

Table of Contents

[Data](#)

[Data Types](#)

[Primitive Data Types](#)

[Pointer](#)

[Array](#)

[Struct](#)

[Bitfield](#)

[Union](#)

[Variables](#)

[Globals](#)

[On the Stack](#)

[On the Heap](#)

[Register](#)

[Instructions](#)

[Operations](#)

[Arithmetical](#)

[Bitwise boolean](#)

[Relational](#)

[Logical](#)

[Functions](#)

[Expressions](#)

[Control Flow](#)

[Goto](#)

[If else](#)

[Switch case](#)

[Function Pointer](#)

[For](#)

[While](#)

[Do while](#)

[Preprocessor](#)

[Portabilty](#)

[Twos' Complement](#)

[sizeof](#)

[shift](#)

[Endianness](#)

[Libraries](#)

[C and C++ Compiler](#)

[Books](#)

[Publications](#)

[Forum Posts](#)

[1999](#)

[2000 ...](#)

[2005 ...](#)

[2010 ...](#)

[2015 ...](#)

[External Links](#)

[References](#)

[Home](#) * [Programming](#) * [Languages](#) * **C**

C is a pragmatical, general purpose, [block structured](#), [procedural](#), [imperative programming language](#). C was developed in **1972** by [Dennis Ritchie](#) at the [Bell Laboratories](#). It was first intended as a [system](#) programming language for the [Unix](#) operating system, but has spread to many other platforms and application programming as well. C and its derivations are likely the most often used languages so far for computer chess programming.

Due to explicit pointers (address of a variable or function), C can be considered as a high level assembly language, but has some weak spots in its initial design and implementation, which are addressed in

- [ANSI C from Wikipedia](#)
- [C99 from Wikipedia](#)
- [C11 \(C standard revision\) from Wikipedia](#)
- [Clang from Wikipedia](#) ^[1]
- [C++](#)

Data

Data Types

Primitive Data Types

To be aware of the scalar 64-bit origin of [bitboards](#) in computer chess, we use so far a type defined unsigned integer U64 in our C and [C++](#) source snippets. The macro C64 will append a suffix to 64-bit constants as required by some compilers:

```
typedef unsigned __int64 U64;    // for the old microsoft compilers
typedef unsigned long long  U64; // supported by MSC 13.00+ and C99
#define C64(constantU64) constantU64##ULL
```

Pointer

Array

Struct

A structure in C refers to [Object composition](#) to encapsulate related scalar datatypes inside one structured

item. The size of the structure is the sum of its element sizes. To access the structure elements the dot-operator separates the element from the variable or reference. Pointers require arrow operator.

```
struct MOVE
{
    char from;
    char to;
};

...
MOVE m, a, *b;
m.from = square;
...
if ( a.from == b->to )
```

Bitfield

So called [Bitfields](#) might be implemented as structure where integer members are declared with explicit bit length specifier from 1 .. 31. However due to portability issues of various C-compilers and platforms concerning bit ordering, padding and eventually the sign, most programmer rely on explicit bitfields to composite and extract sub-items by shift and masks, i.e. in [encoding moves](#).

Union

Variables

Variables are either stored in various memory areas or kept inside processor registers.

Globals

On the Stack

On the Heap

Register

Instructions

Operations

Arithmetical

Bitwise boolean

Relational

Logical

Functions

- [C mathematical functions from Wikipedia](#)

Expressions

Control Flow

- [Control flow from Wikipedia](#)

Goto

- [Goto from Wikipedia](#)
- [Considered harmful from Wikipedia](#)^{[2] [3] [4] [5] [6] [7]}
- [Using gotos](#) by [Steve McConnell](#)^[8]

If else

- [Conditional \(computer programming\) from Wikipedia](#)

Switch case

- [Switch statement from Wikipedia](#)

Function Pointer

- [Function pointer from Wikipedia](#)

For

- [For loop from Wikipedia](#)

While

- [While loop from Wikipedia](#)

Do while

- [Do while loop from Wikipedia](#)
- [The amazing Duff's Device](#) by [Tom Duff](#)
- [Duff's Device from Wikipedia](#)

Preprocessor

- [C preprocessor from Wikipedia](#)

Portabilty

Twos' Complement

sizeof

shift

Endianness

Main article [Endianness](#)

Libraries

- [C standard library](#) (libc)

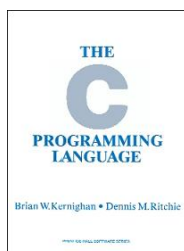
C and C++ Compiler

A C-Compiler is used to translate the source program, usually ascii-text files with the extension .C, to so called object files, containing machine instructions. A [linker](#) binds all the object files together with libraries containing external functions (and data) to build an executable program.

