The Cambridge Handbook of Computing Education Research

This handbook describes the extent and shape of computing education research today. Over 50 leading researchers from academia and industry (including Google and Microsoft) have contributed chapters that together define and expand the evidence base.

The foundational chapters set the field in context, articulate expertise from key disciplines, and form a practical guide for new researchers. They address what can be learned empirically, methodologically, and theoretically from each area. The topic chapters explore issues that are of current interest, why they matter, and what is already known. They include discussion of motivational context, implications for practice, and open questions that might suggest avenues of future research.

The authors provide an authoritative introduction to the field that is essential reading for policy makers, as well as both new and established researchers.

SALLY A. FINCHER is Professor of Computing Education in the School of Computing at the University of Kent, UK, where she leads the Computing Education Research Group. She is also an Association for Computing Machinery Distinguished Scientist, a UK National Teaching Fellow, a Senior Fellow of the UK Higher Education Academy, and a Fellow of the Royal Society of Arts.

ANTHONY V. ROBINS is Professor of Computer Science at the University of Otago, New Zealand. He is also Associate Journal Editor of *Computer Science Education* and has co-organized multinational research studies. He has worked for the Ministry of Education, New Zealand, on new programming assessment standards and related instructional materials for secondary schools.

Published online by Cambridge University Press

Published online by Cambridge University Press

The Cambridge Handbook of Computing Education Research

Edited by

Sally A. Fincher
University of Kent

Anthony V. Robins
University of Otago

CAMBRIDGE
UNIVERSITY PRESS

Published online by Cambridge University Press

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
Information on this title: www.cambridge.org/9781108496735
DOI: 10.1017/9781108654555

© Cambridge University Press 2019

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 2019

Printed and bound in Great Britain by Clays Ltd, Elcograf S.p.A.

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

Library of Congress Cataloging-in-Publication Data
Names: Fincher, Sally, 1959– editor. | Robins, Anthony (Anthony V.) editor.
Title: The Cambridge handbook of computing education research /
edited by Sally A. Fincher, University of Kent, Anthony V. Robins,
University of Otago.
Other titles: Handbook of computing education research
Description: New York, NY: Cambridge University Press, 2019. |
Includes bibliographical references and index.
Identifiers: LCCN 2018039947 | ISBN 9781108496735 (hardback) |
ISBN 9781108721899 (paperback)
Subjects: LCSH: Computer science – Study and teaching – Handbooks, manuals, etc.
Classification: LCC QA76.27.C355 2019 | DDC 004.071–dc23
LC record available at https://lccn.loc.gov/2018039947

ISBN 978-1-108-49673-5 Hardback
ISBN 978-1-108-72189-9 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.

Published online by Cambridge University Press

The Cambridge Handbook of Computing Education Research

This handbook describes the extent and shape of computing education research today. Over 50 leading researchers from academia and industry (including Google and Microsoft) have contributed chapters that together define and expand the evidence base.

The foundational chapters set the field in context, articulate expertise from key disciplines, and form a practical guide for new researchers. They address what can be learned empirically, methodologically, and theoretically from each area. The topic chapters explore issues that are of current interest, why they matter, and what is already known. They include discussion of motivational context, implications for practice, and open questions that might suggest avenues of future research.

The authors provide an authoritative introduction to the field that is essential reading for policy makers, as well as both new and established researchers.

SALLY A. FINCHER is Professor of Computing Education in the School of Computing at the University of Kent, UK, where she leads the Computing Education Research Group. She is also an Association for Computing Machinery Distinguished Scientist, a UK National Teaching Fellow, a Senior Fellow of the UK Higher Education Academy, and a Fellow of the Royal Society of Arts.

ANTHONY V. ROBINS is Professor of Computer Science at the University of Otago, New Zealand. He is also Associate Journal Editor of *Computer Science Education* and has co-organized multinational research studies. He has worked for the Ministry of Education, New Zealand, on new programming assessment standards and related instructional materials for secondary schools.

Published online by Cambridge University Press

The Cambridge Handbook of Computing Education Research

Edited by
Sally A. Fincher
University of Kent
Anthony V. Robins
University of Otago

CAMBRIDGE
UNIVERSITY PRESS

Published online by Cambridge University Press

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
Information on this title: www.cambridge.org/9781108496735
DOI: 10.1017/9781108654555

© Cambridge University Press 2019

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 2019

Printed and bound in Great Britain by Clays Ltd, Elcograf S.p.A.

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

Library of Congress Cataloging-in-Publication Data
Names: Fincher, Sally, 1959– editor. | Robins, Anthony (Anthony V.) editor.
Title: The Cambridge handbook of computing education research /
edited by Sally A. Fincher, University of Kent, Anthony V. Robins,
University of Otago.
Other titles: Handbook of computing education research
Description: New York, NY: Cambridge University Press, 2019. |
Includes bibliographical references and index.
Identifiers: LCCN 2018039947 | ISBN 9781108496735 (hardback) |
ISBN 9781108721899 (paperback)
Subjects: LCSH: Computer science – Study and teaching – Handbooks, manuals, etc.
Classification: LCC QA76.27.C355 2019 | DDC 004.071–dc23
LC record available at https://lccn.loc.gov/2018039947

ISBN 978-1-108-49673-5 Hardback
ISBN 978-1-108-72189-9 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.

Published online by Cambridge University Press

The Cambridge Handbook of Computing Education Research

This handbook describes the extent and shape of computing education research today. Over 50 leading researchers from academia and industry (including Google and Microsoft) have contributed chapters that together define and expand the evidence base.

The foundational chapters set the field in context, articulate expertise from key disciplines, and form a practical guide for new researchers. They address what can be learned empirically, methodologically, and theoretically from each area. The topic chapters explore issues that are of current interest, why they matter, and what is already known. They include discussion of motivational context, implications for practice, and open questions that might suggest avenues of future research.

The authors provide an authoritative introduction to the field that is essential reading for policy makers, as well as both new and established researchers.

SALLY A. FINCHER is Professor of Computing Education in the School of Computing at the University of Kent, UK, where she leads the Computing Education Research Group. She is also an Association for Computing Machinery Distinguished Scientist, a UK National Teaching Fellow, a Senior Fellow of the UK Higher Education Academy, and a Fellow of the Royal Society of Arts.

ANTHONY V. ROBINS is Professor of Computer Science at the University of Otago, New Zealand. He is also Associate Journal Editor of *Computer Science Education* and has co-organized multinational research studies. He has worked for the Ministry of Education, New Zealand, on new programming assessment standards and related instructional materials for secondary schools.

Published online by Cambridge University Press

The Cambridge Handbook of Computing Education Research

Edited by

Sally A. Fincher
University of Kent

Anthony V. Robins
University of Otago

CAMBRIDGE
UNIVERSITY PRESS

Published online by Cambridge University Press

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
Information on this title: www.cambridge.org/9781108496735
DOI: 10.1017/9781108654555

© Cambridge University Press 2019

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 2019

Printed and bound in Great Britain by Clays Ltd, Elcograf S.p.A.

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

Library of Congress Cataloging-in-Publication Data
Names: Fincher, Sally, 1959– editor. | Robins, Anthony (Anthony V.) editor.
Title: The Cambridge handbook of computing education research /
edited by Sally A. Fincher, University of Kent, Anthony V. Robins,
University of Otago.
Other titles: Handbook of computing education research
Description: New York, NY: Cambridge University Press, 2019. |
Includes bibliographical references and index.
Identifiers: LCCN 2018039947 | ISBN 9781108496735 (hardback) |
ISBN 9781108721899 (paperback)
Subjects: LCSH: Computer science – Study and teaching – Handbooks, manuals, etc.
Classification: LCC QA76.27.C355 2019 | DDC 004.071–dc23
LC record available at https://lccn.loc.gov/2018039947

ISBN 978-1-108-49673-5 Hardback
ISBN 978-1-108-72189-9 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.

Published online by Cambridge University Press

To our parents, present and passed. A hardy generation – they should build battleships out of them.

Published online by Cambridge University Press

Published online by Cambridge University Press

Contents

List of Figures *page* x
List of Tables xii
List of Contributors xiii
Acknowledgments xv

0 An Important and Timely Field 1
SALLY A. FINCHER AND ANTHONY V. ROBINS

Part I Background 9

1 The History of Computing Education Research 11
MARK GUZDIAL AND BENEDICT DU BOULAY

2 Computing Education Research Today 40
SALLY A. FINCHER, JOSH TENENBERG, BRIAN DORN, CHRISTOPHER HUNDHAUSEN, ROBERT MCCARTNEY, AND LAURIE MURPHY

3 Computing Education: Literature Review and Voices from the Field 56
PAULO BLIKSTEIN AND SEPI HEJAZI MOGHADAM

Part II Foundations 79

4 A Study Design Process 81
AMY J. KO AND SALLY A. FINCHER

5 Descriptive Statistics 102
PATRICIA HADEN

6 Inferential Statistics 133
PATRICIA HADEN

7 Qualitative Methods for Computing Education 173
JOSH TENENBERG

8 Learning Sciences for Computing Education 208
LAUREN E. MARGULIEUX, BRIAN DORN, AND KRISTIN A. SEARLE

Published online by Cambridge University Press

9 Cognitive Sciences for Computing Education 231
ANTHONY V. ROBINS, LAUREN E. MARGULIEUX, AND BRIANA B. MORRISON

10 Higher Education Pedagogy 276
KERRY SHEPHARD

11 Engineering Education Research 292
MICHAEL C. LOUI AND MAURA BORREGO

Part III Topics 323

Systemic Issues 325

12 Novice Programmers and Introductory Programming 327
ANTHONY V. ROBINS

13 Programming Paradigms and Beyond 377
SHRIRAM KRISHNAMURTHI AND KATHI FISLER

14 Assessment and Plagiarism 414
THOMAS LANCASTER, ANTHONY V. ROBINS, AND SALLY A. FINCHER

15 Pedagogic Approaches 445
KATRINA FALKNER AND JUDY SHEARD

16 Equity and Diversity 481
COLLEEN M. LEWIS, NIRAL SHAH, AND KATRINA FALKNER

New Milieux 511

17 Computational Thinking 513
PAUL CURZON, TIM BELL, JANE WAITE, AND MARK DORLING

18 Schools (K–12) 547
JAN VAHRENHOLD, QUINTIN CUTTS, AND KATRINA FALKNER

19 Computing for Other Disciplines 584
MARK GUZDIAL

20 New Programming Paradigms 606
R. BENJAMIN SHAPIRO AND MIKE TISSENBAUM

Systems Software and Technology 637

21 Tools and Environments 639
LAURI MALMI, IAN UTTING, AND AMY J. KO

Published online by Cambridge University Press

22 Tangible Computing 663
MICHAEL HORN AND MARINA BERS

23 Leveraging the Integrated Development Environment for Learning Analytics 679
ADAM CARTER, CHRISTOPHER HUNDHAUSEN, AND DANIEL OLIVARES

