

## Table of Contents

[Design Principles](#)

[Native Features](#)

[Suitable for Chess Programming](#)

[D Engines](#)

[Publications](#)

[Forum Posts](#)

[External Links](#)

[References](#)

[What links here?](#)

[Home](#) \* [Programming](#) \* [Languages](#) \* **D-Programming Language**

**D** is a programming language developed by [Walter Bright](#) from [Digital Mars](#) <sup>[1]</sup> <sup>[2]</sup>. The first stable version was released 2007. Walter Bright designed many other compilers as the first native compiler, the [Zortech C++](#) compiler, the [Symantec](#) C++ compiler, and the Digital Mars C++ Compiler and many [C](#) compilers (as the [Datalight](#) C compiler). D is in the Top 30 of the [TIOBE index](#) <sup>[3]</sup>.

## Design Principles

The design principles have been to write a compiler for a language that:

- is **fast** without any compromise
- has a **clean syntax** free of the [C++](#) quirks
- has all **modern concepts** of [interpreted languages](#). In fact D has some [Ruby](#) + [Python](#) elements that make **programming fun**.
- is **easy** to learn for [C](#) and C++ programmers

## Native Features

D might be a competitive choice for chess programming because:

- The performance / **speed** is comparable to C / C++
- It has **ulong** (8 bytes) and is very data-type rich

## Suitable for Chess Programming

Although it looks like D has tons of advantages over C/C++ in terms if a modern language, two problems might be worth to know

- It has been reported that D is not as good in bit fiddling as C / C++ if you are using chess magic
- It is unknown if the garbage collector has any impact on predictable results

D has the following native features:

- Easy string handling
- Static and dynamic [arrays](#) + [hashes](#)
- [Overloading](#), default parameters, varargs
- Nested functions (as Macros)
- Classes and inheritance and Modules/Mixins and templates
- Advanced testing features as contract programming and class invariants
- [Thread](#) and [fiber](#) support (actors in work)
- [Exception handling](#)
- Automatic or explicit [garbage collection](#): use internal GC or [free / malloc](#) as you like.
- [Closures](#)
- Direct [C](#) access and normal debugger support
- [Contract programming](#)
- Subset of constructs for easy [functional programming](#)
- and much more...

## D Engines

Dynamic list with [tag](#) 'engine'. Engines (at least some versions) written in D:

1. [Amoeba](#)
2. [Dumb](#)
3. [Shield](#)

## Publications

- [Andrej Alexandrescu \(2010\). The D Programming Language. Addison-Wesley Professional. ISBN 978-0321635365](#)

## Forum Posts

- [digitalmars.D - gdc and the Computer Language Benchmarks Game](#) by [Guido Schimmels](#), [digitalmars.D](#), June 28, 2007
- [Re: c vs c++](#) by [Brian Haskin](#), [CCC](#), July 14, 2011
- [programming languages - What does C++ do better than D? - Programmers - Stack Exchange](#)

## External Links

- [D Homepage](#)
- [D from Wikipedia](#)
- [Wiki4D: FrontPage](#)
- [Learn D Programming](#)
- [TutorialFundamentals - tutorials - dsource.org](#)
- [Language Description](#)
- [goshawk / gdc / wiki / Home — Bitbucket](#)

## References

1. [^ The D Programming Language - Walter Bright Interview](#) by [Huw Collingbourne](#) and [Dermot Hogan](#), [Bitwise Magazine](#)
2. [^ The A-Z of Programming Languages: D - Walter Bright talks about D and his desire to improve on systems programming language, Computerworld](#)
3. [^ TIOBE Software: Tiobe Index](#)

## What links here?

Page	Date Edited
<a href="#">Amoeba</a>	Mar 19, 2018
<a href="#">Array</a>	Dec 1, 2016
<a href="#">Assembly</a>	Sep 3, 2017
<a href="#">Brian Haskin</a>	Aug 2, 2011
<a href="#">D-Proglanguage</a>	Nov 28, 2016
<a href="#">Dumb</a>	Mar 10, 2018
<a href="#">Engines</a>	Mar 10, 2018
<a href="#">Languages</a>	Nov 26, 2017
<a href="#">Pseudorandom number generator</a>	May 11, 2017
<a href="#">Richard Delorme</a>	Sep 30, 2017
<a href="#">Shield</a>	May 17, 2017
<a href="#">Stefan Edlich</a>	Dec 11, 2017

[Up one Level](#)