you want static and dynamic binding? Use Objective-C or Boo. What if you want expressiveness and quick coding? Use Python, Ruby or Smalltalk. What if you want runtime performance? Use C#, Java, C++, etc. What if you want first class language support for unit tests? Use D.

But what if you want *all* of those? ... You can't get them! And that's frustrating because none of those productivity-boosting features are incompatible with each other. You shouldn't have to choose between C++'s speed, Python's expressiveness and Eiffel's contracts. There's no theoretical reason that you can't have it all. There's "just" a lot of work required to make it happen.

One way to characterize Cobra is with these high level points:

1. Quick, expressive coding
2. Fast execution
3. Static and dynamic binding
4. Language level support for quality

Cobra achieves 1 by following Python and Ruby (but not religiously). It achieves 2 by favoring static typing ("i = 5" means "i" is an integer and always will be) and leveraging .NET/Mono for machine code generation. It does 3 by using the .NET typing system at compile-time for static types, and using the .NET run-time for dynamic binding. It takes language features for 4 from multiple sources including Eiffel, Python and its own compile-time nil tracking.

There are additional refinements in Cobra such as defaulting to accurate decimal math and providing detailed postmortem exception reports.

So the "what's new" in Cobra is not the individual elements such as contracts, classes, etc. It's the combination of everything that goes into it. Consider what would happen if you were coding in Cobra, then switched to another language:

- If moving from Cobra to Python, you would give up optional static typing, contracts, compile-time nil tracking, closures and more.
- If moving from Cobra to C#, you would give up native contracts, clean collection literals, expressive syntax, uniform compile-time nil tracking, mixins and more.

Note that Cobra has few new elements because it was never intended to be experimental. It's a practical synthesis of already-proven features that are currently scattered across multiple languages.

Current Status

As of Winter 2013, Cobra can be considered a "late beta" offering. Features include:

- OOP and friends:
 - classes, interfaces, structs
 - methods, properties, indexers
 - mixins, extension methods

- generics, attributes
- Quality control:
 - contracts, assertions
 - unit tests, doc strings
 - compile-time nil tracking
- Expressiveness:
 - static and dynamic binding
 - literals for lists, dictionaries and sets
 - [in](#) and [implies](#) operators
 - [for](#) and [if](#) expressions
 - slicing
 - interpolated strings
 - compile-time type inference
 - lambdas and closures
 - optional parameters, default values
- General productivity:
 - exception handling
 - postmortem exception report
 - garbage collection
- Scripting conveniences
 - clean syntax
 - one step run
 - pound-bang line (#!)
- Compiled language performance

Built-in tools include:

- Compiler
- Unit test runner
- Document generator
- Syntax highlighter
- Shared/static data lister
- Exception reporter

Other tools such as debuggers and profilers are readily available from the .NET/Mono ecosystem.

Other Notes

A port to JVM is underway and could use more volunteers.

Modern IDE support is available for MonoDevelop / Xamarin Studio.

Cobra is open source under the permissive MIT license.

Where to next?

To learn the very basics of the language, which will be helpful when reading everything else, see [Hello, world. \(and more\)](#).

You'll eventually be lead to the following links, but maybe you also want to check them out now:

There are more details on [language level support for quality](#).

There is a [detailed comparison to Python \(and IronPython\)](#).

There are [monthly updates](#).

And since Cobra draws so much from other languages, there are [acknowledgements](#).

Home

[Downloads](#)
[How To](#)
[Samples](#)
[Discussion](#)
[Source](#)
[Contact](#)

Wiki

[How to install from source](#)
[Get support](#)
[Command line](#)
[Troubleshooting](#)
[Language topics](#)
[Editor support](#)

User Testimonial

"I'm really getting the hang of testing Cobra-style.
I feel more productive (and willing to write tests)."
- Todd Alexander

© 2003 - 2012 Cobra Language LLC. All rights reserved.