- [Microsoft Visual C++](#)
- [Intel C++](#)
- [GNU C Compiler](#)

Books

- [Brian W. Kernighan](#), [Dennis M. Ritchie](#) (1978, 1988). [The C Programming Language](#). First Edition ISBN 0-13-110163-3, Second Edition ISBN 0-13-110362-8



- [Andrew Koenig](#) (1989). [C Traps and Pitfalls](#). Addison-Wesley, ISBN 0-201-17928-8, [pdf preprint](#)
- [Patrick Winston](#) (1994). [On To C](#). ISBN-13: 978-0201580426
- [Andrew Appel](#), [Maia Ginsburg](#) (1998). [Modern Compiler Implementation in C](#). Cambridge University Press

Publications

- [Robert C. Seacord](#) (2010). *Dangerous Optimizations and the Loss of Causality*. CS 15-392 © 2010 Carnegie Mellon University, [slides as pdf](#)
- [Xi Wang](#), [Haogang Chen](#), [Alvin Cheung](#), [Zhihao Jia](#), [Nickolai Zeldovich](#), [M. Frans Kaashoek](#) (2012). *Undefined Behavior: What Happened to My Code?* [pdf](#)
- [Will Dietz](#), [Peng Li](#), [John Regehr](#), [Vikram Adve](#) (2012). *Understanding Integer Overflow in C/C++*. [pdf](#)

Forum Posts

1999

- [C or C++ for chess programming: speed](#) by [Marc-Philippe Huget](#), [CCC](#), October 20, 1999

2000 ...

- [One \(silly\) question about "C"](#) by Antonio Senatore, [CCC](#), February 05, 2002

2005 ...

- [Re: chess engines written in C](#) by [Dann Corbit](#), [CCC](#), January 13, 2005
- [ansi-C question](#) by [Vincent Diepeveen](#), [CCC](#), June 08, 2008
- [setjmp\(\) - another one](#) by [Chris Whittington](#), [CCC](#), August 27, 2008
- [kbhit\(\) taking huge CPU??](#) by [John Merlino](#), [CCC](#), April 01, 2009 » [Thread](#)
- [Critic: Pascal vs C](#) by [Richard Vida](#), [CCC](#), August 27, 2009 » [Pascal](#)

2010 ...

- [MSVC calloc question](#) by [Harm Geert Muller](#), [CCC](#), March 17, 2011
- [My experience with Linux/GCC](#) by [Richard Vida](#), [CCC](#), March 23, 2011 » [Linux](#)

- [a cautionary tale about simple-looking macros](#) by [Wylie Garvin](#), [CCC](#), July 03, 2011
- [c or c++ ?](#) by [ethan ara](#), [CCC](#), July 10, 2011
- [VisualStudio - fastcall instead of cdecl?](#) by [Martin Sedlak](#), [CCC](#), June 18, 2012
- [C vs ASM](#) by [Ed Schröder](#), [CCC](#), March 05, 2013 » [Assembly](#)
- [Re: goto thread \(split\)](#) by [Steven Edwards](#), [CCC](#), August 01, 2013 » [Iterative Search](#), [Symbolic](#)
- [A note for C programmers](#) by [Robert Hyatt](#), [CCC](#), November 23, 2013
- [Re: A note for C programmers](#) by [Rein Halbersma](#), [CCC](#), November 28, 2013
- [A note on strcpy](#) by [User923005](#), [OpenChess Forum](#), November 26, 2013
- [strcpy\(\) revisited](#) by [Robert Hyatt](#), [CCC](#), December 08, 2013

2015 ...

- [Using more than 1 thread in C beginner question](#) by [Uri Blass](#), [CCC](#), January 11, 2016 » [Thread](#)
- [C programming style question](#) by [Michael Sherwin](#), [CCC](#), January 19, 2016
- [Crafty c questions](#) by [J. Wesley Cleveland](#), [CCC](#), March 10, 2016 » [Crafty](#)
- [I'm not very happy with the do { } while\(\) statement in C](#) by [Michael Sherwin](#), [CCC](#), February 18, 2018

External Links

- [C from Wikipedia](#)
- [The C Book - Table of Contents](#), an online version of the popular introduction and reference on the ANSI Standard C programming language
- [C utilities](#) by [Gijsbert Wiesenekker](#)
- [Programming Bits](#) by [Paul Hsieh](#)
- [comp.lang.c](#) Discussion about C
- [comp.lang.c Frequently Asked Questions](#)
- [Chess Engine In C - YouTube](#) Videos by [BlueFeverSoft](#) » [Vice](#)
- [How to C in 2016](#)

References

1. [^ LLVM from Wikipedia](#)
2. [^ Edsger Dijkstra \(1968\)](#). *Go To Statement Considered Harmful*. [Communications of the ACM](#), Vol. 11, No. 3, [pdf](#)
3. [^ William A. Wulf \(1971\)](#). *Programming Without the GOTO*. [IFIP](#), Ljubljana, Yugoslavia, August 1971
4. [^ William A. Wulf \(1972\)](#). *A Case Against the GOTO*. Proceedings of the [ACM](#) National Conference, Boston, August 197
5. [^ Donald Knuth \(1974\)](#). *Structured Programming with go to Statements*. [ACM Computing Surveys](#), Vol. 6, No. 4, [pdf](#)
6. [^ Ward Douglas Maurer \(1996\)](#). *Attitudes toward the go-to statement (or, hydrogen considered harmful)*. [Computers & Education](#), Vol. 26, No. 4
7. [^ Coding Horror: I'd Consider That Harmful, Too](#) by [Jeff Atwood](#), October 25, 2007

8. [^ Steve McConnell \(1993\). Code Complete: A Practical Handbook of Software Construction. Microsoft Press](#)

[Up one Level](#)