Teacher and Student Knowledge 707

24 Teacher Knowledge for Inclusive Computing Learning 709
JOANNA GOODE AND JEAN J. RYOO

25 Teacher Learning and Professional Development 727
SALLY A. FINCHER, YIFAT BEN-DAVID KOLIKANT, AND KATRINA FALKNER

26 Learning Outside the Classroom 749
ANDREW BEGEL AND AMY J. KO

27 Student Knowledge and Misconceptions 773
COLLEEN M. LEWIS, MICHAEL J. CLANCY, AND JAN VAHRENHOLD

28 Motivation, Attitudes, and Dispositions 801
ALEX LISHINSKI AND AMAN YADAV

29 Students As Teachers and Communicators 827
BETH SIMON, CHRISTOPHER HUNDHAUSEN, CHARLIE MCDOWELL, LINDA WERNER, HELEN HU, AND CLIF KUSSMAUL

Case Studies 859

30 A Case Study of Peer Instruction: From University of California, San Diego to the Computer Science Community 861
LEO PORTER AND BETH SIMON

31 A Case Study of Qualitative Methods 875
COLLEEN M. LEWIS

Index 895

Figures

4.1 A study design process spanning four iterative phases that come before executing a study *page* 82
5.1 PI, OD, and DV 105
5.2 Generic FD 108
5.3 Comparing FDs 108
5.4 FDs with positive and negative skew 110
5.5 Positive skew in time to solve a homework problem 111
5.6 Bimodal FD 111
5.7 Factorial plot 112
5.8 Histogram-style factorial plot 113
5.9 Small, medium, and large standard deviations 116
5.10 Comparing scores from different distributions 117
5.11 Hypothetical scatter plots for height and weight, age and visual acuity, and height and IQ 120
5.12 Strong and weak positive correlations 120
5.13 Scatter plots of data sets 122
5.14 The influence of outliers on *r* 124
5.15 Possible experimental design 127
6.1 Population frequency distributions showing no effect and a real effect of the independent variable 137
6.2 The hypothesis testing decision matrix 142
6.3 Distribution of values of *t* for a given sample size and population variability when H_0 is true 148
6.4 Crossover interaction 152
6.5 Moderate interaction 152
6.6 No interaction 153
6.7 Generic one-way ANOVA table in SPSS 24 154
6.8 Factorial plot for computing interest study (hypothetical) 155
6.9 SPSS 24 output for two-way ANOVA 156
6.10 CI for a population mean 157
6.11 SPSS 24 output for a significant Pearson product moment correlation 160
6.12 Salary by years employed (hypothetical) 161
6.13 Outcome of a regression study predicting CS1 mark from math mark (hypothetical) 163
6.14 Linear regression (hypothetical data) with line of best fit in Excel 164

Published online by Cambridge University Press

6.15 Output of linear regression analysis in SPSS 24 165
6.16 A poor predictor 165
6.17 Line of best fit for a poor predictor 166
6.18 Linear regression analysis for a poor predictor in SPSS 24 166
9.1 Levels of analysis and their application to an example system 233
15.1 The combination of student-centered learning activity and teacher-centered instruction as adopted within blended learning 457
15.2 Memory visualization exploring primitive data types and the correspondence between variable name and value 466
15.3 Memory visualization of a C++ program facilitating understanding of memory addressing, allocation, and deallocation 467
17.1 Agreement and disagreement around two views of what computational thinking should be 515
20.1 A neural network with hidden layers 611
20.2 FitBit for dogs 621
20.3 Dog collar beacon program 626
22.1 A prototype tangible programming language based on computer vision technology 669
22.2 KIBO robot and its blocks 669
23.1 Process model for IDE-based learning analytics in computing education 685
23.2 Programming process data that can be automatically collected through a standard IDE 688
23.3 Data that can be automatically collected through an IDE augmented with additional features and functionality 689
23.4 Taxonomy of design dimensions for IDE-based interventions 696
24.1 Teacher knowledge 713
29.1 Reduction in course fail rates by course 838
29.2 Reduction in course fail rates for instructors teaching the same course with and without PI 839
29.3 Two locations of isomorphic, multiple-choice questions to test learning gain (q2) and retention (q3) 840
31.1 Completed drawing of a brick wall in Scratch 881

Published online by Cambridge University Press

Tables

0.1 Frequently discussed topics and the chapters they occur in *page* 7

5.1 Strength of positive correlations (apply symmetrically for negative correlations) 123

7.1 Data collection methods 184

7.2 Data analysis strategies 194

12.1 A programming framework 335

15.1 Pedagogic approaches and relevant pedagogic practices 450

23.1 Comparison of five IDEs used in computing education based on the data they collect for the programming category 691

23.2 Comparison of five IDEs used in computing education based on the data they collect for the social, testing, survey/quiz, and physiological categories 692

23.3 A taxonomy of useful information derivable from IDE data 693

25.1 Knowledge in the TPACK model 734

27.1 Truth table for modus ponens, P → Q 782

27.2 Overlap between logical operators AND, if-then, and if-and-only-if 784

29.1 Key findings of a study of PCRs 845

Published online by Cambridge University Press

Contributors

ANDREW BEGEL, Microsoft, USA

TIM BELL, University of Canterbury, New Zealand

MARINA BERS, Tufts University, USA

PAULO BLIKSTEIN, Stanford University, USA

MAURA BORREGO, University of Texas, USA

ADAM CARTER, Humboldt State University, USA

MICHAEL J. CLANCY, University of California Berkeley, USA

PAUL CURZON, Queen Mary University of London, UK

QUINTIN CUTTS, University of Glasgow, UK

MARK DORLING, Queen Mary University of London, UK

BRIAN DORN, University of Nebraska at Omaha, USA

BENEDICT DU BOULAY, University of Sussex, UK

KATRINA FALKNER, University of Adelaide, Australia

SALLY A. FINCHER, University of Kent, UK

KATHI FISLER, Brown University, USA

JOANNA GOODE, University of Oregon, USA

MARK GUZDIAL, Georgia Institute of Technology, USA

PATRICIA HADEN, University of Otago, New Zealand

MICHAEL HORN, Northwestern University, USA

HELEN HU, Westminster College, USA

CHRISTOPHER HUNDHAUSEN, Washington State University, USA

AMY J. KO, University of Washington, Seattle USA

YIFAT BEN-DAVID KOLIKANT, Hebrew University of Jerusalem, Israel

SHRIRAM KRISHNAMURTHI, Brown University, USA

Published online by Cambridge University Press

CLIF KUSSMAUL, Muhlenberg College, USA

THOMAS LANCASTER, Imperial College London, UK

COLLEEN M. LEWIS, Harvey Mudd College, USA

ALEX LISHINSKI, Michigan State University, USA

MICHAEL C. LOUI, Purdue University, USA

LAURI MALMI, Aalto University, Finland

LAUREN E. MARGULIEUX, Georgia State University, USA

ROBERT MCCARTNEY, University of Connecticut, USA

CHARLIE MCDOWELL, University of California Santa Cruz, USA

SEPI HEJAZI MOGHADAM, Google, USA

BRIANA B. MORRISON, University of Nebraska at Omaha, USA

LAURIE MURPHY, Pacific Lutheran University, USA

DANIEL OLIVARES, Washington State University, USA

LEO PORTER, University of California San Diego, USA

ANTHONY V. ROBINS, University of Otago, New Zealand

JEAN J. RYOO, University of California Los Angeles, USA

KRISTIN A. SEARLE, Utah State University, USA

NIRAL SHAH, Michigan State University, USA

R. BENJAMIN SHAPIRO, University of Colorado, USA

JUDY SHEARD, Monash University, Australia

KERRY SHEPHARD, University of Otago, New Zealand

BETH SIMON, University of California San Diego, USA

JOSH TENENBERG, University of Washington Tacoma, USA

MIKE TISSENBAUM, University of Illinois at Urbana Champaign, USA

IAN UTTING, University of Kent, UK

JAN VAHRENHOLD, Westfälische Wilhelms-Universität Münster, Germany

JANE WAITE, Queen Mary University of London, UK

LINDA WERNER, University of California Santa Cruz, USA

AMAN YADAV, Michigan State University, USA

Published online by Cambridge University Press

Acknowledgments

We would like to acknowledge and thank everyone listed as an author for their contributions to the open reviewing culture within which this book was written. Many of them also took part in the systematic reviewing process.

Other reviewers or advisors (who are not also authors) were Mordechai (Moti) Ben-Ari, Neil Brown, Paul Carins, Sebastian Dziallas, Christopher Hoadley, Yasmin Kafai, Raymond Lister, and Elizabeth Patitsas. Our thanks to them for taking the time to help us improve the work.

Shriram Krishnamurthi and Kathi Fisler at Brown University organized the reviewing of most chapters by students in a graduate computing education course. Thanks to those participants: Natasha Danas, Nicholas DeMarinis, Justin Pombrio, Sorawee Porncharoenwase, Sam Saarinen, Preston Tunnell Wilson, and John Wrenn. You helped a lot!

