

Cambridge University Press

978-0-521-83425-4 - Talking with Computers: Explorations in the Science and Technology of Computing

Thomas Dean

Frontmatter

[More information](#)

TALKING WITH COMPUTERS

In this lively series of essays, Tom Dean explores interesting fundamental topics in computer science with the aim of showing how computers and computer programs work and how the various subfields of computer science are connected. Along the way, he conveys his fascination with computers and enthusiasm for working in a field that has changed almost every aspect of our daily lives.

The essays touch on a wide range of topics, from digital logic and machine language to artificial intelligence and searching the World Wide Web, considering such questions as

- How can a computer learn to recognize junk email?
- What happens when you click on a link in a browser?
- How can you program a robot to do two things at once?
- Are there limits on what computers can do?

The author invites readers to experiment with short programs written in several languages. Through these interactions he grounds the models and metaphors of computer science and makes the underlying computational ideas more concrete. The accompanying Web site www.cs.brown.edu/~tld/talk provides easy access to code fragments from the book, tips on finding and installing software, and links to online resources, exercises, and sample lectures.

Thomas Dean is Professor of Computer Science at Brown University. Dean is a Fellow of AAI and a former member of the IJCAI Inc. Board of Trustees. He has served on the Executive Council of AAI and the CRA Board of Directors. In addition, Dean has acted as program chair to a number of international conferences in Artificial Intelligence, was recipient of an NSF Presidential Young Investigator Award, and has co-authored two popular books. His research interests include automated planning, machine learning, and robotics.

Cambridge University Press

978-0-521-83425-4 - Talking with Computers: Explorations in the Science and Technology of Computing

Thomas Dean

Frontmatter

[More information](#)

Cambridge University Press

978-0-521-83425-4 - Talking with Computers: Explorations in the Science and Technology of Computing

Thomas Dean

Frontmatter

[More information](#)

TALKING WITH COMPUTERS

Explorations in the Science and
Technology of Computing

THOMAS DEAN

Brown University, Providence, Rhode Island



Cambridge University Press

978-0-521-83425-4 - Talking with Computers: Explorations in the Science and Technology of Computing

Thomas Dean

Frontmatter

[More information](#)

CAMBRIDGE UNIVERSITY PRESS

Cambridge, New York, Melbourne, Madrid, Cape Town,
Singapore, São Paulo, Delhi, Mexico City

Cambridge University Press

The Edinburgh Building, Cambridge CB2 8RU, UK

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

www.cambridge.org

Information on this title: www.cambridge.org/9780521834254

© Thomas Dean 2004

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 2004

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

Library of Congress Cataloguing in Publication Data

Dean, Thomas L., 1950–

Talking with computers : explorations in the science and technology of computing / Thomas Dean.

p. cm.

Includes bibliographical references and index.

ISBN 0-521-83425-2 – ISBN 0-521-54204-9 (pb.)

1. Computer Science. 2. Computer programs. I. Title

QA76.D3333328 2004

946-dc22 2003065531

ISBN 978-0-521-83425-4 Hardback

ISBN 978-0-521-54204-3 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. Information regarding prices, travel
timetables, and other factual information given in this work is correct at
the time of first printing but Cambridge University Press does not guarantee
the accuracy of such information thereafter.

Contents

<i>Preface</i>	<i>page</i> ix
<i>Acknowledgments</i>	xiii
1 TALKING WITH COMPUTERS	1
1.1 Computers everywhere	2
1.2 Everyday magic	4
1.3 Hacking in mathematics	12
1.4 Programming in logic	13
1.5 Scheming in Lisp	17
2 THE SHELL GAME	22
2.1 Shell programming	24
2.2 Shell variables	29
2.3 Information passing	32
2.4 Asynchronous processes	37
3 KEEPING TRACK OF YOUR STUFF	41
3.1 Finding stuff	42
3.2 Organizing your stuff	47
3.3 Database management	51
4 DON'T SWEAT THE SYNTAX	57
4.1 Specifications and implementations	58
4.2 Syntactic variations across languages	63
4.3 Stylistic variations across implementations	67
4.4 Developing a facility for language	68

