

SOFTWARE AND MIND

Andrei Sorin

EXTRACT

Index

**This extract includes the book's front matter
and the index.**

Copyright ©2013 Andrei Sorin

**The digital book and extracts are licensed under the
Creative Commons
Attribution-NonCommercial-NoDerivatives
International License 4.0.**

The index has three levels and detailed descriptions, and functions also as an alphabetical summary of the book's contents.

The entire book, each chapter separately, and also selected sections, can be viewed and downloaded at the book's website.

www.softwareandmind.com

SOFTWARE
AND
MIND

The Mechanistic Myth
and Its Consequences

Andrei Sorin

ANDSOR BOOKS

Copyright © 2013 Andrei Sorin
Published by Andsor Books, Toronto, Canada (January 2013)
www.andsorbooks.com

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning, or otherwise, without the prior written permission of the publisher. However, excerpts totaling up to 300 words may be used for quotations or similar functions without specific permission.

For disclaimers see pp. vii, xv–xvi.

Designed and typeset by the author with text management software developed by the author and with Adobe FrameMaker 6.0. Printed and bound in the United States of America.

Acknowledgements

Excerpts from the works of Karl Popper: reprinted by permission of the University of Klagenfurt/Karl Popper Library.

Excerpts from *The Origins of Totalitarian Democracy* by J. L. Talmon: published by Secker & Warburg, reprinted by permission of The Random House Group Ltd.

Excerpts from *Nineteen Eighty-Four* by George Orwell: Copyright ©1949 George Orwell, reprinted by permission of Bill Hamilton as the Literary Executor of the Estate of the Late Sonia Brownell Orwell and Secker & Warburg Ltd.; Copyright ©1949 Harcourt, Inc. and renewed 1977 by Sonia Brownell Orwell, reprinted by permission of Houghton Mifflin Harcourt Publishing Company.

Excerpts from *The Collected Essays, Journalism and Letters of George Orwell*: Copyright ©1968 Sonia Brownell Orwell, reprinted by permission of Bill Hamilton as the Literary Executor of the Estate of the Late Sonia Brownell Orwell and Secker & Warburg Ltd.; Copyright ©1968 Sonia Brownell Orwell and renewed 1996 by Mark Hamilton, reprinted by permission of Houghton Mifflin Harcourt Publishing Company.

Excerpts from *Doublespeak* by William Lutz: Copyright ©1989 William Lutz, reprinted by permission of the author in care of the Jean V. Naggar Literary Agency.

Excerpts from *Four Essays on Liberty* by Isaiah Berlin: Copyright ©1969 Isaiah Berlin, reprinted by permission of Curtis Brown Group Ltd., London, on behalf of the Estate of Isaiah Berlin.

Library and Archives Canada Cataloguing in Publication

Sorin, Andrei

Software and mind : the mechanistic myth and its consequences / Andrei Sorin.

Includes index.

ISBN 978-0-9869389-0-0

1. Computers and civilization.
2. Computer software – Social aspects.
3. Computer software – Philosophy. I. Title.

QA76.9.C66S67 2013

303.48'34

C2012-906666-4

Printed on acid-free paper.

Don't you see that the whole aim of Newspeak is to narrow the range of thought?... Has it ever occurred to you ... that by the year 2050, at the very latest, not a single human being will be alive who could understand such a conversation as we are having now?

George Orwell, *Nineteen Eighty-Four*

Disclaimer

This book attacks the mechanistic myth, not persons. Myths, however, manifest themselves through the acts of persons, so it is impossible to discuss the mechanistic myth without also referring to the persons affected by it. Thus, all references to individuals, groups of individuals, corporations, institutions, or other organizations are intended solely as examples of mechanistic beliefs, ideas, claims, or practices. To repeat, they do not constitute an attack on those individuals or organizations, but on the mechanistic myth.

Except where supported with citations, the discussions in this book reflect the author's personal views, and the author does not claim or suggest that anyone else holds these views.

The arguments advanced in this book are founded, ultimately, on the principles of demarcation between science and pseudoscience developed by Karl Popper (as explained in "Popper's Principles of Demarcation" in chapter 3). In particular, the author maintains that theories which attempt to explain non-mechanistic phenomena mechanistically are pseudoscientific. Consequently, terms like "ignorance," "incompetence," "dishonesty," "fraud," "corruption," "charlatanism," and "irresponsibility," in reference to individuals, groups of individuals, corporations, institutions, or other organizations, are used in a precise, technical sense; namely, to indicate beliefs, ideas, claims, or practices that are mechanistic though applied to non-mechanistic phenomena, and hence pseudoscientific according to Popper's principles of demarcation. In other words, these derogatory terms are used solely in order to contrast our world to a hypothetical, ideal world, where the mechanistic myth and the pseudoscientific notions it engenders would not exist. The meaning of these terms, therefore, must not be confused with their informal meaning in general discourse, nor with their formal meaning in various moral, professional, or legal definitions. Moreover, the use of these terms expresses strictly the personal opinion of the author – an opinion based, as already stated, on the principles of demarcation.

This book aims to expose the corruptive effect of the mechanistic myth. This myth, especially as manifested through our software-related pursuits, is the greatest danger we are facing today. Thus, no criticism can be too strong. However, since we are all affected by it, a criticism of the myth may cast a negative light on many individuals and organizations who are practising it unwittingly. To them, the author wishes to apologize in advance.

Contents

	Preface	xiii
Introduction	Belief and Software	1
	Modern Myths	2
	The Mechanistic Myth	8
	The Software Myth	26
	Anthropology and Software	42
	Software Magic	42
	Software Power	57
Chapter 1	Mechanism and Mechanistic Delusions	68
	The Mechanistic Philosophy	68
	Reductionism and Atomism	73
	Simple Structures	92
	Complex Structures	98
	Abstraction and Reification	113
	Scientism	127
Chapter 2	The Mind	142
	Mind Mechanism	143
	Models of Mind	147

	Tacit Knowledge	157
	Creativity	172
	Replacing Minds with Software	190
Chapter 3	Pseudoscience	202
	The Problem of Pseudoscience	203
	Popper's Principles of Demarcation	208
	The New Pseudosciences	233
	The Mechanistic Roots	233
	Behaviourism	235
	Structuralism	242
	Universal Grammar	251
	Consequences	273
	Academic Corruption	273
	The Traditional Theories	277
	The Software Theories	286
Chapter 4	Language and Software	298
	The Common Fallacies	299
	The Search for the Perfect Language	306
	Wittgenstein and Software	328
	Software Structures	347
Chapter 5	Language as Weapon	368
	Mechanistic Communication	368
	The Practice of Deceit	371
	The Slogan "Technology"	385
	Orwell's Newspeak	398
Chapter 6	Software as Weapon	408
	A New Form of Domination	409
	The Risks of Software Dependence	409
	The Prevention of Expertise	413
	The Lure of Software Expedients	421
	Software Charlatanism	440
	The Delusion of High Levels	440
	The Delusion of Methodologies	470
	The Spread of Software Mechanism	483
Chapter 7	Software Engineering	492
	Introduction	492
	The Fallacy of Software Engineering	494
	Software Engineering as Pseudoscience	508

Structured Programming	515
The Theory	517
The Promise	529
The Contradictions	537
The First Delusion	550
The Second Delusion	552
The Third Delusion	562
The Fourth Delusion	580
The <i>GOTO</i> Delusion	600
The Legacy	625
Object-Oriented Programming	628
The Quest for Higher Levels	628
The Promise	630
The Theory	636
The Contradictions	640
The First Delusion	651
The Second Delusion	653
The Third Delusion	655
The Fourth Delusion	657
The Fifth Delusion	662
The Final Degradation	669
The Relational Database Model	676
The Promise	677
The Basic File Operations	686
The Lost Integration	701
The Theory	707
The Contradictions	721
The First Delusion	728
The Second Delusion	742
The Third Delusion	783
The Verdict	815
Chapter 8 From Mechanism to Totalitarianism	818
The End of Responsibility	818
Software Irresponsibility	818
Determinism versus Responsibility	823
Totalitarian Democracy	843
The Totalitarian Elites	843
Talmon's Model of Totalitarianism	848
Orwell's Model of Totalitarianism	858
Software Totalitarianism	866
Index	877

Preface

The book's subtitle, *The Mechanistic Myth and Its Consequences*, captures its essence. This phrase is deliberately ambiguous: if read in conjunction with the title, it can be interpreted in two ways. In one interpretation, the mechanistic myth is the universal mechanistic belief of the last three centuries, and the consequences are today's software fallacies. In the second interpretation, the mechanistic myth is specifically today's mechanistic *software* myth, and the consequences are the fallacies *it* engenders. Thus, the first interpretation says that the past delusions have caused the current software delusions; and the second one says that the current software delusions are causing further delusions. Taken together, the two interpretations say that the mechanistic myth, with its current manifestation in the software myth, is fostering a process of continuous intellectual degradation – despite the great advances it made possible. This process started three centuries ago, is increasingly corrupting us, and may well destroy us in the future. The book discusses all stages of this degradation.

The book's epigraph, about Newspeak, will become clear when we discuss the similarity of language and software (see, for example, pp. 411–413).

Throughout the book, the software-related arguments are also supported with ideas from other disciplines – from philosophy, in particular. These discussions are important, because they show that our software-related problems

are similar, ultimately, to problems that have been studied for a long time in other domains. And the fact that the software theorists are ignoring this accumulated knowledge demonstrates their incompetence. Often, the connection between the traditional issues and the software issues is immediately apparent; but sometimes its full extent can be appreciated only in the following sections or chapters. If tempted to skip these discussions, remember that our software delusions can be recognized only when investigating the software practices from this broader perspective.

Chapter 7, on software engineering, is not just for programmers. Many parts (the first three sections, and some of the subsections in each theory) discuss the software fallacies in general, and should be read by everyone. But even the more detailed discussions require no previous programming knowledge. The whole chapter, in fact, is not so much about programming as about the delusions that pervade our programming practices. So this chapter can be seen as a special introduction to software and programming; namely, comparing their true nature with the pseudoscientific notions promoted by the software elite. This study can help both programmers and laymen to understand why the incompetence that characterizes this profession is an inevitable consequence of the mechanistic software ideology.

There is some repetitiveness in the book, deliberately introduced in order to make the individual chapters, and even the individual sections, reasonably independent. Thus, while the book is intended to be read from the beginning, you can select almost any portion and still follow the discussion. An additional benefit of the repetitions is that they help to explain the more complex issues, by presenting the same ideas from different perspectives or in different contexts.

The book is divided into chapters, the chapters into sections, and some sections into subsections. These parts have titles, so I will refer to them here as *titled* parts. Since not all sections have subsections, the lowest-level titled part in a given place may be either a section or a subsection. This part is, usually, further divided into *numbered* parts. The table of contents shows the titled parts. The running heads show the current titled parts: on the right page the lowest-level part, on the left page the higher-level one (or the same as the right page if there is no higher level). Since there are more than two hundred numbered parts, it was impractical to include them in the table of contents. Also, contriving a short title for each one would have been more misleading than informative. Instead, the first sentence or two in a numbered part serve also as a hint of its subject, and hence as title.

Figures are numbered within chapters, but footnotes are numbered within the lowest-level titled parts. The reference in a footnote is shown in full only the first time it is mentioned within such a part. If mentioned more than once,

in the subsequent footnotes it is usually abbreviated. For these abbreviations, then, the full reference can be found by searching the previous footnotes no further back than the beginning of the current titled part.

The statement “italics added” in a footnote indicates that the emphasis is only in the quotation. Nothing is stated in the footnote when the italics are present in the original text.

In an Internet reference, only the site’s main page is shown, even when the quoted text is from a secondary page. When undated, the quotations reflect the content of these pages in 2010 or later.

When referring to certain individuals (software theorists, for instance), the term “expert” is often used mockingly. This term, though, is also used in its normal sense, to denote the possession of true expertise. The context makes it clear which sense is meant.

The term “elite” is used to describe a body of companies, organizations, and individuals (for example, the software elite); and the plural, “elites,” is used when referring to several entities, or groups of entities, within such a body. Thus, although both forms refer to the same entities, the singular is employed when it is important to stress the existence of the whole body, and the plural when it is the existence of the individual entities that must be stressed. The plural is also employed, occasionally, in its normal sense – a group of several different bodies. Again, the meaning is clear from the context.

The issues discussed in this book concern all humanity. Thus, terms like “we” and “our society” (used when discussing such topics as programming incompetence, corruption of the elites, and drift toward totalitarianism) do not refer to a particular nation, but to the whole world.

Some discussions in this book may be interpreted as professional advice on programming and software use. While the ideas advanced in these discussions derive from many years of practice and from extensive research, and represent in the author’s view the best way to program and use computers, readers must remember that they assume all responsibility if deciding to follow these ideas. In particular, to apply these ideas they may need the kind of knowledge that, in our mechanistic culture, few programmers and software users possess. Therefore, the author and the publisher disclaim any liability for risks or losses, personal, financial, or other, incurred directly or indirectly in connection with, or as a consequence of, applying the ideas discussed in this book.

The pronouns “he,” “his,” “him,” and “himself,” when referring to a gender-neutral word, are used in this book in their universal, gender-neutral sense. (Example: “If an individual restricts himself to mechanistic knowledge, his performance cannot advance past the level of a novice.”) This usage, then, aims solely to simplify the language. Since their antecedent is gender-neutral (“everyone,” “person,” “programmer,” “scientist,” “manager,” etc.), the neutral

sense of the pronouns is established grammatically, and there is no need for awkward phrases like “he or she.” Such phrases are used in this book only when the neutrality or the universality needs to be emphasized.

It is impossible, in a book discussing many new and perhaps difficult concepts, to anticipate all the problems that readers may face when studying these concepts. So the issues that require further discussion will be addressed online, at www.softwareandmind.com. In addition, I plan to publish there material that could not be included in the book, as well as new ideas that may emerge in the future. Finally, in order to complement the arguments about traditional programming found in the book, I plan to publish, in source form, some of the software applications I developed over the years. The website, then, must be seen as an extension to the book: any idea, claim, or explanation that must be clarified or enhanced will be discussed there.

Index

A

- abbreviations, *see* acronyms
- abstract terms, used without a definition 58, 374–5, 376–8, 384
- abstraction
 - benefits of 96
 - in classification systems 95, 114, 636–7, 651–2
 - definition of 95, 636–7
 - examples of 95–6
 - as general programming concept 669–70
 - in hierarchical structures 95–6, 636–7, 651–2
 - levels of, *see* levels
 - in object-oriented programming 636–7, 657, 669–70
 - in the relational model 682, 683, 703, 706, 718, 723, 788, 800
 - in structured programming 525, 531, 592, 613
- abstraction, fallacy of
 - causing loss of alternatives 97, 113–16, 290
 - consequences of 97–8, 114–15
 - definition of 96–7, 113
 - and the elites 114, 115–16
 - examples of 97, 113–14, 115–16
 - in human phenomena 285, 831, 861
 - in knowledge structures 155, 418
- abstraction, fallacy of (*cont.*)
 - and mind models 155
 - and reification 122–3
 - see also* alternatives, loss of; mechanistic fallacies
- academia, *see* universities
- academic mechanism, *see* mechanism:
 - promoted by universities
- ACM, *see* computer associations
- acronyms
 - definition of 373 n. 1
 - used as new words 373–4, 403–4
 - used to impoverish knowledge 373–4, 395–6, 403–4, 864
- advertising
 - deception in 22, 220–1, 845, 857, 867–8
 - and the falsification principle 220–1
 - as mechanistic indoctrination 845, 857
 - using confirmations to deceive in 220–1
- agnosia, and novice performance 160
- agriculture, expertise in, vs. magic 47–8
- alphabet of thought, as basis of all knowledge 314–15, 316
- alternatives
 - in complex structures 119, 283–6, 829–31
 - in simple structures 93, 113–15, 283–6

- alternatives (*cont.*)
 in software phenomena 288, 831–2, 867
- alternatives, large number of
 and mechanistic models 283–6, 828–31
 misinterpreted as all alternatives 115, 290,
 296, 830–1, 832–3
 misinterpreted as creativity and free will
 280, 281, 283–6, 288, 292–4, 461–2,
 828, 831–2
 misinterpreted as indeterminism 828–31
- alternatives, loss of
 due to abstraction 97, 113–16, 290
 due to both mechanistic fallacies
 by degrading the concept of
 indeterminism 831, 832–3
 in the new pseudosciences 285
 by simplifying complex structures 123–4
 through software devices 418, 434–5,
 450, 462
 through software mechanism 126,
 289–90, 365, 367, 370, 446–8, 469,
 628–30
 and software totalitarianism 867, 870
 due to mechanistic approximation 829–31
 due to mechanistic language
 and deception 371–2, 384, 385
 and Newspeak 124–5, 400
 and similarity to software 292, 366–7,
 369–70, 430–1, 628–9
 and software totalitarianism 870
 and the term “information technology”
 398
 and the term “technology” 394
 and totalitarianism 863–4
 due to reification 120–1
 in software applications
 and the delusion of high levels 290–2,
 296, 446–8, 469, 628–30, 832–3
 due to abstraction 115–16
 due to emulating manufacturing
 concepts 450
 due to programming incompetence
 126, 291
 and similarity to language 370, 628–9,
 706
 and software totalitarianism 867, 870
- analogies
 and formal theories 76
 as metaphors 78
- analysis, in simple structures 94–5
- analytic philosophy, and language mechanism
 319–21
- Andreski, Stanislav, and structuralism 250
- animals, used to explain human behaviour
 237, 239, 241
- applications, *see* software applications
- approximation
 mechanistic, *see* mechanistic approximation
 non-mechanistic, complex structures as
 148, 170–1
- art, and creativity 175–6
- artificial intelligence
 based on mechanism 145–6
 failure of 145–6, 183–6
 and mechanistic mind theories 145–6
 naivety of 145, 261–2, 435
- artificial languages
 based on simple structures 112, 309, 313–17,
 640–1
 as classification systems 314, 315–16, 640–1
 as means to attain new knowledge 314–15
 as means to express knowledge 314
see also perfect language
- artisans, *see* craftsmen
- aspects
 of software applications 348, 355, 419, 440–1,
 444, 445–6, 448, 472, 521, 542, 628,
 680–2
 of stories 351, 357, 401, 442–3
- assumption, false, *see* false assumption
- astrology
 and the fallacy of software engineering 510
 as pseudoscience 13
 resembling rational pursuits 510
- atomism
 ancient 73–4
 as basis of understanding 80
 causing the mechanistic myth 81
 in corpuscular theories 78
 definition of 9, 68
 in hierarchical structures 85–6
- atomistic philosophy, ancient 73–4
- atomists, ancient 73–4
- atoms, in the atomistic philosophy 73–4
- attributes
 in classification systems
 and complex structures 103–4, 642–4,
 652, 659
 and the idea of a perfect language 316,
 641
 and mechanistic approximation 11, 106,
 359
 and simple structures 95, 637, 646
 and Wittgenstein’s later philosophy
 338–41, 342–3, 356–9, 360–3
 combining in one structure 101–2, 338–41,
 356–9, 360–3, 642–4, 659
- in complex structures
 causing interactions 99, 105, 172–3, 641,
 642–4

attributes (*cont.*)
 in complex structures (*cont.*)
 and the idea of a perfect language 316,
 641
 and Newspeak 401
 and Wittgenstein's later philosophy
 337-41, 342-3, 346-7, 356-9, 360-3
 constant vs. changeable 173
 depicted as one within another 101-2,
 338-41, 356-9, 360-3, 642-4
 determining contexts 173
 examples of 11-12, 95-6, 98-9, 100, 337-41
 flow-control, *see* flow-control attributes
 in language structures 112, 401
 multiple 100, 101, 337-41, 342-3, 356-9,
 360-3, 542-3, 563-4, 642-4, 652, 659
 multivalued vs. two-valued 338 n. 11
 and non-mechanistic capabilities 104, 316,
 337-41, 356-9
 in object-oriented programming, *see* object-
 oriented programming: attributes in
 and reification 107, 117-18, 154-5
 repeated 101-2, 339-41, 356-9, 360-3, 642,
 659
 shared by several elements 101-2, 338-41,
 342-3, 356-9, 360-3, 642
 in simple structures 11-12, 95-6, 98-9, 100,
 106, 362
 of software entities 12, 28, 347, 360-3, 542,
 594-5
 and software processes 348, 360-3, 542,
 563-4
 in software structures, causing interactions
 360-3, 542, 563-4, 593-6, 658, 659
 attributes, in the relational model, *see*
 relational model: fields (columns,
 attributes) in
 autism, as failure to combine knowledge
 structures 180-1, 185
 automata, and formal languages 261-2
 automotive technology, analysis of term 397

B

Babel, and the idea of a perfect language 303-4
 Bacon, Francis, and utopianism 131-2
 basic file operations, *see* file operations
 begging the question, *see* circularity
 behaviour
 illogical, creativity as 183-4
 logical, lack of creativity as 183-4
 normal, as unspecifiable quality 183-4,
 188-9
 science of, behaviourism perceived as 235,
 236

behaviourism
 based on animal behaviour 237, 239, 241
 based on mechanism 236, 238
 definition of 235-6
 failure of 238, 240-2
 as means to create a perfect society 279
 perceived as science of behaviour 235, 236
 as pseudoscience 235, 238-9, 240-1, 279
 and reification 238
 reinstating the traditional concepts of mind
 240-1
 and simple structures 277
 stimulus-response process in 236, 238-9,
 240-2
 and the traditional concepts of mind 235-6,
 240-1
 Berlin, Isaiah
 and free will vs. determinism 840-1
 and modern myths 5
 and scientism 134-5
 binary operations, in structuralism 244-7,
 248
 biology, and scientism 136-7
 black-box principle, *see* structured
 programming: black-box principle in
 blocks, *see* software blocks
 Boden, Margaret A., and structuralism 249
 Boehm, Barry W., and software engineering
 499-500, 508
 Böhm, Corrado, *see* Böhm and Jacopini,
 work of
 Böhm and Jacopini, work of
 misrepresented as practical concept 566-7,
 568-9, 571-2, 605
 misrepresented as saying three elementary
 constructs 572-7, 585
 misrepresented as the black-box principle
 584
 perceived as proof of the structured
 programming theory 523, 566, 571-2,
 577, 578
 and the structure theorem 523
 as theoretical concept 568, 571-2, 577-8
 and Turing machines 571-2, 573
 Boole, George, and language structures 312-13
 Boolean logic
 invoked in structuralism 246
 as logic system 711
 and the rationalist language fallacies 312-13
 Brackett, Michael H., and denormalization
 796-7
 Brathwaite, Ken S.
 and normalization 772-4
 and the relational model 723
 Bruner, Jerome S., and creativity 174, 175-6

bureaucracy
 role of, under totalitarianism 30, 490–1,
 844, 847
 in software, *see* software bureaucracy;
 software bureaucrats
 universities as 14, 16, 275, 855
 business, *see* corporations
 business mechanism 844, 845, 856–7

C

C (programming language)
 GOTO statement in 610, 617
 as high-level language 464
 as hybrid language, in object-oriented
 programming 656
 statements in, vs. subroutines 463
 canoe building, expertise in, vs. magic 48–9
 capabilities
 mechanistic, *see* mechanistic capabilities
 non-mechanistic, *see* non-mechanistic
 capabilities
 Carnap, Rudolf, and logical positivism 325
 Carter, John, and normalization 776–7
 CASE (Computer-Aided Software Engineering)
 and application definitions 479–81
 failure of 479, 481–3, 600, 820–1
 and methodologies 481–3
 promises of 479, 535–6
 promotion of 536, 820–1
 and software mechanism 481–2, 535–6,
 600
 CASE (software construct), *see* software
 constructs, flow-control: CASE
 Cassirer, Ernst
 and artificial languages 308, 313
 and mana 57, 61, 62
 causality, fallacy of 211–12
 centralized economy
 as growing trend 860
 totalitarian nature of 133, 860–2
 century of genius, corpuscular fantasies in 75
 Channell, David F., and scientism 135, 136
 charlatanism
 due to mechanism 20–2, 92, 124, 713, 857
 in promotional work 21–2, 741–2, 846, 857
 in research, *see* corruption: in universities
 in software, *see* software charlatanism
 chick sexing
 through intuitive knowledge 167–9
 as non-mechanistic phenomenon 169
 similar to face recognition 169
 chicken's quandary
 and the problem of induction 212–13
 similar to human problems 212–13
 Chomsky, Noam
 and artificial intelligence 262
 and the concept of freedom and creativity
 281–3
 and formal grammars 262–3
 and the humanist cause 282
 and informal models 76
 and linguistics, *see* Chomskyan linguistics
 and Skinner's views on language 242
 Chomskyan linguistics
 and arguments against innate faculties
 170, 270–3
 based on mechanism 261–3
 circularity of 255, 270
 facts adduced to support theory 253–4
 failure of 251–2, 259, 264–5, 268, 273
 fallacies of 255–8, 262–3, 271–3
 fifth model (minimalist program) 266 n. 23
 first model 260–1
 as formal system 251, 257–8, 260–1, 262–3,
 264–5
 fourth model (principles and parameters)
 265–8
 and grammar 251, 257–8, 260–1, 262–3
 harmful effects of 252, 263, 281–3
 as informal system 76, 254
 and innate faculties 76, 252–5, 269–70,
 280–1, 303
 and mind models 261–2, 268, 270, 281–2
 and modularity 268
 and programming languages 263
 as project in artificial intelligence 262
 as pseudoscience 251–2, 259–60, 263–6,
 273, 280
 and rationalism vs. empiricism 280–2
 and reification 257, 258–9, 268
 reinstating the traditional concepts 267
 as revolution 251, 265
 second model (standard theory) 264
 and semantics 257–8, 263, 264–5
 and simple structures 277
 third model (extended standard theory) 265
 transformations in 260–1, 262–3
 and universal grammar 254–5, 273
 circular definition, software engineering
 definition as 495–6, 508–9
 circularity
 in Chomskyan linguistics 255, 270
 in the idea of a general will 851, 852
 in the idea of social forces 840
 in the idea of software engineering 495–7,
 499, 505, 508–9
 in the idea of software forces 842–3
 in the idea of totalitarianism 851–2, 853
 in ignoring falsifications 512

- circularity (*cont.*)
 in language mechanism 320
 in mechanistic theories 134, 152, 154, 207,
 255, 270, 320, 851–2
 in mind models 154
 in scholastic explanations 70, 151
- classes, in object-oriented programming, *see*
 object-oriented programming:
 classes in
- classification systems
 and abstraction 95, 114, 636–7, 651–2
 artificial languages as 314, 315–16, 641
 attributes in, *see* attributes: in classification
 systems
 and hierarchical structures 8, 93–4, 98–9,
 100, 103–4, 643–4
 as mechanistic approximation 11, 103–4,
 106, 359, 646
 and object-oriented programming 636–7,
 638–9, 642–3, 645–6, 652–4, 658–9,
 660
 and reification 315–16, 641, 645–6
 and software structures 349, 352, 354, 360–3,
 441, 542
 and Wittgenstein's later philosophy 338–41,
 342–3, 356–9, 360–3
- COBOL (programming language)
 file operations in 686, 690–3, 694, 697,
 698–9, 705
 GOTO statement in 617, 623 n. 47
 as high-level language 464
 as hybrid language, in object-oriented
 programming 656
 statements in, vs. subroutines 463
- Codd, E. F.
 as inventor of the relational model 713
 and the mathematics of normalization 766
- Cohen, I. Bernard, and scientism 134, 135, 137
- collective thinking, myths as 3–4
- columns, in the relational model, *see* relational
 model: fields (columns, attributes) in
- combinations, of attributes, in one structure
 101–2, 338–41, 356–9, 360–3, 642–4,
 659
- combinations, of file relationships 754
- combinations, of knowledge structures
 and creativity 174–6, 177–8, 183–7
 and mental disorders 178–81, 185
- comic strips, as myths 6
- Communism
 as myth 5
 as totalitarian democracy 849
 and utopianism 132
- complex capabilities, *see* non-mechanistic
 capabilities
- complex knowledge, *see* non-mechanistic
 knowledge
- complex phenomena, *see* non-mechanistic
 phenomena
- complex problems, *see* non-mechanistic
 problems
- complex structures
 alternatives in 119, 283–6, 829–31
 attributes in, *see* attributes: in complex
 structures
 and contexts 172–3, 184–5, 187, 188–9
 and creativity 175, 181, 182, 184–6
 definition of 90, 100
 examples of 109, 110, 112, 147–8
 knowledge as 148, 154–5, 164, 165–7, 176,
 270–3, 315–17, 323, 417–18, 641, 651–2
 and the mind 164, 194, 247, 270–3, 416–18
 as models, *see* non-mechanistic models
 as non-mechanistic approximation 148,
 170–1
 and non-mechanistic capabilities 110–11, 112,
 148, 164, 194–5, 417–18, 501, 504, 593–4
 and non-mechanistic phenomena 11–12,
 104–5, 108–9, 127, 204–5, 233–4, 238,
 248–9, 277–8, 861
 non-mechanistic qualities of 104–5, 148
 simplification of, *see* abstraction, fallacy of;
 mechanistic approximation;
 mechanistic fallacies; reduction: of
 complex to simple structures;
 reification, fallacy of
 and tacit knowledge 160–4
 and Wittgenstein's later philosophy 336–43,
 345–7
see also interactions, of structures;
 non-mechanistic phenomena
- complexity
 levels of, *see* levels
 in structures 93, 93 n. 1, 96, 99–100, 108–9,
 121, 636, 709 n. 2, 729 n. 1
- computer associations
 contributing to the software bureaucracy
 39–40
 promoting software mechanism 39–40
 supporting the elites 39–40
- conditional constructs, *see* software constructs,
 flow-control: conditional
- confirmations
 ease of finding 210, 216
 looking for
 as illogical method 210–11, 215–16, 220–3,
 249–50, 511, 674
 and old theories 222
 as popular method 218–19, 220–2, 674
 and software testing 223

- confirmations (*cont.*)
 sought by pseudoscientists 238–9, 249–50, 511, 514, 627
 used to deceive
 in advertising 220–1
 in the media 221–2, 627, 674, 822
see also falsification principle; falsifications
- conformism
 vs. freedom 853
 to a mechanistic ideology 853–4, 855, 857–8, 862, 868–9, 873
 to the mechanistic software ideology 199, 411–13, 437–8, 490–1, 842–3, 868–9, 874–5
 perceived as education 851, 854, 873
 as reflection of the general will 851, 853, 874
 replacing individuality 853, 857–8, 862, 873
- construction, emulated in software theories,
see manufacturing: emulated in software theories
- constructs, software, *see* software constructs, flow-control
- context-dependent knowledge
 and contexts 173
 and skill levels 158
- context-free knowledge
 and contexts 173
 and mechanistic capabilities 164, 188–9
 and skill levels 158
- contexts
 as complex structures 172–3, 184–5, 187, 188–9
 definition of 172–3
 failure to recognize 179–81, 184–5
 in humour 177–8, 187
 and language 122, 256–7, 259
 recognizing 184–5, 187, 188–9
- Copi, Irving M., and artificial languages 319
- corporations
 corruption in, *see* corruption: in corporations
 as elite 844, 846, 849–50, 856–8
 inefficiency in 37–8, 200, 487–9
 irresponsibility in 468–9, 487–9, 819–20
 promoting mechanistic concepts 844, 845, 856–7
 promoting totalitarian concepts 843–4, 846, 853, 857–8
 software, *see* software companies
- corpuscular theories
 as fantasies 75, 79
 and mechanism 78
 as metaphors 78
 similar to the ancient theories 74–5
- correspondence, one-to-one, *see* mapping
- corroborating evidence, failed tests as 219, 278
- corruption
 in corporations
 allowing dependence on the software elites 37, 468–9, 475–6, 487–9, 792, 816–17, 819–20, 821–2, 871–2
 deskilling workers 200, 295–6
 following pseudoscientific software theories 29, 483, 485, 487–9, 536, 793
 preventing programming expertise 29, 34–5, 41, 126, 197–9, 468–9, 794, 819–20, 822–3, 871
 suppressing software failures 793, 822–3
 in society
 apathy toward 818, 823, 872
 due to language abuse 382
 due to mechanism 439, 818, 859–61
 due to software totalitarianism 411–13, 490–1, 847–8, 871–2
 due to totalitarianism 846, 865–6, 869, 871
 in universities
 accepting blindly the mechanistic ideology 21, 129–30, 274–5, 513, 837, 854–5, 859–60
 assuming that minds are like machines 263, 270–3, 277–80, 282, 321–2, 405–6, 438, 835, 837, 838–9, 854
 causing totalitarianism 16, 854–5, 859–60
 degrading the concept of responsibility 835, 837–8
 promoting pseudoscientific notions 204, 273–5, 279, 513, 742, 794, 871
 pursuing pseudoscientific theories 77, 141, 202, 230–1, 235, 242–3, 251–2, 484–5, 794
 struggling to save fallacious theories 234–5, 264–6, 269, 523–4, 549, 651, 672
 studying non-mechanistic phenomena mechanistically 191, 204–5, 242, 244–6, 256–8, 273–5, 283–6, 712–13
 using stratagems to avoid falsifications 240–2, 248–50, 263–6, 513, 854
- cost
 of programming incompetence 32–3, 200, 416, 486, 490, 520, 820 n. 2
 of software mechanism 33, 126, 457–8
- craftsmanship
 definition of 497
 misrepresented by the software theorists 497–9, 505, 635–6, 675
 as non-mechanistic knowledge 497, 504, 505
 programming expertise as 505–6, 515

craftsmen
 programmers compared to 494, 497–9,
 505, 635–6, 675
see also craftsmanship

creativity
 in art 175–6
 as capacity for low levels 434–5
 as combining knowledge structures 174–6,
 177–8, 183–7
 and complex structures 175, 181, 182, 184–6
 degrading the concept of 277–8, 281–3,
 827–8, 831–2, 833–4, 835
 denied by mechanistic theories 278–80,
 438, 827–8
 denied by software theories 292–4, 296–7,
 842
 in humour 177–8
 as illogical behaviour 183–4
 and mental disorders 178–81, 185
 in metaphoric thinking 176
 misinterpreted as large number of
 alternatives 280, 281, 283–6, 288,
 292–4, 461–2, 828, 831–2
 and non-mechanistic capabilities 172, 174–6,
 177–8, 277–8, 434–5
 in normal behaviour 174, 178, 181–2
 in recognizing objects 183–4
 replacing with software devices 282, 293–4,
 296–7, 831–2

crisis, in software, *see* software crisis

cultural fashions, as myths 6–7

culture of knowledge substitutes 190, 874–5

current index, *see* file operations: current
 index in

current key, *see* file operations: current key in

current pointer, *see* file operations: current
 pointer in

cursor, in SQL 812

D

Dante Alighieri, and the idea of a perfect
 language 303

data fields, *see* fields: in data files

data files
 definition of 688
 fields in, *see* fields: in data files
 record keys in, *see* keys: in data files
 records in, *see* records: in data files
see also index files

data inconsistencies, *see* relational model: data
 redundancy and inconsistencies in

data integrity, *see* relational model: integrity
 functions in

data records, *see* records: in data files

data redundancy, *see* relational model: data
 redundancy and inconsistencies in

database fields, *see* fields: in data files

database operations
 definition of 686–7
 difficulty of 701–2, 703
 vs. file operations 687
 as high-level operations 678–9, 683, 687,
 702–3, 800–1

database structures
 definition of 687
 description of 702
 interacting with other software structures
 227–9, 702, 724, 733, 734–5, 737–8, 757,
 761, 768, 800
see also file relationships

database systems
 definition of 687
 and the delusion of high levels 683, 702–3,
 800–1
 early models of 678–9
 vs. file systems 687
 as frauds 678–9
 as management tools 678–9
 motivation for 677, 686
 and programming incompetence 678–9,
 705, 759–60, 792, 817
 as programming substitutes 678–80, 759–60,
 817
 promises of 679
see also relational model

databases
 and applications, integration of 705, 800, 811
 aspects of 780–2
 definition of 687
 design of
 based on field dependencies 761–3,
 768–9, 772, 773–4, 775–7, 778–9, 782–3,
 802
 based on file relationships 755–7, 762–3,
 772–3, 774–5, 776, 777–8, 781–2, 798
 as informal procedure 760, 762–3, 764–5,
 780, 781–3, 796–8
 as non-mechanistic phenomena 780–2
 relationships in, *see* file relationships
 as software principle 351–3
 as software processes 352, 361, 440–1, 702
 as structures 352, 361, 440–1, 702

Date, C. J.
 and denormalization 795–6
 and the first normal form 789–90
 and the mathematics of normalization 767
 and normalization 762, 774–6, 777–9
 and ordinary field dependencies 771
 and the relational model 723

- deception
 in advertising 22, 220–1, 845, 857, 867–8
 through language, *see* language: deception
 through
 in promoting ideologies 846, 865
 in selling 22, 845
 through software, *see* software charlatanism
- deduction
 and hierarchical structures 87–8, 310, 708
 and the problem of demarcation 213–14
 in refuting theories 215
- definitions, of software applications
 and CASE 479–82
 as mechanistic approximation 473–4, 477
 as mechanistic delusion 472–4, 477, 479–82
 requiring non-mechanistic knowledge
 480–1
- definitions, of structures 85, 92–3, 94–5, 554–5,
 636–7
- dehumanization, through totalitarianism
 19–20, 855
- delusions, mechanistic, *see* mechanistic
 delusions
- demarcation
 Popper's criterion of 214, 216–18
 principles of
 as concept, explanation of 210–11, 214,
 215–18, 223–5
 dishonesty toward 230–1
 generally ignored 218, 220–2, 230–2
 misinterpreted 230–2
 and philosophical subtleties 230–1
 used to expose pseudosciences 208, 231,
 512
 problem of 208, 209, 210–11
 and the problem of induction 211–12,
 213–14
- democracy
 liberal, *see* liberal democracy
 and progression toward totalitarianism
 858–9, 872
 totalitarian, *see* totalitarian democracy
- denormalization, *see* normalization, in the
 relational model: denormalization in
- dependence
 fostered by the software elites
 causing irresponsibility 486–90, 833
 and the delusion of high levels 449–52,
 462–3, 465–8, 628–30, 650, 675–6, 705
 due to programming incompetence
 416, 475–6, 626, 675–6, 792, 816–17,
 819–20
 preventing non-mechanistic knowledge
 291–2, 423–5, 427–8, 436, 493, 870–1,
 874–5
- dependence (*cont.*)
 fostered by the software elites (*cont.*)
 and the promise to solve non-mechanistic
 problems 195–7, 420, 421
 on knowledge substitutes
 causing irresponsibility 487–90
 vs. non-mechanistic capabilities 190
 preventing non-mechanistic knowledge
 200–1, 419–20, 427–8, 874–5
 and the promise to solve non-mechanistic
 problems 65–7, 195–7, 295–6
 on the mechanistic myth 25, 26
 on programming substitutes
 causing irresponsibility 486–7
 and the delusion of high levels 456–7,
 465–9, 626, 628–30, 650, 675–6, 705
 due to programming incompetence 32–3,
 38–9, 54–5, 198–9, 294–5, 411–13, 415,
 819–20
 on ready-made products 856, 858
 on software 298–9, 409–10, 411–13, 432,
 437–8
 on software devices
 causing irresponsibility 487–90, 833,
 834–5, 839
 and the delusion of high levels 449–52,
 462–3
 vs. non-mechanistic capabilities 420–1,
 437–8
 preventing non-mechanistic knowledge
 200, 423–5, 436, 439, 874–5
 and the promise to solve non-mechanistic
 problems 195–7, 422–3
- dependencies, field, *see* relational model: field
 dependencies in
- dependency theory, *see* normalization, in the
 relational model
- Descartes, René
 and artificial languages 313
 and hierarchical structures 88, 89, 143–4,
 208, 306–7
 and language structures 310
 and material world vs. mental world 306–7
- deskilling
 of managers and workers 200, 295–6
 of programmers 34–5, 35 n. 2, 486–7, 493,
 497–500, 498 n. 13, 618–21, 625–6, 675–6
- determinism
 and academic fashion 827–8
 definition of 824
 vs. free will, and responsibility 823–4, 827–8,
 833–4, 836–7, 840–3
 and hierarchical structures 99, 277–8
 Laplacean 72, 569, 824–5
 and mathematics 89

- determinism (*cont.*)
 and mechanism, *see* mechanism: and
 determinism
 and mechanistic approximation 826, 828–31
 naivety of 72
 in the new pseudosciences 277–8, 281–2,
 283–6, 828
 and non-mechanistic phenomena 91, 234,
 283–6, 826–7
 and scientific theories 91–2
 and sociology 840–1
 and software theories 288, 292–4, 831–2,
 841–3
see also indeterminism
- deterministic models, *see* mechanistic models
- deterministic phenomena, *see* mechanistic
 phenomena
- deterministic systems
 human beings perceived as
 in the new pseudosciences 181–3, 277–8,
 437–8, 827–8, 833–4, 854–5
 in software-related affairs 834–5
 under totalitarianism 836, 852, 861–2
 societies perceived as
 in the new pseudosciences 833–4, 854–5
 under totalitarianism 852
- development environments
 as low levels in high-level systems
 dishonest nature of 449–52, 462–3, 464–8
 and object-oriented programming
 628–30, 648–9, 650, 675–6
 and the relational model 676–7, 685,
 704, 800–1
 and structured programming 626
 reversing the principles of programming
 462, 465, 467–8, 629–30, 648–9, 685
see also software devices
- devices, software, *see* software devices
- die rolling
 as analogy of human phenomena 829–30
 as non-mechanistic phenomenon 829–30
- Dijksterhuis, E. J.
 and corpuscular theories 75, 79
 and mechanism 71, 79, 88, 89
- Dijkstra, E. W.
 and the GOTO debate 522–3, 608, 620
 and mathematical programming 501–2
 and structures, static vs. dynamic 546
- doctrine, mechanistic, *see* mechanism
- dogma
 mechanism as 7, 26, 91, 109, 233–5, 274–5,
 503
 software mechanism as 26, 503, 504
- domination, through software, *see* software
 domination
- doublespeak
 definition of 382
 examples of 382–4
- Dreyfus, Hubert L., and the philosophy of
 artificial intelligence 146 n. 7
- Dreyfus, Hubert L. and Stuart E.
 and artificial intelligence 146
 and mind substitutes 190
 and skill acquisition 157–60
- dynamic sequence, *see* sequence: of operations,
 static vs. dynamic
- dynamic structure, *see* structures: static vs.
 dynamic
- ## E
- ease of use, *see* software: easy to use
- Eco, Umberto, and the idea of a perfect
 language 302–4, 309
- economics
 and reification 861
 and scientism 133, 135–6, 861
- economy
 centrally-planned, *see* centralized economy
 as non-mechanistic phenomenon 861
- education
 conformism perceived as 851, 854, 873
 as mechanistic indoctrination 16, 845, 855
 of programmers 32, 197, 414–15, 497, 505–6,
 619, 621
- Einstein's theory of relativity, and mechanistic
 approximations 106
- elements, of structure
 abstract concepts as 114
 atomic and independent 155–6, 285–6,
 287–8, 296, 311–12, 315–16, 323
 classes as, in object-oriented programming
 637–8
 concrete things as 114
 definition of 92–3, 99 n. 1
 objects as, in object-oriented programming
 637–8
 shared by several structures 99, 105, 446–8,
 469, 628
 starting 93, 95, 554
 terminal 92–3
 top 92–3, 95, 554
- Eliade, Mircea
 and mana 57, 61
 and modern myths 3–4, 5–7
 and myths 2–3
- elite
 bad 15–16, 23–4
 controlling language in imaginary society
 366, 429–31, 468

elite (*cont.*)

- corporations as 844, 846, 849–50, 856–8
 - good 15, 23–4
 - justified by a mechanistic ideology 844, 865
 - manipulating language 124, 370, 398, 399–400, 406–7, 408–9, 863–4, 865, 870–1
 - and myths 15–16, 23–4
 - promoting language mechanism 251, 838–9, 870
 - role of, under totalitarianism 30, 490–1
 - software, *see* software elites
 - in totalitarian democracy 849, 851–2
 - universities as 844, 846, 849–50, 854–5, 859–60
 - weakness of 847
 - see also* corporations; universities
- empiricism
- explaining the world through 209–10, 216
 - and mind models 280–2
 - in Popper's criterion of demarcation 216
 - and the problem of induction 213
- engineering
- emulated in software theories 86, 494–6, 503–4, 506–7, 508–9, 633
 - and hierarchical structures 86–7
 - misinterpreted by the software theorists 86, 506–7
 - non-mechanistic aspects of 506–7
- entities, of structure, definition of 99 n. 1
- environment, natural, destruction of 19–20
- environment, software, *see* development environments
- Epicureanism, as atomistic philosophy 74
- ethics, and the idea of responsibility 823–4
- Euclid's geometry, based on hierarchical structures 87, 89
- evolution, and non-mechanistic capabilities 142, 154, 271
- experience, personal, *see* personal experience
- expert systems, as mechanistic delusion 435
- expertise
- characteristics of 157–60, 272
 - degrading the concept of 274–5, 416, 431, 436, 835, 845, 868
 - due to intuitive knowledge 157–60, 168–9, 432
 - in natural skills 159–60, 271
 - prevention of 65–7, 196–7, 411–13, 419–20, 487–90, 493
 - in programming, *see* programming expertise
 - substitutes for, *see* knowledge substitutes; programming substitutes
- experts
- as charlatans 22

experts (*cont.*)

- in myth-related services 15
- in software, *see* software theorists
- explanation, mechanistic, *see* mechanistic explanation
- exploitation, through software, *see* software domination

F

- face recognition, as non-mechanistic phenomenon 110–11, 119–20, 161
- failure, due to mechanism
- in artificial intelligence 145–6, 183–6
 - of knowledge substitutes 27, 143, 160, 200–1, 296
 - of mechanistic theories, *see* mechanistic theories: failure of
 - of programming substitutes 27, 146–7, 160, 198, 294–5, 821–2
 - of software applications, *see* software failures
 - of software theories, *see* software theories: failure of
- fallacies, mechanistic, *see* mechanistic fallacies
- false assumption
- causing invalid theories 205
 - in magic 49–50, 52
 - in myths 14
 - in pseudosciences 205, 512
 - in software theories 32, 49–50, 52
- falsifiability
- as basis of testability 216–18
 - lacking in pseudoscientific theories 217–18, 224–6, 234–5
 - as misunderstood concept 218, 231–2
 - as quality of scientific theories 216–18, 269
 - vs. verifiability, as criterion of demarcation 214, 216–17, 324
- falsification principle
- vs. confirmations 215–17, 238–9, 249–50, 514, 627, 674
 - vs. defending the theory 215–16, 234–5, 269, 328 n. 1, 511
 - definition of 214
 - generally ignored 218, 220–2
 - and old theories 222
 - used in decision-making 220–2
 - used in testing software 223
 - used to expose deception in advertising 220–1
- falsifications
- avoiding, stratagems for 224–6, 238–9, 248–50, 263–6, 344, 672, 854
 - ignored by pseudoscientists 221, 239, 250, 511, 514, 627