Thanks also to our editor David Repetto, content manager Adam Hooper, and editorial assistant Emily Watton at Cambridge University Press, and project manager Céline Durassier at Newgen Publishing UK, who provided us excellent support and allowed us the flexibility that we needed.

Finally, of course, we have to extend enormous thanks to our families for their patience and support.

Published online by Cambridge University Press

Published online by Cambridge University Press

0 An Important and Timely Field

Sally A. Fincher and Anthony V. Robins

Computing education (CEd) is important, everyone agrees. President Obama committed hundreds of millions of dollars to "Computer Science for All" (White House, 2016); governments have developed curricula and made computing a school subject across the world (see Chapter 18); online providers compete to teach coding (such as Code Academy, code.org, the Hour of Code, Khan Academy and Coursera); and tech giants put money into supporting CEd projects (CISCO supports the BlueJ and Scratch initial programming environments and Google funds substantial professional development programs and has produced a series of CEd research reports – one of which forms the basis for Chapter 3).

With all of the effort and resources going into CEd, it would be comforting to think that we know what we are doing – that the problems of teaching and learning computing topics are well understood, that the solutions are known, and that best practice is widely shared. But this ideal picture is very much a work in progress. We still don't know enough about how students learn computing subjects, what effects different teaching approaches have, or how to equally engage people of all races and genders in the field. CEd research (CEdR) is how we work to understand and improve this. In order to make the most of the resources currently going into CEd efforts around the world, CEdR is an important and timely field.

0.1 Why It's Important

The sudden interest and investment in CEd are no accidents. Computing technology is reshaping the world around us at an ever-increasing pace, changing the way that we work (or don't work), communicate, consume, learn, create, entertain ourselves, and more. At the time of writing, some of the major computing-driven issues being widely debated include: the rise of artificial intelligence applications (automation of work, digital assistants, self-driving vehicles); the use and abuse of social media and personal data; and the advent of disruptive cryptocurrencies.

In this context, there are many rationales as to why CEd and therefore CEdR are important. Blikstein and Moghadam (Chapter 3 of this handbook, summarizing their Google report into the current state of CEd) outline four of

https://doi.org/10.1017/9781108654555.001 Published online by Cambridge University Press

them. *The labor market rationale*: computing skills are explicitly required for an increasingly large number of jobs and will be generally useful for very many more. *The computational thinking rationale*: computational ways of thinking (e.g., algorithms, heuristics, problem-solving skills) are all useful and transferable. In practical terms, the belief that they are good preparation for later specialist topics (e.g., learning to program) has helped to drive the widespread introduction of computational thinking in schools (these and related issues are focuses of Chapters 17–20). *The computational literacy rationale*: general familiarity with programming and other computing skills is sometimes equated with mathematical or textual literacy (Chapters 18 and 19). Some further argue that computational literacy goes beyond computational thinking to enable new types of mental operations, knowledge representations, and modes of expression (see Chapter 3). *The equity of participation rationale*: computing knowledge will be increasingly required for the best jobs, for civic participation, and even for understanding the functioning of the society around us. As is readily apparent, current participation in CEd has a long way to go to achieve anything like equity of participation in terms of race or gender (see Chapters 16 and 24).

0.2 Why It's Timely

In 2004, when *Computer Science Education Research* (Fincher & Petre, 2004) was written, CEd researchers were essentially all academics teaching in university computer science departments. Almost universally they had strong disciplinary computing backgrounds and their loci of research were their interests, which meant (almost universally) tertiary-level CEd. In the intervening 15 years, times have changed. Researchers in schools of education are becoming interested in the field, there are many agencies funding research, and there are increasing numbers of specialist academics with PhDs in CEd. The locus of research has expanded to include adult "returners" and children in K–12 education (which adds the complexity of cognitive development to the mix). And the subject itself has expanded beyond its academic disciplinary construction in university computing departments, bursting through the classroom walls into everyday life and "computational thinking."

One of the ways in which these changes are reflected is in our decision, as editors, to use the inclusive phrases "computing" education (CEd) and "computing" education research (CEdR), which the reader will find in widespread use in this handbook, instead of the previously common "computer science" phrasing.

As interest in the subject explodes, and as teaching and learning of computing happen in ever more diverse ways in ever more diverse environments, so CEdR must keep pace. This handbook is a contribution to the widening discourse – in it, we capture what is already known, look out to what is known in other fields, and examine what we might be moving toward as computing technologies continue to evolve and our knowledge of CEd develops.

https://doi.org/10.1017/9781108654555.001 Published online by Cambridge University Press

0.3 How This Book Is Organized

All of the chapters are new and have been written explicitly for this handbook (Chapters 3 and 23 draw on previous sources). However, we wanted this to be more than just a collection of writings from a collection of interesting authors. We solicited chapters in three **parts**. The purpose of the **Background** part is to briefly orient the reader within the field of CEdR, its history, and its current status. The substantive Foundations and Topics parts each have a specific purpose and a different set of shared **themes** running "vertically" through their chapters.

The **Foundations** part serves a "textbook" function. It is intended to set our field in context and to be a practical guide for new researchers. In 2005, the Association for Computing Machinery's Special Interest Group in Computer Science Education (SIGCSE) held a panel discussion on challenges to CEdR. Panelists commented on the "isolation" of our discipline, that "Too much of the research in computing education ignores the hundreds of years of education, cognitive science, and learning sciences research that have gone before us." They reflected on "the challenge of diversifying the research methods we employ" and on the need to understand our methods and seek rigor (Almstrum et al., 2005). Looking back on these challenges more than ten years later, although there has been progress, we think it is fair to say that every one of them remains relevant today. This handbook works to address those concerns, to provide an overview of CEdR work and methods, and to show how they fit with other intellectual traditions.

Some of the chapters in this part are broadly concerned with "methods," both quantitative and qualitative (Chapters 4–7). Although a reader might expect to find similar chapters – certainly similarly titled chapters – in many books, our authors have closely contextualized these within CEdR work. Note that there are two extensive chapters on statistical methods. In our opinion (frequently reinforced while reviewing in various contexts), the need for an improved level of statistical rigor is a particular priority within our field.

CEdR (like education research more broadly) borrows techniques (and terminology and methods) from other disciplines in a "trading zone" activity. The idea of an intellectual "trading zone" was first proposed by Peter Galison in his work on physics:

> I intend the term "trading zone" to be taken seriously, as a social, material, and intellectual mortar binding together the disunified traditions of experimenting, theorizing, and instrument building [in subcultures of physics]. Anthropologists are familiar with different cultures encountering one another through trade, even when the significance of the objects traded – and of the trade itself – may be utterly different for the two sides.
> (Galison, 1997)

It is one of the goals of this handbook to situate CEdR with related fields, our "intellectual trading partners," from whom we have much to learn, and to whom we have much to offer: Chapters 8–11 in the Foundations part seek to

https://doi.org/10.1017/9781108654555.001 Published online by Cambridge University Press

do just that. They each articulate expertise from key partner disciplines and explore their boundaries and our common edges. Over time, of course, the nature of the trading zones and their borders shift. These chapters focus on historical CEd trading zones where researchers trained as computer scientists encountered unfamiliar epistemologies and methods. As computing becomes a school subject and becomes an integrated part of other disciplines (such as bioinformatics), new and important trading partners will emerge.

The authors of chapters in the Foundations part were asked to addresses three particular themes: what can we learn **empirically**, **methodologically**, and **theoretically** from their distinctive, separate foundational fields. Themes are signaled in different ways in various chapters, sometimes as section headings, sometimes as bold keywords in relevant places in the text.

The **Topics** part contains chapters that explore the "state of the art." We have chosen topics that are of current and abiding interest in CEdR to illustrate the kinds of problems that we are trying to address and why they matter to us. Very often in this part, chapters draw on a considerable body of existing work, which should help orient new researchers. The themes of this part were **motivational context** (why we care about this issue), **implications for practice**, and **open questions**/suggestions for future research. Once again, these themes will be reflected in various ways in the chapters.

CEdR is essentially applied research, and there is no point in doing this kind of research if you are not interested in affecting practice and making CEd better in some way, perhaps more effective or more equitable. So "implications for practice" was chosen as a theme for this part, as a reminder both to researchers and practitioners that our work is meant to be useful.

The chapters in this part are grouped into four subsections. *Systematic Issues* (Chapters 12–16) are the "bread and butter" areas that continually offer questions and often attract the interest of new researchers. Chapters in the *New Milieux* section (Chapters 17–20) consider more recent issues that have arisen with the spread of computing beyond the "traditional" university setting, situated in a formal classroom within a department of computing. The *Systems and Software Technology* section (Chapters 21–23) recognizes a disciplinary advantage that CEd researchers have, in that we can build computational tools both in support of CEd and to provide new lenses for CEdR. The *Teacher and Student Knowledge* section (Chapters 26–29) investigates issues concerned with the production and acquisition of computing knowledge.

0.3.1 Case Studies

We conclude the Topics part with two case studies. They are written from a different viewpoint and serve a different purpose to other chapters. Chapter 30, *A Case Study of Peer Instruction*, covers the life cycle of an intervention, demonstrating how results from CEdR may be directly applied to practice. Chapter 31, *A Case Study of Qualitative Methods*, details the progress of a paper from inception to publication. This is an uncommon view, and a valuable one. Many CEd

https://doi.org/10.1017/9781108654555.001 Published online by Cambridge University Press

researchers come from computing backgrounds, with a grounding in analytic knowledge. Seeing how qualitative research "plays out" in a study, and in presentation, broadens understanding.

0.4 How This Book Was Written

Although talked about in a desultory fashion for a period of time, a definite point marked the genesis of this handbook. A fortuitous period of study leave following Dagsthul seminar 16072 (Assessing Learning in Introductory Computer Science) allowed the prospective editors to be in the same place for an extended time.

We met frequently over a period of weeks to write the proposal, mapping out an initial list of chapters and identifying a pool of possible contributors (many of whom we had already talked with at Dagsthul and other community venues, such as the annual International Computing Education Research conference). Once the proposal had been reviewed and accepted, we formally invited contributors, directing them to a Google document indexing the proposal, our provisional chapter list, and other relevant resources. Over the first few weeks, building on feedback from both reviewers and contributors, the chapter list was modified and extended, and prospective authors "voted" for the topics and chapters they were interested in contributing to. From this, a lead author was identified for each chapter and writing teams were organized. An initial deadline for chapter completion was set.

Every chapter was written as a Google document, viewable and editable by all contributors to the handbook. We established the convention that only the authors writing a particular chapter were expected to edit the main text, with other contributors suggesting changes, leaving marginal comments, or adding notes in preface pages specifically designated for that purpose. This totally open model was designed to encourage a high level of peer review and discussion, and in that it was moderately successful. Collectively, the contributors left 2,070 comments on their own and each other's chapters, and many more extended notes in the preface pages. Most chapters benefited significantly from this communal sharing of knowledge.

This was a high-trust model from the outset, and we were alert to a number of infringements that would be possible. We were worried that the model might risk authors making unauthorized changes to another person's text; that there might be public and unresolvable disputes between authors; that text might be "borrowed" and published elsewhere; and, in good open source style, that someone (or some group) might decide to disrupt the project. None of that happened.

There were minor abuses, which are common to all models of collective action (Ostrom, 1990). There was some *free-riding* (where contributors took their share of the benefits, but did not make an equal contribution) and some *rule-breaking* (authors not delivering on time, or at all; authors extending their

https://doi.org/10.1017/9781108654555.001 Published online by Cambridge University Press

authorial invitation to others), but these could have equally occurred in less trustful forms of engagement. A more significant issue – and not obvious to us at the outset – was that some chapters went in different (sometimes very different) directions from what we expected. This was sometimes a good thing, where our initial expectations were poorly informed, but occasionally led to more mixed outcomes.

Our initial deadline came and went (as deadlines will do) with many chapters incomplete. As experienced academics, we had anticipated and allowed for this, and so put the second deadline in place. During this period, we began a systematic review, where every chapter was read and commented on by a group made up of both the editors, other handbook contributors with relevant expertise, and, on occasion, additional subject experts. Each week, a review group met with chapter authors (as time zones allowed) via Skype to discuss reviewers' feedback.

In parallel, a second, independent, and rather unusual review process was taking place. Contributors Shriram Krishnamurthi and Kathi Fisler were teaching a graduate course in CEd at Brown University and suggested their class might read and review the chapters. As graduate students with an interest in the field, they represented one of the key intended audiences for the handbook; this was a great opportunity that we enthusiastically accepted. The class subsequently reviewed almost every chapter (a huge achievement!), and contributors benefited from the additional, external, targeted feedback resulting from this process. At the end of these review processes, authorial teams responded to feedback (some chapters changed markedly in this process) and finalized chapters.

0.5 Accessible Structure

As already mentioned, one of our goals for the Handbook was to present a well-organized body of work, where the structure is made evident and accessible to the reader. This goal drove the organization into parts, and the "vertical themes" running through Parts II and III. We also encouraged authors to make explicit cross-references to other chapters where applicable. This resulted in more than 120 internal references to other chapters to help the reader find related information all over the book (a result that would not have been possible without the open writing process).

As a final effort in this regard, Table 0.1 lists topics that receive substantial attention across several chapters. It is a mix of broad theoretical frameworks (constructivism, cognitivism, behaviorism) and topics specific to our field (learning to program, the notional machine, the McCracken study of novice programmers). We were surprised by some of the entries (Logo commands a lot of attention!). Many other topics are discussed, in varying levels of detail, in more than one chapter. We regard this overlap as a feature, not a bug – from the multiple perspectives of multiple authors, the reader should get a sense of the scope and richness of these topics.

https://doi.org/10.1017/9781108654555.001 Published online by Cambridge University Press

Table 0.1 *Frequently discussed topics and the chapters they occur in.*

Topic	Chapters
Logo (Papert)	1, 3, 8, 17, 19, 20, 22, 27
Constructivism (Vygotsky, Bruner)	1, 8, 9, 10, 11, 15, 24, 29
Teaching methods	1, 8, 10, 12, 15, 24, 27
Learning to program	1, 3, 12, 13, 21, 27
The notional machine	1, 12, 13, 15, 21, 27
Cognitivism	1, 9, 10, 11, 15
Behaviorism	1, 9, 10, 11, 15
Assessment	10, 11, 14, 18, 21
Motivation/efficacy	3, 11, 17, 24, 28
McCracken study	1, 4, 12, 13, 14
Computational thinking	3, 17, 18, 20, 24
Equity/diversity	11, 15, 16, 24

0.6 Looking Back

Our process was "loosely specified." In retrospect, we might have made life easier for ourselves if we had implemented a more formal process. We could have asked authors to submit outlines and drafts of chapters for approval, or we could have asked more people to take on editorial oversight. Such measures would have likely made chapters more consistent and may have led to there being less overlap in some topic areas, with authors "carving up the territory" and claiming the right to reference certain topics exclusively.

What we gained through our process, however, is a collection of strong contributions, with every authorial team working on topics they cared deeply about. Our trade-off is a plurality of views on some topics and idiosyncratic presentation in some chapters.

It is unlikely that this is the first work that has been produced in such an open and collaborative fashion, but it is almost impossible to conceive of such a process being possible in earlier times without the infrastructure of the twenty-first century internet.

0.7 Looking Forward

The handbook is finished, but the work is not. Given the technological, social, and educational changes currently in progress, there is every reason to expect a new wave of interest and participation in CEdR. We hope that the handbook will serve as a useful resource for some time to come as instruction for the novice, a guide for the curious, and a companion for the experienced.

https://doi.org/10.1017/9781108654555.001 Published online by Cambridge University Press

References

Almstrum, V. L., Hazzan, O., Guzdial, M., & Petre, M. (2005). Challenges to computer science education research. In *Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education* (pp. 191–192). New York: ACM.

Fincher, S., & Petre, M. (2004). *Computer Science Education Research.* London: RoutledgeFalmer.

Galison, P. (1997). *Image and Logic: A Material Culture of Microphysics*. Chicago, IL: University of Chicago Press.

Ostrom, E. (1990). *Governing the Commons: The Evolution of Institutions for Collective Action*. Cambridge: Cambridge University Press.

White House (2016). Computer Science for All. Retrieved from https://obamawhitehouse.archives.gov/blog/2016/01/30/computer-science-all

https://doi.org/10.1017/9781108654555.001 Published online by Cambridge University Press

PART I

Background

Published online by Cambridge University Press

PART II

Foundations

Published online by Cambridge University Press

PART III

Topics

Published online by Cambridge University Press

New Milieux

Systems Software and Technology

Published online by Cambridge University Press

Published online by Cambridge University Press

Teacher and Student Knowledge

Published online by Cambridge University Press

Case Studies

Published online by Cambridge University Press

Index

Abelson, H., 19, 384, 622, 624, 663
abstraction, 524, 526–527, 531, 533–534, 555–557, 622, 784
  CRA, 784–786
academic development, 282–283
academic integrity, 421, 429, 430, 431, 434–435
academic misconduct, 415, 421, 428–429, 434, 435
access, 482, 483–484, 489
accounts, 183
ACM (Association for Computing Machinery), 293, 327, 328, 329
active learning, 304–305, 448–449, 451–452, 491–492, 502
activities, 182–183, 456–457
activity theory, 217
Adaptive Control of Thought (ACT / ACT-R) model, 246
Advanced Placement (AP) exams, 425
Ahadi, A., 680, 686
Aivaloglo, E., 562
algorithm animations, 28–29
algorithm visualization (AV), 650, 651–652
algorithmic thinking, 524, 525–526, 531
algorithms, 29, 256, 380, 513–514, 520, 525–526, 530, 615–617, 786
  approximation, 617
aliasing, 379, 388
Alice programming environment, 394, 756
Almstrum, V., 532–533
alpha bloat, 168–169
AlSulaiman, S., 670, 671
Amos, D. M., 790–791
analogical encoding, 253
Anderson, G. L., 728
Anderson, J. R., 22, 23, 210, 342
Anderson, R., 26
Angel, P., 464, 501
Angotti, J. A. P., 462
animated representations, 28–29
ANOVA (analysis of variance), 144, 147, 150–157
Anton, G., 464, 501
Apa Pura, 627
APIs (Application Programming Interfaces), 612–613, 700
approximation algorithms, 617
Armoni, M., 390, 533, 537, 553, 556, 561, 562
artifacts, 183
assessment, 284–286, 358, 414, 415, 416, 421–423, 425–426, 431–433, 435
  automated, 427, 433, 643, 648–650
  computational thinking, 534–537
  K–12 education, 563
  learning outcomes, 416–417
  plagiarism, 416
assessment design, 414, 415, 418, 429–430
assessment literacy, 424–425
assessment repositories, 426
assurance of learning, 284, 288
ASSYST, 648, 649
attention, 234–235
attitudes, 804, 812
  emotions, 816–817
  engagement, 814–816
  interest, 813–814
automated assessment, 427, 433, 643, 648–650

Ball, D. L., 715, 729
Bandura, A., 299, 806
Barnett, S. M., 250
BASIC, 11, 15, 21, 343
Basso, K. H., 197–198
batch programs, 382
Batista, A. L. F., 462
Bayesian analysis, 133–134
beacons, 22
Bebras (international competition), 535, 537
Beck, L., 454–455
behavioral psychology, 446–447
behaviorism, 244–245, 278–279, 447
Ben-Ari, M., 390, 553, 556, 560, 561, 562, 788, 793–794
Ben-Bassat Levry, R., 560
Bennedsen, J., 71, 249, 260, 330, 331, 341–342, 345, 347, 354, 355, 357
Benotti, L., 560
Berland, M., 66, 67, 68, 337
Berlin Model, 549–550, 551, 563
Berliner, D. C., 730
Berry, M., 341, 529

https://doi.org/10.1017/9781108654555.033 Published online by Cambridge University Press

Bers, M. U., 519
biased statements, 494–496
Biggs, J., 280, 282
Black, P., 416–417
black students, 489, 492
blended learning, 449, 457, 458–459
Blikstein, P., 665
blocks-based languages, 19–20, 28, 382, 388, 389–391, 594
BlockyTalky project, 223–224, 620–621
Bodner, G. M., 299–300
Bonferroni correction, 169
Boolean operators, 783
Booth, S. E., 23, 24, 759
Bootstrap project, 26, 244, 566, 590
Boustedt, J., 198–199, 449, 451, 757
Boxer programming language, 19–20, 586, 663
Boyer, E. L., 276–277, 563
Boys' Needlework, 174–178, 179, 181, 186, 194–196, 201
Brady, C. E., 464, 501, 591
Brennan, K., 519, 536, 628
Brooks, F. P., 329
Brooks, R. E., 14, 17, 22, 337, 357
Brown, M. H., 28
Brown, N., 426, 430, 759
Bruner, J. S., 212, 278
Buechley, L., 64, 66, 67–68, 71–72
Burke, Q., 567
Butler, M., 434–435

Campos, P. G., 454
camps, 755–756
Cannara, A. B., 16, 17
Carbone, A., 418, 424, 453
Carroll, J., 431
Carver, S., 21, 587, 588
case comparison, 196, 200–201
Caspersen, M. E., 249, 260, 330, 331, 345, 347, 357
Catrambone, R., 348
Ceci, S. J., 250
childhood education, early, 668–670
chi-squared test, 144, 159
Chizhik, A., 454–455
Christopher (vignette), 492, 493
chunking, 236, 344, 780–781
Clancy, M. J., 27, 66, 73, 344, 788
CLT (cognitive load theory), 84, 257, 258–259, 298–299, 344–345
COBOL, 11, 17
Cochrane, J., 741, 742
code reuse, 419, 433–434
coding, 14–15, 72
cognition, 711
cognitive apprenticeship, 211, 213–215
cognitive constructivism, 448
cognitive development, 242–244, 346–347
cognitive factors, 802–803
cognitive load, 242, 256–259, 261, 343–344, 391
  notional machines, 388
  Parsons' problems, 261
  subgoal labels, 253, 260–261
  worked examples, 259–260
cognitive processes, 279, 790–791
  Adaptive Control of Thought model, 246
  Soar Cognitive Architecture, 246
cognitive psychology, 446, 447–449
cognitive science, 12, 20, 22, 209–210, 231, 232
cognitivism, 246–247, 279, 298, 423
Cohen's *d*, 130–131
coherence, study design, 98–99
collaborative learning, 452–454, 502, 562–563, 829–830
collusion, 419
Common Lisp, 788
community engagement, 758–759
comparison groups, 128
computation, 513–514, 517, 608–609
computational agents, 515, 516, 517–518, 520, 591
computational design, 532–533
computational literacy, 19, 27, 61–63, 585, 587, 597, 670
computational literacy rationale, 2, 59, 61–63
computational modelling, 513, 524, 529–530
computational systems, 514, 516, 627–628
computational thinking (CT), 61, 211, 513, 514–517, 518–519, 522–525, 530–531, 537–539, 540–541, 587–588, 625, 626–627, 631–632, 722–723
  abstraction, 524, 526–527, 533–534
  algorithmic thinking, 524, 525–526
  assessment, 534–537
  computational agents, 515, 516, 517–518, 520
  computational design, 532–533
  computational literacy, 62–63
  computational modelling, 529–530
  decomposition, 524, 528–529
  evaluation, 529
  generalization, 524, 527–528
  K–12 education, 555–557
  logical thinking, 524, 526
  Papert, 30, 516
  programming, 532–533, 585
  unplugged activities, 532
  Wing, 60–61, 513, 516, 520–521, 587–588
computational thinking rationale, 2, 59, 60–61
Computational Thinking Test (CTt), 536, 537
computer education, 209–211, 216, 557–559, 588–589
computer music, 628–630
computer science (CS), 59–60, 64, 65, 66–68, 70, 72–73, 208, 547–549, 550–551, 594–596, 622–625, 631–632, 670–671
  K–12 education, 569–570

https://doi.org/10.1017/9781108654555.033 Published online by Cambridge University Press

computer science teachers, 72–74, 595
computing, 11, 12, 292, 590, 625, 631–632, 640
Computing at School (CAS) UK, 426, 519, 539, 735
computing devices, 559–561
computing education (CEd), 1–2, 29–30, 31, 56–57, 59–60, 65–66, 67, 68–69, 74–76, 104, 249, 499–500, 584, 622–625, 631–632, 709–710, 718–719, 794–795, 801–802
computing education research (CEdR), 1, 2–4, 11–13, 27, 30, 52–54, 102–103, 225, 231, 246, 360–362, 749, 778, 801
computing paradigms, 607–608
computing pedagogies, 625
computing systems, 607
computing teacher knowledge, 709, 710, 712–716, 719–722
computing technology, 1, 11, 12
concept inventories (CIs), 426
conceptual change, 778–779, 791
confidence interval (CI), 157–158
confounding variable, 129–130
Connolly, T., 462
construct validity, study design, 94
constructionism, 213, 586
constructive alignment, 280, 281, 284, 416
constructivism, 211–213, 218, 278, 423, 447–449
constructivist pedagogies, 212–213, 222, 254–255, 447
content creation, 456, 457
contract cheating, 421, 428, 434
control groups, 128
Cook, P. R., 629–630
Cooper, G., 261, 345
Cooper, S., 394, 460, 559
cooperative learning, 454–455, 829–830
CoRe (content representation), 732–733
Corral, L., 554–555
correctness, 355, 427, 611, 615–618, 648
correlation, 118–125, 159–160
correlation studies, 146
  p-values, 144
CORT (code restructuring tool), 260
cost models, 400–401
course design, 358–359
  K–12 education, 565–568
CPD (continuing professional development), 727–728, 731–732, 734–735, 742–743
  Disciplinary Commons, 739–740
  informal learning, 759–760
  lead teachers, 734–737
  online communities, 759–760
  online professional communities, 741–742
  professional development of teachers, 727–728
  professional learning community, 737–741
    Craig, M., 418, 422, 424
critical thinking, 289, 290

Crosby, M. E., 453
CS1 (Computer Science 1), 328, 330–333, 356–358, 359–360, 617
Csizmadia, A. P., 422, 426
CSP (contributing student pedagogy), 455–457
CSTA (Computer Science Teachers Association), 56, 57
Cullen, R., 815
cultural narratives, 494–497
cultural–historical activity theory (CHAT), 217
curricula, 70–72, 394, 521–522, 547–548, 550, 709
curricular materials, 70
Curzon, P., 518–519, 532
CWSEI (Carl Wieman Science Education Initiative), 862–863

Dalbey, J., 354
Darmstadt Model, 550, 570
data analysis, 97, 187, 193–194, 196–197, 201
  Boys' Needlework, 186, 201
  case comparison, 192–193, 196, 200–201
  ethnographic adequacy, 192, 199–200, 201
  inductive categorization, 191–192, 196, 198–199, 201
  thick description, 190–191, 196, 197–198, 201
data collection, 97, 182–186, 201
  Boys' Needlework, 181, 195
  IDE, 686–695, 699–700
data scales, 106–107
Davis, E. A., 337, 788
Davis, M., 431
de Croock, M. B. M., 344
de Raadt, M., 261
debugging, 12, 14, 21, 22, 391, 399
declarative (explicit) memory, 239–240
decomposition, 524, 528–529, 531
DeLyser, L. A., 565
demonstration languages, 666–667
Denning, P. J., 516, 517, 518, 529, 534, 537
dependent variable (DV), 104–106
descriptive statistical techniques, 107, 131, 133
  graphical, 107–112
  numerical, 113–118
design thinking, 469
design-based research (DBR), 219–222
Dewey, J., 279
Diaz, D., 532–533
Digital Beads, 664
direct instruction, 212–213, 254
DiSalvo, B. J., 223, 391
discipline-based education research (DBER), 292, 295
diSessa, A. A., 19, 27, 61–62, 63, 72, 384, 585, 663
  Boxer programming language, 586
  student misconceptions, 792
dispositions, 289, 804, 817–819

https://doi.org/10.1017/9781108654555.033 Published online by Cambridge University Press

distributed computing, 609, 612–615, 618–622, 625–627
diversity, 463–464, 482–483, 500, 501
diversity of activity, 686
DLCI (Digital Logic Concept Inventory), 308
Doorenbos, J., 754
Dorn, B., 345, 594–595, 742, 753
Driscoll, M. P., 279
Druin, A., 20
du Boulay, J. B. H., 13, 26, 341, 355
Duval, E., 679
Dwyer, H. A., 553, 568
Dynabook, 586

e-books, 645, 647–648, 761
Eckerdal, A., 449, 451, 757, 817
EDM (educational data mining), 679
educational research, 20, 21, 276–277, 297
Ehrlich, K., 22, 23, 337, 386
El Ioini, N., 554–555
embedded programming, 402
emotions, 816–817
Emrich, C., 487
Enbody, R., 809
engagement, 814–816
Engeström, Y., 217
engineering assessment, 303, 308–309
engineering design, 305
engineering education, 292, 293–295, 299–301, 303, 307–308
engineering education research, 292–293, 295–297, 301–303, 309–312
  engineering assessment, 303, 308–309
  engineering epistemologies, 303–304
  engineering learning mechanisms, 303, 304–305
  engineering learning systems, 303, 305–306
  learning theories, 298–299, 301
  mixed methods, 302
  qualitative methods, 302
  systematic literature reviews, 302
engineering epistemologies, 303–304
engineering identity, 297–298, 307–308
engineering learning mechanisms, 303, 304–305
engineering learning systems, 303, 305–306
episodic memory, 239–240
epistemological pluralism, 70
equity, 482, 484, 500–502
equity of participation rationale, 2, 59, 63–65
error messages, 399–400, 650
Esmail, A., 451
ESP (empirical studies of programmers), 24–25
ethical study design, 96
ethnographic adequacy, 192, 199–200, 201
evaluation, 284, 286–288, 529
Evans, C., 424–425
event-driven programs, 382
expectancy-value theory (EVT), 211, 217–218, 219, 300
experimental hypothesis (**H**$_1$), 139, 140, 146
experts, 299
Exploring Computer Science (ECS) curriculum, 563–564
external validity, study design, 95–96
externally compiled languages, 666, 667
extraneous cognitive load, 84, 257, 258, 298, 343

