Program construction

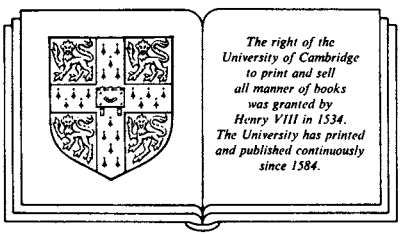
Also in this series

1 An Introduction to Logical Design of Digital Circuits
C. M. Reeves 1972
2 Information Representation and Manipulation in a Computer
E. S. Page and L. B. Wilson, Second Edition 1978
3 Computer Simulation of Continuous Systems
R. J. Ord-Smith and J. Stephenson 1975
4 Macro Processors
A. J. Cole, Second Edition 1981
5 An Introduction to the Uses of Computers
Murray Laver 1976
6 Computing Systems Hardware
M. Wells 1976
7 An Introduction to the Study of Programming Languages
D. W. Barron 1977
8 ALGOL 68 – A first and second course
A. D. McGettrick 1978
9 An Introduction to Computational Combinatorics
E. S. Page and L. B. Wilson 1979
10 Computers and Social Change
Murray Laver 1980
11 The Definition of Programming Languages
A. D. McGettrick 1980
12 Programming via Pascal
J. S. Rohl and H. J. Barrett 1980
13 Program Verification using Ada
A. D. McGettrick 1982
14 Simulation Techniques for Discrete Event Systems
I. Mitrani 1982
15 Information Representation and Manipulation using Pascal
E. S. Page and L. B. Wilson 1983
16 Writing Pascal Programs
J. S. Rohl 1983
17 An Introduction to APL
S. Pommier 1983
18 Computer Mathematics
D. J. Cooke and H. E. Bez 1984
19 Recursion via Pascal
J. S. Rohl 1984
20 Text Processing
A. Colin Day 1984
21 Introduction to Computer Systems
Brian Molinari 1985

22 Cambridge Computer Science Texts

Program construction

R. G. Stone and D. J. Cooke
Department of Computer Studies, Loughborough University of Technology



Cambridge University Press
Cambridge
New York *Port Chester*
Melbourne *Sydney*

CAMBRIDGE UNIVERSITY PRESS
Cambridge, New York, Melbourne, Madrid, Cape Town, Singapore, São Paulo

Cambridge University Press
The Edinburgh Building, Cambridge CB2 2RU, UK

Published in the United States of America by Cambridge University Press, New York

www.cambridge.org
Information on this title: www.cambridge.org/9780521268233

First published 1987
Reprinted 1988, 1990

A catalogue record for this publication is available from the British Library

Library of Congress Cataloguing in Publication data

Stone, R. G, 1950–
Program construction.
(Cambridge computer science texts; 22)
Bibliography: p.
Includes index.
1. Electronic digital computers – Programming.
I. Cooke, D. J. (Derek John), 1947–
II. Title. III. Series.
QA76.6S767 1987 005.1 86-12954

ISBN-13 978-0-521-26823-3 hardback
ISBN-10 0-521-26823-0 hardback

ISBN-13 978-0-521-31883-9 paperback
ISBN-10 0-521-31883-1 paperback

Transferred to digital printing 2006

Contents

Preface

This text promotes the disciplined construction of procedural programs from formal specifications. As such it can be used in conjunction with any of the more conventional programming texts which teach a mixture of 'coding' in a specific language and *ad hoc* algorithm design.

The awareness of the need for a more methodical approach to program construction is epitomised by the use of phrases such as 'software engineering', 'mathematical theory of programming', and 'science of programming'. The hitherto all-too-familiar practices of 'designing' a program 'as you write it' and 'patching' wrong programs being more appropriate to a cottage industry rather than a key activity in the current technological revolution.

The cost of producing hardware is decreasing while the production of software (programs) is becoming more expensive by the day. The complexity and importance of programs is also growing phenomenally, so much so that the high cost of producing them can only be justified when they are reliable and do what they are supposed to do – when they are correct.

No methodology can exist by which we can produce a program to perform an arbitrary task. Consequently that is **not** the aim of the book. What we **shall** do is to show how, by using a Program Design Language and templates for your chosen target language, you can develop programs from certain forms of specification.

Although programming is essentially a practical activity, the degree of formality adopted throughout the development process means that sufficient information is available to enable correctness proofs to be investigated if and when required. Moreover, the structured programming forms used throughout the text are all supported by verification rules derived from their total correctness proofs – the notion of correctness never being far from our thoughts.

The material presented has grown out of courses presented to first year Computer Science undergraduates, to 'conversion' M.Sc. students and in

industrial short courses in software engineering, and as such has been under development since 1980.

During the evolution of the teaching material included herein we have been influenced by many sources. Of particular note is the work of Cliff Jones (now at Manchester University but previously at Oxford PRG and various IBM research establishments), on Specification; and the work of John Darlington (now at Imperial College, London and previously at Edinburgh University), on Program Transformations. At a more tangible level we wish to record our thanks to Terry Butland of UKAEE Winfrith and Morry van Ments of Loughborough's Centre for Extension Studies for their help in organising our industrial courses, to our colleagues, Mike Woodward and Dave Gittins, who helped modify earlier drafts of the text. to Jacqui Bonsor, Carole Hill and Deborah Harrison who produced the bulk of the typescript and to Ernest Kirkwood of Cambridge University Press for his encouragement and patience.

R. G. Stone
D. J. Cooke
Loughborough, 1985