5 COMPUTATIONAL MUDDLES	70
5.1 Computational models	71
5.2 The substitution model	77
5.3 Syntax and style revisited	81
6 GETTING ORIENTED	85
6.1 Structuring large programs	86
6.2 Procedures that remember	87
6.3 Object-oriented programming	92
6.4 Programming with constraints	95
7 THANKS FOR SHARING	103
7.1 Code for the taking	105
7.2 Class conscious	109
7.3 It's just syntax	114
8 YOU'VE GOT (JUNK) EMAIL	121
8.1 Artificial intelligence	123
8.2 Machine learning	128
8.3 Learning with probabilities	133
8.4 Learning more about learning	138
9 MODERN ARCHITECTURE	140
9.1 Logic gates	141
9.2 The digital abstraction	144
9.3 Addition and multiplication	146
9.4 Computer memory	149
9.5 Machine language	152
10 DO ROBOTS SLEEP?	162
10.1 Stacks and subroutines	163
10.2 Managing tasks	168
10.3 Multithreaded robots	172
10.4 Allocating resources	178
10.5 Metaphorically speaking	184
11 UNDER THE HOOD	185
11.1 Client-server model	186
11.2 Acronym city	189
11.3 Alphabet soup	191
11.4 Smart milk cartons	193

CONTENTS

vii

12 ANALYZE THIS	196
12.1 Analyzing algorithms	197
12.2 Computational limitations	205
12.3 Theory that matters	210
13 FOREST FOR THE TREES	213
13.1 Graph theory	214
13.2 Graph algorithms	217
13.3 File systems as graphs	228
13.4 The web graph	230
13.5 Pianos and robots	232
14 SEARCHING THE WILD WEB	237
14.1 Spiders in the web	237
14.2 Measuring similarity	240
14.3 Measuring authority	248
14.4 Searching for exotic fruit	254
15 DARWIN'S DANGEROUS ALGORITHM	257
15.1 Competing hypotheses	258
15.2 Genetic algorithms	259
15.3 Survival of the fittest	263
16 AIN'T NOBODY HERE BUT US MACHINES	271
16.1 Machine intelligence	272
16.2 Other minds	277
16.3 Freedom to choose	281
16.4 Carrying on	285
<i>Bibliography</i>	289
<i>Index</i>	295

Cambridge University Press

978-0-521-83425-4 - Talking with Computers: Explorations in the Science and Technology of Computing

Thomas Dean

Frontmatter

[More information](#)

Preface

Computers are changing almost every aspect of our lives. They're changing how we relate to one another and even changing how we think of ourselves. The very idea that my brain is a biological computer that could be, in some fundamental mathematical sense, no more powerful than the laptop on which I'm typing these words is mind-boggling. The fact that I can program a computer to control a robot, play chess, or find a cure for disease is tremendously empowering.

This book is organized as a series of essays that explore interesting and fundamental topics in computer science with the aim of showing how computers and computer programs work and how the various aspects of computer science are connected. Along the way I hope to convey to you some of my fascination with computers and my enthusiasm for working in a field whose explosive growth is fueled in no small measure by the ability of computers to support collaboration and information sharing.

While not meant to be exhaustive, this book examines a wide range of topics, from digital logic and machine language to artificial intelligence and searching the World Wide Web. These topics are explored by interacting with programs and experimenting with short fragments of code while considering such questions as:

- How can a computer learn to recognize junk email?
- What happens when you click on a link in a browser?
- How can we program a robot to do two things at once?
- What can evolutionary biology teach us about computation?
- How can we find what we're looking for on the World Wide Web?

You can read this book without ever touching a computer, but I think you'll have a lot more fun and learn more if you follow along by running the code

fragments in the text. Think of each fragment as a thread in a complex pattern; I try to unravel the thread and explain how it relates to the larger pattern. Then, by making small changes to a code fragment, you can explore some of the nearby threads and test your understanding of the concepts.

Reading a book is different from attending lectures, but a book, like a lecture, adopts a style of teaching and assumes a style of learning. My style in this book is to start with examples and introduce general principles as they seem appropriate. For the most part, I'm hoping that the general principles will come peeking through the examples without my being too heavy-handed. The numerous code fragments invite you to experiment on your own.