Facione, P. A., 289
factorial design, 128
factorial plots, 109–112
Falkner, K., 434–435, 460, 502, 741
Falkner, N., 460, 502, 741
FCI (Force Concept Inventory), 308
feasibility, study design, 92–94
Feaster, Y., 561, 568
feedback, 255, 283–284, 286, 287, 416, 643
  informal learning, 762
Felleisen, M., 566
Fenwick, J. B., 451
Feynman, R. P., 615
Fiebrink, R., 629–630
file drawer problem, 169–170
Fincher, S. A., 26, 68, 84, 97, 338, 339, 354, 415, 422, 426, 759
Fisher, A., 463
Fisler, K., 355, 396, 566, 776, 793
Fitzgerald, S., 787, 793
Fletcher, G. H., 518
flipped classrooms, 304, 449, 457, 458–459
Flot, J., 560
Follman, D. K., 299–300
Foreman, B., 535
formative analytics, 469, 470
formative assessments, 286, 358, 416–417, 432
FORTRAN, 11, 15, 17
Franklin, D., 553, 568
frequencies, 158–159
frequency distributions (FDs), 107–109, 113, 115
Friend, J., 14, 16
Fronza, I., 554–555
functional (FP) paradigms, 380, 381, 382
functional context transfer, 251
functional languages, 383
fundamental ideas, 338

Galison, P., 3
Garrison, L., 790–791
Gasson, J., 118
Geertz, C., 190
Gellenbeck, E., 386
gender narratives, 497–498
gender participation gap, 803, 804–805
generalizability, 223
generalization, 224, 524, 527–528, 531

https://doi.org/10.1017/9781108654555.033 Published online by Cambridge University Press

Georgia Computes! project, 755, 756
germane cognitive load, 257, 258, 298, 343
Gick, M. L., 252
Gidget programming game, 84, 399, 646, 761
Giordano, D., 422, 426
Glitch Game Testers, 223, 391
Go, S., 742
goal orientation, 809–810
Gomez, M. J., 560
Good, J., 809
Goodwin, C., 199, 200
Gordon, M., 754
graduate attributes, 288–289
graphical descriptive statistics, 107–112
graphical languages, 592, 593–594, 665
Green, T. R. G., 13, 14, 28, 334–336, 593
Grissom, S., 787, 793
Grossman, P. L., 729
ground truth annotation, 686
Grover, S., 60, 61, 63, 65, 68–69, 70, 72, 389, 460, 553, 555, 556, 559, 564
Guo, P. J., 754, 757
Guzdial, M., 26, 61, 63, 67, 69, 71, 73, 345, 425, 463, 530, 532–533, 558, 590, 591, 680, 753

hackathons, 758
Haden, P., 118, 422, 452
Hagan, H., 532–533
Hallstrom, J. O., 561, 568
Hancock, C., 336–337
Hansen, A. K., 553, 568
Hansen, E. R., 568
HAR (human activity recognition), 685
Hardin, G., 200
Harel, I., 589
Harlow, D. B., 553, 568
Hattie, J., 283, 739
Hazzan, O., 424, 554
heapsort, 775–776
Heimann, P., 549
Herman, G. L., 783
Hermans, F., 562
hierarchical linear modelling, 167
Hobbs, R., 336–337
Hodges, S., 559–560
Holyoak, K. J., 247, 252
Horn, M. S., 60, 62, 63, 591, 670, 671
House, E. R., 731, 740
Huang, S.-L., 560
human languages, 27
Hundhausen, C. D., 28, 392, 453, 593
Hutchins, E., 514, 711
Hutchison-Green, M. A., 299–300
hypothesis testing, 134–135, 136, 139, 141–143
  morning coffee, 135, 139, 140
  pair programming, 136–137, 138–139, 140

identity development, 790, 791
IDEs (integrated development environments), 391, 392, 644–645
  data collection, 686–695, 699–700
  interventions, 680–682, 695–699
  learning analytics, 683–690, 700–701
Ihantola, P., 680, 686
importance, research questions, 86–87
indirection, 777–778
inductive categorization, 195, 196, 198–199, 201
inferential hypothesis
  morning coffee, 135
  pair programming, 136–137
inferential statistical techniques, 107, 113, 115, 131, 133–134, 137, 143–145, 146
  ANOVA, 150–157
  chi-squared test, 159
  confidence interval, 157–158
  correlation, 159–160
  frequencies, 158–159
  hypothesis testing, 136
  predictions, 146, 160–163
  t-tests, 144, 146, 147–150
informal learning, 749–751, 756, 760–763
  camps, 755–756
  community engagement, 758–759
  CPD, 759–760
  feedback, 762
  MOOCs, 754, 761
  online communities, 759–760
  online learning, 751–754
  peer learning, 757–758
  self-directed learning, 757
innate ability, 489, 495
Instructional Design Software Project, 589
intelligence, 248, 300–301, 350
intended learning outcome (ILO), 280–281, 285–286
interactive games, 645–647, 761
interactive machine learning (IML), 628, 630–631
interclass similarity, 686
interest, 813–814
internal validity, study design, 94–95
International Computing Education Research (ICER) Conference (2005), 11, 26
International Computing Education Research (ICER) Conference (2017), 40–54
Internet, 27
interventions, 358–359
  IDEs, 680–682, 695–699
interviews, 57–59
intraclass variability, 685
intrinsic cognitive load, 257, 258, 298, 343, 344
intrinsic motivation, 300, 538
introductory computing course, 490–491
introductory programming, 12, 260, 327, 353–356
intuition, 778–780, 791

https://doi.org/10.1017/9781108654555.033 Published online by Cambridge University Press

IQ (intelligence quotient), 243, 248, 350
Ishii, H., 664
Israel, M., 562–563
iteration, 394–396
ITSs (intelligent tutoring systems), 683
IVs (independent variables), 109, 125–128
  multiple, 128

Jadud, M., 84, 680
Java, 17, 213, 341, 381, 388, 390, 392, 403
Jaylisha (vignette), 492, 493
Jocuns, A., 790–791
John (vignette), 494, 495, 496

K–10 education, 548
K–12 education, 218, 547–550, 551, 552, 569–570, 616
  assessment, 563
  Berlin Model, 549–550, 551, 563
  collaborative learning, 562–563
  computational thinking, 555–557
  computing devices, 559–561
  course design, 565–568
  Darmstadt Model, 550, 570
  distributed computing, 619
  programming, 552–554
  software engineering, 554–555
Kaczmarczyk, L. C., 783
Kafai, Y. B., 174, 175, 181, 186, 194–196, 501, 519, 558, 567, 589
Kahney, H., 16, 787–788
Kanaparan, G., 815
Kastl, P., 555
Kay, A., 18, 19, 586, 588
Keating, D. P., 814–815
Kelleher, C., 356, 643, 756
Kellogg, S. B., 759
Kemeny, J. G., 584, 589
Khayarallah, H., 564
Kiara (vignette), 488–489, 490
KIBO, 667, 668–670, 671
Kiesler, S., 756
KIesmüller, U., 555
Kim, A. S., 754
Kim, J., 754
Kinnunen, P., 808, 817
knowledge domain transfer, 250–251
Ko, A. J., 83–84, 96, 97, 383, 399, 754
Koh, J., 670, 671
Kolikant, Y. B., 532–533, 730
Kölling, M., 335, 341, 426, 759, 760
Krishnamurthi, S., 566, 776
Kurland, D. M., 20–21
Kurtz, B. L., 451
Kurtz, T. E., 584, 589

labor market rationale, 2, 59, 60
Lane, D. M., 487
language acquisition, 244, 403
language design, 13, 14–15
language tools, 639, 641
Lash, T., 562–563
Lave, J., 216, 278, 790
Laxer, C., 532–533
lead teachers, 734–737
Leake, M., 760
learning analytics, 449, 469–470, 654, 679–682, 683
  IDE, 683–690, 700–701
learning communities, 448
learning outcomes, 352–353, 416–417, 801–802
learning programming, 11, 12–13, 20–23, 584, 639–640
learning sciences, 208–211, 218–219, 222–223, 224–225
  cognitive apprenticeship, 211, 213–215
  constructivism, 211–213, 218
  expectancy-value theory, 211, 217–218
  sociocultural theory, 211, 216–217, 219
learning sciences research, 209, 219
  design-based research, 219–222
learning theories, 246, 277–280, 423, 700–701, 709
  engineering education research, 298–299, 301
  sociocultural theory, 211, 216–217, 219, 710–712
Lee, I., 501, 518, 538
Lee, M. J., 83–84, 399
LEM (learning edge momentum), 351–352, 359–360
lesson study, 738–741
levels of analysis, 232–234, 262–263
Lewis, C. M., 389, 564, 760
Likert scales, 114, 115
linear regression, 161, 162–166
  hierarchical linear modeling, 167
  logistic regression, 167
  multiple linear regression, 167
Linn, M. C., 27, 344, 354, 788
Lishinski, A., 809
LISP, 22, 23
literacy, 597
Liu, A., 560
Lloyd, M., 741, 742
logic paradigms, 380
logical thinking, 524, 526, 531
logistic regression, 167
Logo programming language, 13, 15–16, 17, 18, 20, 21, 56, 210, 585–586, 587, 663, 787–788
Logo turtle, 15, 18, 20, 663
long-term memory (LTM), 237–240, 251
Loui, M. C., 783
Lu, J. J., 518
Luria, A. R., 189–190

