Noel Kalicharan
Christian Posthoff (Hrsg.)

C Programming for Beginners

SHAKER
VERLAG

C
Programming
For Beginners

Noel Kalicharan

Senior Lecturer, Computer Science
The University of the West Indies
St. Augustine, Trinidad

First published September 2005
Republished August 2008
Revised 2015, 2020

© Noel Kalicharan, 2005, 2008, 2015, 2020
noelkali@gmail.com
noel.kalicharan@sta.uwi.edu

Berichte aus der Informatik

Dr. Noel Kalicharan

herausgegeben von Prof. Dr. Christian Posthoff

C Programming for Beginners

Shaker Verlag
Düren 2020

Bibliographic information published by the Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at http://dnb.d-nb.de.

Copyright Shaker Verlag 2020
All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of the publishers.

Printed in Germany.

ISBN 978-3-8440-7422-2
ISSN 0945-0807

Shaker Verlag GmbH • Am Langen Graben 15a • 52353 Düren
Phone: 0049/2421/99011-0 • Telefax: 0049/2421/99011-9
Internet: www.shaker.de • e-mail: info@shaker.de

Dedicated to

My esteemed colleague and friend

Professor Christian Posthoff

and his wife

Barbara

My children

Anushka Nikita

&

Saskia Anyara

My grand-daughter

Vaishnavi

Preface

This book attempts to teach computer programming to the complete beginner using the C language. As such, it assumes you have no knowledge whatsoever about programming. And if you are worried that you are not good at high-school mathematics, don't be. It is a myth that you must be good at mathematics to learn programming. In this book, a knowledge of primary school mathematics is all that is required—basic addition, subtraction, multiplication, division, finding the percentage of some quantity, finding an average or the larger of two quantities.

Some of our most outstanding students over the last thirty years have been people with little mathematics background from all walks of life—politicians, civil servants, sports people, housewives, secretaries, clerical assistants, artists, musicians and teachers. On the other hand, we've had mathematical folks who didn't do as well as might be expected.

What *will* be an asset is the ability to think logically or to follow a logical argument. If you are good at presenting convincing arguments, you will probably be a good programmer. Even if you aren't, programming is the perfect vehicle for teaching logical thinking skills. You should learn programming for these skills even if you never intend to become a serious programmer.

The main goal of this book is to teach fundamental programming principles using C, one of the most widely used programming languages in the world today. C is considered a 'modern' language even though its roots date back to the 1970s. Originally, C was designed for writing 'systems' programs—things like operating systems, editors, compilers, assemblers and input/output utility programs. But, today, C is used for writing all kinds of applications programs as well— wordprocessing programs, spreadsheet programs, database management programs, accounting programs, games, educational software—the list is endless.

However, this book is more about teaching programming basics than it is about teaching C. We discuss only those features and statements in C that are necessary to achieve our goal. Once you learn the *principles* well, they can be applied to any language.

Chapter 1 gives an overview of the programming process. Chapter 2 describes the basic building blocks needed to write programs. Chapter 3 explains how to write programs with the simplest kind of logic—sequence logic. Chapter 4 shows how to write programs which can make decisions. Chapter 5 explains the notion of 'looping' and how to use this powerful programming idea to solve more interesting problems. Chapter 6 deals with the oft-neglected, but important, topic of working with characters. Chapter 7 introduces functions—the key concept needed for writing large programs. And Chapter 8 tackles the nemesis of many would-be programmers—array processing.

The first step in becoming a good programmer is learning the syntax rules of the programming language. This is the easy part and many people mistakenly believe that this makes them a programmer. They get carried away by the cosmetics—they learn the *features* of a language without learning how to use them to solve problems. Of course, you must learn *some* features. But it is far better to learn a few features and be able to use them to solve many problems rather than learn many features but can't use them to solve anything. For this reason, this book introduces a feature (an `if` statement, say) and then discusses many examples to illustrate how the feature can be used to solve different problems.

This book is intended for anyone who is learning programming for the first time, regardless of age or institution. The material has been taught successfully to students preparing for high-school examinations in Computer Studies or Information Technology, students at college, university and other tertiary-level institutions.

The presentation is based on the experience that many people have difficulty in learning programming. To try and overcome this, we use an approach which provides clear examples, detailed explanations of very basic concepts and numerous interesting problems (not just artificial exercises whose only use is to illustrate some language feature).

While computer programming is essentially a mental activity and you *can* learn a fair amount of programming from just *reading* the book, it is important that you "get your hands dirty" by writing and running programs. One of life's thrills is to write your first program and get it to run successfully on a computer. Don't miss out on it.

But do not stop there. The only way to learn programming well is to write programs to solve new problems. The end-of-chapter exercises are a very rich source of problems, a result of the author's more than 40 years in the teaching of programming.

Thank you for taking the time to read this book. I hope your venture into programming is a successful and enjoyable one.

Noel Kalicharan

Contents