I've included snippets of code from a variety of languages: C, Java, Perl and Scheme. I've tried hard to help you see past the differences in syntax to the basic underlying computations. Despite what you may have heard about introductory courses, computer science has little to do with arcane syntax; instead, it's all about thinking about problems computationally. Syntax is just a means to the end of communicating with computers, and here we focus first on what we want to say and only second on how to say it.

But it's no fun if you can't communicate with the computer, and so you'll meet lots of code and lots of syntax in this book. I don't, however, include a comprehensive introduction to programming. You'll need additional sources of information if your experiments deviate significantly from the programs presented here. Fortunately, much of what you need as supplementary material is available online.

Many of my examples involve communicating directly with the operating system by using interactive programs called shells. Shell programs are available for most modern operating systems including Unix, Linux, Solaris, and the Apple and Microsoft operating systems. Shells provide commands for accessing a great deal of online documentation. For example, `man` formats and displays the online manual pages for all the basic commands; `info` provides similar services with a somewhat nicer interface.

Of course, you've got to know where to look and how to call up all this wonderful information. The Web site for this book (www.cs.brown.edu/~tld/talk/) is a good place to start. It lists resources for setting up shells for different operating systems as well as pointers on installing the other programs mentioned in the book. The Web site also lists various online forums and news services that you can subscribe to and post your own questions and suggestions.

Despite all the legends about isolated hackers, programming is a primarily social activity. Programmers use, build on, and improve programs written by others.

Cambridge University Press

978-0-521-83425-4 - Talking with Computers: Explorations in the Science and Technology of Computing

Thomas Dean

Frontmatter

[More information](#)**PREFACE**

xi

Programmers learn by talking with other programmers and reading one another's code. Learning to program can be accelerated by learning with others. Even if you don't initially have close friends who share your programming interests, the Web makes it easy to meet other programmers at all skill levels. If you're shy you can start by reading newsgroups, but don't miss out on all the fun of sharing programs and interacting with other programmers face to face and online.

Cambridge University Press

978-0-521-83425-4 - Talking with Computers: Explorations in the Science and Technology of Computing

Thomas Dean

Frontmatter

[More information](#)

Acknowledgments

I've talked for years with students considering majoring in computer science who are curious what it's about. I really enjoy these conversations, and invariably as the student gets ready to leave I suggest some book that I think might extend our conversation. I've always wanted to name a book that explores a wide range of topics while remaining grounded in real and useful programs. I want students to experience firsthand the power of computing and gain some appreciation of its many applications. This book was inspired by those conversations and is dedicated to those students.

I couldn't have written this book without the encouragement and patient feedback of many friends and colleagues. I especially want to thank John Bazik, Roger Blumberg, Tom Doepfner, John Hughes, Philip Klein, Shriram Krishnamurthi, Martha Pollack, Bart Selman, Eli Upfal, and Joel Young. Lauren Cowles at Cambridge University Press helped refocus my initial draft and then provided concrete suggestions on how to improve every chapter. And there's not a paragraph that didn't benefit from Trina Avery's careful and insightful editing; Trina made the often tedious process of copyediting mostly interesting and frequently fun.

But most of all, I want to thank my students for their enthusiasm, their creativity, and their friendship. I wish I could list them all, but I especially want to thank Kalin Agrawal, Candace Batts, Nick Beaudrot, Sarah Bell, James Brock, Christine Davis, Michelle Engel, Erika Faires, Sam Hazlehurst, Kate Ho, Aron Holzman, Eliot Horowitz, Albert Huang, Danielle Karr, Roger Lederman, Katrina Ligett, Luke Ma, Gideon Mann, Paul Melnikow, Curran Nachbar, Bryant Ng, Brian O'Neil, Leah Pearlman, Ana-Maria Popescu, Brock Pytlik, Seema Ramchandani, Susannah Raub, Jennifer Rosenbaum, Brandon Roy, Maryam Saleh, Gloria Satgunam, Caitlyn Schmidt, Andrew Schulak, Tracy Schultz, Ravi Sitwala, Damien Suttle, Anisa Virji, Stacy Wong, and Thomas Wooldridge.