https://doi.org/10.1017/9781108654555.033 Published online by Cambridge University Press

machine learning (ML), 609–612, 617, 618, 623–624, 628, 630–631
MacLeod, E., 559–560
Maiorana, F., 422, 426
Malik (vignette), 492, 493
Mann, L. M., 788
Mañoso, C., 464, 501
Marcos (vignette), 494, 495, 496
marginalization, 484, 486, 494, 497–499, 500
Margolis, J., 66, 67, 68, 73–74, 463
Margulieux, L. E., 225, 260–261, 348
Marr, D., 232
Marsden, S., 422, 426
Martell, R. F., 487
Martin, F., 336–337
Martinez, C., 560
Mason, D. D. M., 815
Mason, R., 261, 345
Mayer, R. E., 15, 17, 341, 343, 355, 357
Mazur, E., 835, 836
McCartney, R., 449, 451, 455, 757, 787, 793
McCracken Working Group, 25–26, 331, 417–418, 425, 432
McCracken, M., 532–533
Mead, J., 345
measures of central tendency (MCT), 113–115
measures of effect size, 130–131
Medel, P., 464, 501
media computation, 451, 463, 502
Media Computation curriculum, 596
Meerbaum-Salant, O., 390, 424, 553, 554, 556
Melvin, O., 562–563
memory, 232, 235, 240–242, 246, 279
  long-term memory, 237–240, 251
  sensory memory, 235–236, 237
  short-term memory, 236, 237
  working memory, 236–237, 344
memory categories, 240–241
mental models, 22, 26–27, 248, 340–341, 342–343, 358, 403
  notional machines, 341–342, 387, 465, 466–467
message passing, 620
metacognitive self-regulation, 811
microaggressions, 495, 496
Middle-years Computer Science (MyCS) project, 566–567
Miller, B., 754
Miller, L. A., 13–14, 387, 593, 807
MIMN (multi-institutional, multinational), 25–26
Mishra, S., 422, 426
Miura, I. T., 815
mixed-methods research, 302
modality transfer, 251
MOOCs (massive open online courses), 449, 457, 459–461, 502, 653, 741–742, 750, 751, 754, 761
Morgan, M., 434–435
morning coffee (hypothesis testing), 135, 139, 140
Morrison, B. B., 225, 247, 261, 345
Moström, J. E., 757
motivation, 299–301, 644, 804, 805, 812
  expectancy-value theory, 217–218, 219
multinational, multi-institutional (MNMI) studies, 82, 96, 97
multiple independent variables, 128
multiple linear regression, 167
multi-way ANOVA, 150–151
Murphy, L., 787, 793

Narayanan, N. H., 453
narratives, 481, 485, 488
  unconscious bias, 485–488
National Teachers' Centers (NTCs), Israel, 737
NCTM (National Council of Teachers of Mathematics), 57
Nelson, G. L., 383
neo-Piagetian theories, 243, 347, 350
NetLogo, 591, 626
neural networks, 610–612
NHST (null hypothesis significance testing), 133–134
Ni, L., 595
Nia (vignette), 488–489, 490
noncognitive factors, 802, 803, 819–820
  attitudes, 804, 812
  dispositions, 804, 817–819
  gender participation gap, 803, 804–805
  motivation, 804, 805, 812
nonparametric tests, 168
normalization, 115–118
notional machines, 12, 15, 17, 26–27, 248, 358, 382–384, 385–386, 387–388, 651, 786
  mental models, 341–342, 387, 465, 466–467
  object-oriented (OO) languages, 383, 393
  pedagogic practices, 464–467
  programming languages, 383, 398
novelty, research questions, 87–89
novice programmers, 23, 29, 217, 247, 251–252, 327, 335–337, 348–352, 356
  engineering education research, 299
  language design, 14–15
  notional machines, 383
  programming difficulties, 13–14, 18, 242, 334–336
  programming knowledge, 337–339
  programming strategies, 339–340
NSTA (National Science Teachers Association), 57
null hypothesis (**H**$_0$), 139, 140, 141, 142–143, 144–145, 146
numerical descriptive statistics, 113–118

O'Connor, K., 790–791
Obama, B., 1, 709, 722, 736

https://doi.org/10.1017/9781108654555.033 Published online by Cambridge University Press

object-oriented (OO) languages, 380–381, 392–394
notional machines, 383, 393
object-oriented (OO) paradigms, 380–381, 382
object-oriented (OO) programming, 18–19, 788, 793–794
one-tailed t-tests, 149
one-way ANOVA, 150, 154
online communities, 759–760
online learning, 751–754
online professional communities, 741–742
online Q&A communities, 753
operational definition (OD), 104, 105
order effects, 126–127
Orton, K., 464, 501
Ostrom, E., 200–201
outcomes-based education (OBE), 280, 281
outliers, 121

Page, B., 728
pair programming (PP), 160, 422, 451, 455, 827–829, 830–834
hypothesis testing, 136–137, 138–139, 140
Palumbo, D. J., 21, 71, 354, 587, 588
Pane, J. F., 387, 593
Papert, S., 18, 21, 30, 56, 70, 557, 622–623, 624, 627, 628, 631, 720–721
computational thinking, 30, 516
constructionism, 213
learning programming, 12, 584, 623
Logo programming language, 13, 15, 56, 210, 585–586, 663
paradigms, 377, 379–382, 386, 402–403
parametric tests, 168
Parsons' problems, 261, 355, 422, 452, 781
Parsons, D., 118, 422, 452
Pascal, 11, 15, 16–17, 22, 23, 401
Pasternak, A., 565
Pausch, R., 356, 394, 643, 756
PCK (pedagogical content knowledge), 728–731, 732, 734–735, 742–743
CoRe, 732–733
TPACK model, 733–734
Pea, R., 17, 20–21, 26, 30, 386, 460, 559, 586, 587
Pearson product moment correlation (r), 119–121, 159–160
pedagogic practices, 446, 449–451, 461–464, 467–470, 716–718
active learning, 304–305, 448–449, 451–452, 491–492, 502
blended learning, 449, 457, 458–459
collaborative learning, 452–454, 502, 562–563, 829–830
cooperative learning, 454–455, 829–830
CSP, 455–457
notional machines, 464–467
pedagogical code reviews (PCRs), 845
pedagogy, 445–446, 461, 501–502
behavioral psychology, 446–447
cognitive psychology, 446, 447–449
peer assessment, 455–456
peer instruction (PI), 451, 453, 827, 834–842, 861–864, 869, 870–873
UC San Diego, 863–868, 869–870
peer learning, 757–758
peer review, 456
PeerWise system, 457
perception, 234
Perkins, D. N., 336–337
Perlis, A. J., 27, 29, 589
Perlman, R., 20, 665
personal learning environment (PLE), 751
personality traits, 804, 817–819
Petersen, A., 418, 422, 424
Petre, M., 26, 28, 422, 593
p-hacking, 170–171
Phelps, G., 715, 729
phenomenography, 23–24, 198–199, 281
physical context transfer, 251
Piaget, J., 212, 243, 346–347, 350, 533
plagiarism, 414, 416, 418–421, 427–428, 430–431, 433–434, 435
contract cheating, 421, 428, 434
plugin IDE, 687, 699–700
Portillo, J. A. P., 454
Pournaghshband, V., 464, 501
PPIG (Psychology of Programming Interest Group), 24
predictions, 146, 160–163
predictor variables, 161
probabilistic thinking, 617–618
problem-solving, 71, 247, 248–249, 354
analogical encoding, 253
schemata, 22, 253–254
worked examples, 252–253
problem-based learning (PBL), 282, 305, 451
procedural (implicit) memory, 239
process-oriented guided inquiry learning (POGIL), 827, 848–853
productive failure, 215–216, 468
professional communities, 216
professional learning community (PLC), 737–742
program visualization (PV), 28–29, 642, 647, 650–652
programmer aptitude tests (PATs), 328–329
programming, 12–13, 27–29, 327–330, 333–334, 352, 360–362, 377, 584–585, 587, 588–589, 591–592, 596–598, 623, 625, 790
computational thinking, 532–533, 585
difficulties, 13–14, 18, 242, 334–336
programming knowledge, 337–339
programming languages, 11, 17, 20–23, 28, 71–72, 327, 334, 352, 355–356, 377–378,

https://doi.org/10.1017/9781108654555.033 Published online by Cambridge University Press

386–387, 391–392, 397–398, 586–587, 592–594, 597, 598, 639, 644–645
  error messages, 399–400, 650
  notional machines, 383, 398
  paradigms, 377, 379–382, 386, 402–403
  problem-solving, 396–397
  syntax, 378–379
  types, 401
programming models, 402
programming paradigms, 12
programming plans, 22
programming processes, 27
programming strategies, 339–340
Project Quantum, 426, 535
Prolog language, 22, 788–789
property of interest (PI), 103–105
prospective memory, 240
public education, 180
Pulimood, S. M., 463–464
p-values, 144, 145

qualitative research, 4, 201–202, 302, 875–877, 885–886, 888–892
  data analysis, 879–888
  data collection, 877–879
qualitative research studies, 173–174, 187–190
  Boys' Needlework, 174–178, 179, 181, 186, 187–188, 194–196
  case comparison, 192–193
  ethnographic adequacy, 192
  inductive categorization, 191–192
  sampling, 180–181
  thick description, 190–191
quantitative methods, 188, 301–302
quantum computing, 609, 615–616, 617–618, 621–622, 631

racial narratives, 494–496
Racket language, 620
radiation problem (Duncker), 252
Ragonis, N., 788, 793–794
Rainfall Problem (Soloway, 1986), 249, 396–397, 793
reading pedagogy, 789–790
reasoning, 189, 232, 247–249, 781–784, 791
recursion, 394–396
recursive thinking, 525
reductions, 774–775
regression, 161–162
  hierarchical linear modelling, 167
  linear regression, 161, 162–166
  logistic regression, 167
  multiple linear regression, 167
Reinecke, K., 754
Reiser, B. J., 23
repetition, 394–396
research questions, 83–91, 173, 178–179
Resnick, M., 20, 26, 62, 63, 70, 72, 519, 536, 625–626, 628, 664
retrospective memories, 240
Riedesel, C., 422, 426, 455, 533
Rist, R. S., 246, 338, 340, 354
Robins, A., 247, 351
robots, 463, 528, 560, 671–672
  KIBO, 667, 670, 671
  Logo turtle, 15, 18, 20, 663
Rodriguez, S., 464, 501, 563
Romeike, R., 555
Romero-Zaliz, R., 464, 501
Rubin, R., 754
Rubio, M. A., 464, 501
rule systems, 786–789

sampling, 180–181
Sanders, K., 449, 451, 455, 757, 794
scaffolding, 214, 215, 219, 243, 393
scaffolding tools, 643–644
scatter plots (scattergrams), 112, 119, 121
SCCT (Social Cognitive Career Theory), 298
Schaeffer, L. M., 348
Schanzer, E., 566
schemata, 22, 253–254, 298
schemata, memory, 241–242
Schneider, G. M., 354
Schulte, C., 341–342
Schunn, C., 560
Scratch projects, 554, 555, 567, 619
Searle, K., 174, 175, 181, 186, 194–196, 501
Segars, L., 561
Seiter, L., 535, 557
Selby, C., 519, 529, 532
self-determination theory (SDT), 300
self-directed learning, 757
self-efficacy, 298, 299–300, 308, 806–809, 812, 819
self-regulated learning theory (SRL), 89–90, 805
  goal orientation, 809–810
  metacognitive self-regulation, 811
  self-efficacy, 806–809, 812, 819
semantic memory, 239–240
semantic waves, theory of, 539
sensory memory (SM), 235–236, 237
Sentance, S., 62–63, 65, 66, 559–560, 737
Shapiro, R. B., 67, 69, 223–224, 621
Shapka, J. D., 814–815
Sheard, J., 418, 424, 434–435, 453, 455
Shehab, S., 562–563
Sherin, B. L., 590, 591
Shoop, R., 560
short-term memory (STM), 236, 237
Shulman, L. S., 277, 715, 728–729, 730
Siebert, S., 386
SIGCSE (Special Interest Group in Computer Science Education, ACM), 25, 26, 231, 329

https://doi.org/10.1017/9781108654555.033 Published online by Cambridge University Press

Sime, M. E., 14
Simmons, R., 336–337
Simon, B., 423, 424, 434–435, 808, 817
Simon, H. A., 209, 640
Sirkiä, T., 452, 466
Sleeman, D., 17, 27
Smalltalk, 18–19, 586
smart block languages, 666, 667
Smetsers, S., 556
Smetsers-Weeda, R., 556
Snow, E., 29, 563
Soar Cognitive Architecture, 246
social constructivism, 278, 827
social context transfer, 251
social justice, 484, 500
social media, 449, 468
sociocultural constructivism, 448
sociocultural theory, 211, 216–217, 219, 710–712
software engineering, 554–555, 684
software tools, 639–641, 650
SOLO taxonomy, 16, 347–348, 350, 422, 423–424, 535, 589
Soloway, E., 22, 23, 336, 337, 342, 355, 356, 386, 396
Sorva, J., 28, 29, 341, 343, 344, 346, 389, 466, 652
  notional machines, 387
  SOLO taxonomy, 424
SOTL (Scholarship of Teaching and Learning), 222, 294–295
soundness, research questions, 86
spaced practice, 255
Spohrer, J. C., 336, 355, 356
spontaneous transfer, 250, 256
Stager, G., 69
standard deviation, 115
Stasko, J. T., 28
state, 384–385
statistical errors, 168, 171
  alpha bloat, 168–169
  file drawer problem, 169–170
  p-hacking, 170–171
statistical techniques, 103, 106, 107, 121, 171
  descriptive statistical techniques, 131, 133
  descriptive statistics, graphical, 107–112
  descriptive statistics, numerical, 113–118
  inferential statistical techniques, 107, 113, 131, 133–134
statistical thinking, 617–618
Stefik, A., 386, 392
STEM (science, technology, engineering, and mathematics), 56, 500, 631, 721
stereotype threat, 490, 491
Stevens, P., 426, 759
Stevens, R., 790–791
structural racism, 487, 489
student feedback, 287
student misconceptions, 303–304, 388, 773–774, 776–777, 779, 791–794
  heapsort, 775–776
  indirection, 777–778
  reductions, 774–775
student participation, 492–493, 494
student-centered teaching, 281–282
students of color, 489
studio-based learning (SBL), 451, 453–454, 827, 842–847
study design, 81–82, 85, 90–92, 96–100, 187
  ethical, 96
  feasibility, 92–94
  research questions, 83–91
  validity, 94–96
subgoal labels, 225, 253, 260–261, 348
sublanguages, 398
summative assessments, 286
superbug, 17, 26
Suzanne (vignette), 497
syntax, 378–379
System Blocks project, 665
systematic literature reviews, 302

tangibility, 663–664, 667–668, 671
tangible computing, 672–674
tangible languages, 665–666, 671, 672
  demonstration languages, 666–667
  externally compiled languages, 666, 667
  smart block languages, 666, 667
Tashakkori, R., 451
Tate, S. R., 451
Taub, R., 561, 562
taxonomy of learning objectives, Bloom, 346, 348, 422, 423, 424
Teachback, 468
teacher knowledge, 72–74, 595, 714–717, 723, 728
  computing teachers, 709, 710, 712–716, 719–722
  PCK, 728–731, 742–743
teaching methods, 17–18
technology, 449
temporal context transfer, 251
Tenenberg, J., 26
text-based languages, 28, 389, 390, 592, 593, 594, 665
Thames, M. H., 715, 729
thick description, 196, 197–198, 201
Thies, R., 551, 561–562
Thomas, L., 532–533, 757, 794
three-way ANOVA, 153–154
threshold concepts, 338–339
Timperley, H., 283
Toll, D., 680, 686
tools, 11, 29, 639–644, 645–647, 653–656
  assessment, 648–650
  e-books, 645, 647–648, 761

https://doi.org/10.1017/9781108654555.033 Published online by Cambridge University Press

feedback, 648–650
interactive games, 645–647, 761
software tools, 639–641, 650
TPACK (Technological, Pedagogical and Content Knowledge) model, 733–734
trading zones, 3
transfer in learning, 71, 254–256, 355, 385–386, 403, 588
analogical encoding, 253
schemata, 253–254
worked examples, 252–253
translation, 13–14
transparent teaching, 491
Trueman, D., 629–630
Tsai, A., 564
Tseng, I.-C., 560
t-tests, 144, 146, 147–150
Turing machine, 514, 608, 615
Turkle, S., 70, 557, 623, 624, 720–721
turtles, 27
Logo turtle, 15, 18, 20, 663
two-way ANOVA, 151–153, 154–156
types, 401

UC San Diego (UCSD), 863–868, 869–870
Ullmer, B., 664
unconscious bias, 485–488, 489–490, 493, 498–499
underrepresentation, 177, 481–482, 487, 803
units of analysis, 193–194
unplugged activities, 532, 561–562
user interface technologies, 19, 20
Utting, I., 426, 532–533, 759

Vahrenhold, J., 551, 561–562, 565, 809
validity, 92–94, 187, 537
van Merriënboer, J. J. G., 260, 344
variability, 115
Verbert, K., 679
video games, 223, 469
Vihavainen, A., 680, 686
Vinikiene, L., 422, 426
visual languages, 388–389, 665
visualization, 465–467, 644, 652–653
Vivian, R., 460, 502, 741
von Neumann machine, 608
Vygotsky, L. S., 189, 212, 214, 243, 278, 448

Wahba, S. K., 561
Waite, J., 559–560
Warner, J., 754
Wason selection test, 781, 782–783
Web-CAT, 648, 649
Weerasinghe, A., 434–435
Weinberg, G. M., 13, 14, 329, 336
Weintrop, D., 28, 389, 390, 391, 464, 501, 553
Wenger, E., 216, 278, 448, 790
Western Apache, 197–198
Weyer, S. A., 17
Wherfel, Q. M., 562–563
White, J., 754
Wiedenbeck, S., 22, 342, 393, 808, 809
Wilensky, U., 28, 389, 390, 391, 464, 501, 553, 591, 625–626
Wiliam, D., 416–417
Williams, S., 460, 741
Wilson, G., 594
Wilusz, T., 532–533
Wing, J., 60–61, 517
computational thinking, 60–61, 513, 516, 520–521, 587–588
Winslow, L. E., 331, 336, 354
Wolz, U., 463–464, 567, 568
Wood, K., 118, 568
Woollard, J., 519, 529, 532
worked examples, 252–253
subgoal labels, 225, 253
working memory, 236–237, 344
Woszczynski, A. B., 567
Wu, C.-C., 560

Xie, B., 383

Yadav, A., 809
Yeomans, L., 559–560
Yongpradit, P., 61, 66, 67, 69, 72, 73

Zander, C., 449, 451, 757
Zanelatto, R., 754
Zhu, J., 754
Zilles, C. B., 783
Zingaro, D., 418, 422, 424, 840
zone of proximal development (ZPD), 214, 243–244, 448, 712, 716
z-scores (normal scores), 117–118

https://doi.org/10.1017/9781108654555.033 Published online by Cambridge University Press

https://doi.org/10.1017/9781108654555.033 Published online by Cambridge University Press