

## How to Write Good Programs

Learning to program isn't just learning the details of a programming language: to become a good programmer you have to become expert at debugging, testing, writing clear code and generally unsticking yourself when you get stuck, while to do well in a programming course you have to learn to score highly in coursework and exams.

Featuring tips, stories and explanations of key terms, this book teaches these skills explicitly. Examples in Python, Java and Haskell are included, helping you to gain transferable programming skills whichever language you are learning. Intended for students in Higher or Further Education studying early programming courses, it will help you succeed in, and get the most out of, your course and support you in developing the software engineering habits that lead to good programs.

**Perdita Stevens** is a professor at the University of Edinburgh and has taught programming and software engineering in many languages to students ranging from first-year undergraduates to established professional software developers. She previously co-authored *Using UML: Software Engineering with Objects and Components* (1998) and received a 10-year Most Influential Paper award and a Best Paper award at the MODELS conference in 2017 for her work on bidirectional transformations.

CAMBRIDGE

Cambridge University Press  
978-1-108-78987-5 – How to Write Good Programs

Perdita Stevens

Frontmatter

[More Information](#)

---

CAMBRIDGE

Cambridge University Press  
978-1-108-78987-5 – How to Write Good Programs  
Perdita Stevens  
Frontmatter  
[More Information](#)

---

# How to Write Good Programs

## A Guide for Students

PERDITA STEVENS



Cambridge University Press  
978-1-108-78987-5 — How to Write Good Programs  
Perdita Stevens  
Frontmatter  
[More Information](#)

## CAMBRIDGE UNIVERSITY PRESS

University Printing House, Cambridge CB2 8BS, United Kingdom  
One Liberty Plaza, 20th Floor, New York, NY 10006, USA  
477 Williamstown Road, Port Melbourne, VIC 3207, Australia  
314–321, 3rd Floor, Plot 3, Splendor Forum, Jasola District Centre,  
New Delhi – 110025, India  
79 Anson Road, #06–04/06, Singapore 079906

Cambridge University Press is part of the University of Cambridge.

It furthers the University's mission by disseminating knowledge in the pursuit of education, learning, and research at the highest international levels of excellence.

[www.cambridge.org](http://www.cambridge.org)

Information on this title: [www.cambridge.org/9781108789875](http://www.cambridge.org/9781108789875)

DOI: 10.1017/9781108804783

© Perdita Stevens 2020

This publication is in copyright. Subject to statutory exception and to the provisions of relevant collective licensing agreements, no reproduction of any part may take place without the written permission of Cambridge University Press.

First published 2020

Printed in the United Kingdom by TJ International Ltd. Padstow Cornwall

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

*Library of Congress Cataloging-in-Publication Data*

Names: Stevens, Perdita, author.

Title: How to write good programs : a guide for students / Perdita Stevens.

Description: New York : Cambridge University Press, 2020. |

Includes bibliographical references and index.

Identifiers: LCCN 2020003584 (print) | LCCN 2020003585 (ebook) |

ISBN 9781108804783 (epub) | ISBN 9781108789875 (hardback)

Subjects: LCSH: Computer programming—Textbooks.

Classification: LCC QA76.73.P98 (ebook) |

LCC QA76.73.P98 S74 2020 (print) | DDC 005.13—dc23

ISBN 978-1-108-78987-5 Paperback

Cambridge University Press has no responsibility for the persistence or accuracy of URLs for external or third-party internet websites referred to in this publication and does not guarantee that any content on such websites is, or will remain, accurate or appropriate.

## Contents

---

<b>1</b>	Introduction	<i>page</i> 1
1.1	Who Is This Book For?	3
1.2	About the Boxes	4
1.3	Structure of This Book	7
1.4	Acknowledgements	7
<b>2</b>	What Are Good Programs?	9
2.1	Ethics	10
<b>3</b>	How to Get Started	12
3.1	What Is a Program Anyway?	12
3.2	What Do You Need?	13
3.3	Understanding What You Have to Do	18
3.4	Writing Your Program	20
3.5	What to Do If You Get Confused	29
<b>4</b>	How to Understand Your Language	33
4.1	Compilation or Interpretation	34
4.2	Types	37
4.3	Structure	41
4.4	History, Community and Motivation	44
4.5	Paradigms	45
<b>5</b>	How to Use the Best Tools	48
5.1	Using the Most Basic Tools	49
5.2	What Is an IDE?	50
5.3	Looking Forward	53

<b>6</b>	How to Make Sure You Don't Lose Your Program	55
6.1	Immediate Recovery: Undo	55
6.2	Basic Disaster Recovery: Files	56
6.3	Avoiding Disaster: Saving Versions	58
6.4	Automating the Process: Using a Version Control System	59
6.5	Managing Code that Is Not in Use	61
6.6	Backups and the Cloud	65
<b>7</b>	How to Test Your Program	68
7.1	Manual Testing	69
7.2	Basic Automated Testing	71
7.3	Proper Automated Testing	75
7.4	What Tests Should You Have?	77
7.5	When Should You Write Tests?	78
7.6	Property-Based Testing	79
<b>8</b>	How to Make Your Program Clear	83
8.1	How Will Writing Clear Code Help You?	83
8.2	Comments	85
8.3	Names	89
8.4	Layout and Whitespace	92
8.5	Structure and Idiom	96
<b>9</b>	How to Debug Your Program	101
9.1	When You Can't Run Your Program Yet	103
9.2	When Your Program Runs but Behaves Wrongly	109
9.3	Cardboard Debugging	120
9.4	If All Else Fails	121
9.5	Removing the Bug	122
9.6	After Removing the Bug	124
<b>10</b>	How to Improve Your Program	134
10.1	Maintainability	134
10.2	Efficiency	145

	Contents	vii
	10.3 Refactoring	149
	10.4 Improving Your Skills	153
<b>11</b>	How to Get Help (without Cheating)	156
	11.1 Solving a General Problem	157
	11.2 Solving a More Specific Problem	160
	11.3 How to Cope When Your Teacher Is Confusing You	168
<b>12</b>	How to Score Well in Coursework	170
	12.1 Seven Golden Rules	170
	12.2 Lab Exercises	172
	12.3 Individual Projects	173
	12.4 Team Working	174
	12.5 Demonstrations	175
	12.6 Reflective Writing	177
<b>13</b>	How to Score Well in a Programming Exam	179
	13.1 Preparing for the Exam	179
	13.2 In the Exam	182
	13.3 Specific Points for Paper Exams	183
	13.4 Specific Points for Computer-Based Exams	184
	13.5 What about Multiple Choice Exams?	185
<b>14</b>	How to Choose a Programming Language	186
	14.1 Questions to Consider	186
	14.2 A Few Languages You May Encounter	189
	14.3 The Changing Landscape of Languages	192
<b>15</b>	How to Go Beyond This Book	194
	15.1 Doing More Programming	194
	15.2 Specific Programming Languages	195
	15.3 Programming Generally	195

Cambridge University Press  
978-1-108-78987-5 – How to Write Good Programs  
Perdita Stevens  
Frontmatter  
[More Information](#)

---

viii

## Contents

15.4 Software Engineering	196
15.5 Programming Language Theory	198
Bibliography	200
Index	202