C Programming: A Modern Approach

*K. N. King*

Source code available from: http://knking.com/books/c2/

# The C Language

C is a by-product of the UNIX operating system, developed as a higher-level language for use instead of UNIX’s native assembly language. The first iteration was named B but renamed once practical use led to a number of changes that significantly diverged the language from its original form. Eventually, UNIX was entirely written in C.

C is an influential language, and many commonly used languages are based on it:

* **C++** includes all features of C, but adds classes and OOP
* **Java** based on C++
* **C#** Derived from C++ and Java
* **Perl** originally a simple scripting language, eventually adopted much of the features of C

In learning C, a deeper understanding of derivative languages can be gained.

## Core features of C

* **Low level language:** access to machine-level concepts such as bytes and addresses.
* **Fast:** developed for system-level work, efficient for it.
* **Small:** a limited set of simple features, assisted by large function libraries.
* **Permissive:** allows more latitude to the programmer.

## Strengths of C

* **Efficient:** replaced assembly language for OS development historically. Has been fundamentally designed for efficiency and low memory usage.
* **Portability:** coherent across many platforms. A prime candidate for applications that must run on a desktop and a cluster.
* **Power:** many data types and operators mean a lot can be accomplished succinctly.
* **Flexibility:** Used for systems, embedded systems, data processing, etc. Allows things other languages might not, like adding a character to an integer or float. (Note: this gives the programmer more power, but allows them to make mistakes in ways other languages might not).
* **Standard library:** covers many commonly used functions.
* **UNIX integration:** some UNIX tools even assume good C knowledge.

## Weaknesses of C

* **Error-prone:** the programmer is given power to make mistakes the compiler might not recognise. Many might-be compiler errors manifest as run-time errors.
* **Poor readability:** due to its simplicity, code required in a C programme might be covered by a simple function in another language.
* **Difficult to modify:** C lacks classes and packages that support compartmentalisation of a programme available in other languages. This often makes programmes, once written, difficult to modify.

# C Fundamentals

## Running a Programme

To get a simple C programme to run, a few basic steps must be made:

* Creation of the **source code** with a .c file extension.
* The code is given to the **preprocessor**, which executes all lines beginning with # (known as directives). The preprocessor acts like an editor, adding or modifying the programme. E.g.: the most common directive, #include, adds useful subfunctions from libraries.
* **Compilation** translates the human-readable source code into machine-readable instructions (object code). It cannot yet run, until…
* **Linking** combines the object code to any additional code required to run, such as library functions.

## Operators

|  |  |
| --- | --- |
| + (unary) | Denotes positive value |
| - (unary) | Denotes negative value |
| + (binary) | Addition |
| - (binary) | Subtraction |
| \* | Multiplication |
| / | Division |
| % | Modulo, remainder |
| = | Assignment |
| += | Compound assignment for increment (i += 2 is i = i + 2) |
| i++ | Increment i by one (immediately) |
| ++i | Increment i by one (later, before next statement) |
| i-- | Decrement i by one (immediately) |
| --i | Decrement i by one (later, before next statement) |

Note there is no power operator in C. Instead, simply “manually” perform the power (e.g.: i^3 = i\*i\*i) or where that is unwieldly or the power is not an integer, use the pow() function, defined in the math.h header.

# Glossary of Terms

|  |  |
| --- | --- |
| Directive | Commands sent to the pre-processor before compilation. By definition, always begin with a #, always one line long, and not terminated by a semi-colon. Examples include #include and #define. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |