C: A Beginner's Guide

Joshua Weinstein

2018

Contents

0.1 Preface

This book is intended to serve as a tool to learn how to program in the C programming language. The book is targeted at those who have little to no experience in programming or computer science as a whole. Even if someone hasn't learned any prior programming languages, this book can serve as a first learning experience. This book discusses in depth the low level components of the C language, like memory, addresses, the stack, the heap, and more. The book introduces programming and C from the ground up, simplifying concepts often claimed as difficult or complex.

This books also introduces computer science concepts, like functions, types, data structures and algorithms, in the context of the C language. The purpose of this is to teach the fundamentals of programming from a simple, backbone starting point, and work up to more "magic" levels of abstraction, such as those seen in the Python language. This approach exposes students to a mechanical and complex understanding of programming. Such insight is exceptionally useful when optimizing and writing high performance software.

The book pays careful attention to the features of C which are often described as confusing, such as the *void* pointer, the functions *malloc* and *free*, and much more.

Chapter 1

Introduction

C is a low-level programming language designed for high portability and fast performance. C first originated in the late 1970's when Brian Kernighan and Dennis Ritchie wrote the book, *The C Programming Language*. C has tremendously influenced the world of programming, leading to the development of C++, an object oriented version of C, and the syntax of many other languages, like Java. C is still widely used today, in many applications such as databases, kernels, drivers, and much more. One of the most popular programming languages, Python, runs on an interpreter written in C.

This chapter will introduce the very fundamental ideas behind programming, and work as a primer for learning actual C concepts. Ideas such as addressing, memory, the stack, the heap, variables, printing, files will be discussed with friendly examples. If you have no or little experience in programming, it is highly recommended you start with this chapter. Otherwise, skip ahead to the next chapter.

1.1 Programs and Computing

In the most basic sense, a computer is a machine which can complete tasks by running instructions commonly known as "code". A computer cannot read or understand human languages, like English. A computer only understands instructions in a format known as *binary* code. Binary code is composed entirely of one's and zeros. 00000001 is an example of binary, which actually translates to the number, 1.

A programming language is a human readable language which is *compiled* into binary code.