- falsifications (*cont.*)
 ignoring, circularity of 512
 looking for
 as logical method 211, 215–18, 220–3, 511, 674
 and old theories 222
 and software testing 223
 suppressed by expanding the theory
 in behaviourism 240–2
 in Chomskyan linguistics 259–60, 263–6, 269
 in the new pseudosciences 234–5, 344, 513, 727–8, 827
 in object-oriented programming, *see* object-oriented programming:
 expanded in order to suppress falsifications
 in the relational model, *see* relational model:
 expanded in order to suppress falsifications
 in software devices 451, 626, 801
 in software theories 514, 626
 as stratagem, explanation of 224–7, 511
 in structured programming, *see* structured programming:
 expanded in order to suppress falsifications
 turning into features, *see* falsifications:
 suppressed by expanding the theory
see also confirmations; falsification principle
 family resemblance, idea of, in Wittgenstein's later philosophy 337–8, 342–3
 field dependencies, *see* relational model: field dependencies in
 field relationships, and file relationships 754–6
 fields
 in data files
 alphanumeric, definition of 690
 binary, definition of 690
 date, definition of 690
 definition of 688
 numeric, definition of 690
 relating files through 697–700, 751–2, 753–6, 757–8, 768–9
 used as record key 688–9, 693–700, 701, 753–4
 in the relational model, *see* relational model:
 fields (columns, attributes) in
 file maintenance
 as low-level operations 445–8, 683
 using the basic file operations 683, 694
 file operations
 for accessing individual records 693–4
 for adding records 693
 and applications, integration of 705, 800, 811
 basic, definition of 686, 691–2
 file operations (*cont.*)
 as correct level of abstraction 703–4, 705
 current index in
 definition of 691
 uses of 691–2
 current key in
 definition of 691
 uses of 691–2
 current pointer in
 definition of 691
 uses of 692
 for data analysis 696–7
 vs. database operations 687
 definition of 686
 DELETE
 definition of 692
 vs. SQL operations 811, 812
 uses of 693, 696
 for deleting records 693, 696
 for displaying fields 693
 for file maintenance 683, 694
 and file scanning loops 694–700, 701, 703
 flexibility of 704
 and flow-control constructs 690, 694–7, 698–9, 704, 705, 738, 813
 as high-level operations 684
 and interacting structures 702
 as low-level operations 684, 687, 702–4, 800
 for modifying records 693, 696
 as necessary and sufficient operations 686, 693, 703, 800, 811
 and nesting of file scanning loops 697–700, 701, 788, 793
 pointer in
 definition of 691, 812
 uses of 692
 for queries and reports 696, 738
 READ
 definition of 692
 vs. SQL operations 811
 uses of 693–4, 697, 700, 701
 READ NEXT
 definition of 692
 vs. SQL operations 811, 812, 814
 uses of 694, 697, 698–9
 record area in
 definition of 691
 uses of 691–2
 and record selection 694–9, 701, 736
 for relating files 697–700
 requiring expertise 677, 686
 REWRITE
 definition of 691
 vs. SQL operations 811, 812
 uses of 693–4, 696, 697

file operations (*cont.*)

- for scanning records 694-7
- simplicity of 701, 703
- for sorting records 696
- START
 - definition of 692
 - vs. SQL operations 811, 812
 - uses of 694, 697, 698-9
- as traditional operations 687, 705, 811

WRITE

- definition of 691
- vs. SQL operations 811
- uses of 693-4, 696, 697

file relationships

- combinations of 754
- as complex structures 352 n. 3, 702, 761
- database design based on 755-7, 762-3, 772-3, 774-5, 776, 777-8, 781-2, 798
- and field relationships 754-6
- through field values 697-700, 701, 751-2, 753-6, 757-8, 768-9
- through file nesting 697-700, 701, 753-4, 788
- as hierarchical structures 702, 754
- many-to-many
 - definition of 700, 753
 - examples of 700-1, 753-4, 772-3, 777-8
- many-to-one
 - definition of 700, 753
 - examples of 753, 755-6, 768
- one-to-many
 - definition of 700, 753
 - examples of 698-9, 753, 755-6, 774-5, 776
- one-to-one
 - definition of 700, 752
 - examples of 752-3, 755-6, 768, 772-3, 774-5
- as reflection of field value restrictions 757-9
- as reflection of the application's
 - requirements 756-8, 759, 761-3, 764, 768-9, 772-9, 780-1, 797-8

file scanning loops

- as correct level of abstraction 703, 810, 811
- definition of 694-5
- examples of 694-7, 698-9, 701
- implicit, in the relational model 718, 788, 810-13
- nesting of 697-700, 701, 788, 793
- in SQL 810-13, 814, 815-16

file systems

- vs. database systems 687
- definition of 687

files

- data, *see* data files

files (*cont.*)

- index, *see* index files
- in the relational model, *see* relational model:
 - files (tables, relations) in
- relative 688-9
- sequential 688-9
- service, *see* service files
- see also* file operations; file relationships
- Fleming, Candace C.
 - and denormalization 795
 - and the first normal form 790
 - and the relational model 723, 741
- flow-control, as software principle 543
- flow-control attributes
 - and flow-control constructs 594-5
 - and flow-control structures 544, 547-8, 558, 560-1, 593-4
 - and the GOTO statement 543
 - and jumps 543-4, 553, 564-5, 567, 593, 594-5
 - multiple 544, 558-9, 564-5, 593-4
 - and the nesting scheme 545, 558, 560-1, 564-5, 594
- flow-control constructs, *see* software constructs, flow-control
- flow-control features
 - low-level vs. high-level 518
 - as software processes 543
- flow-control structures
 - as complex structures 548, 552, 553, 556-9, 560-1, 562-3, 567-8, 590, 591-2, 597-8, 666
 - and the delusion of the main structure 363-4, 529, 544-5, 550-2
 - and flow-control attributes 544, 547-8, 558, 560-1, 593-4
 - multiple 553-4, 556-9, 590-1, 593-4, 602-3, 663, 666
 - perceived as one structure 529, 544-5, 547-8, 552, 567-8, 663
 - and software processes 559-60
 - as structures in time, vs. structures in space 546, 547, 554, 556
 - as variable structures, vs. fixed structures 555, 558
- flow diagrams
 - examples of 527, 538-41, 573
 - ignored in object-oriented programming 663, 665
 - reflecting the nesting scheme 526-8, 545, 558, 591, 663
 - structured 528, 537-41, 577, 587
- flow of execution
 - definition of 517
 - jumps in, *see* jumps, in the flow of execution

flow of execution (*cont.*)
 in object-oriented programming, *see*
 object-oriented programming: flow
 of execution in
 and programming expertise 518–19, 567–8,
 619
 in structured programming, *see* structured
 programming: flow of execution in
 flowcharts
 vs. classification-style diagrams 364
 and the delusion of the main structure
 345, 364
see also flow diagrams
 formal logic, *see* logic systems
 formal reductionism 9, 76, 82–4
 fourth-generation languages (4GL)
 as frauds 464–5, 704–5
 as low levels in high-level systems 464–5,
 704
 fraud, in research, *see* corruption: in universities
 free will
 degrading the concept of 277–8, 827–8,
 833–4, 835
 denied by deterministic social theories
 840–1
 denied by mechanistic theories 278–80,
 438, 827–8
 denied by software theories 842
 vs. determinism, and responsibility 823–4,
 827–8, 833–4, 836–7, 840–3
 misinterpreted as large number of
 alternatives 280, 281, 283–6, 288, 828
 and non-mechanistic capabilities 277–8
 freedom
 vs. conformism 853
 defending 422
 degrading the concept of 281–3, 853
 in software, *see* software freedom
see also free will
 Frege, Gottlob, and the idea of a perfect
 language 320
 French Revolution
 and totalitarian democracy 849
 and utopianism 132
 Freudianism
 as myth 4–5
 as pseudoscience 210, 224
 Frith, Uta, and contexts 180–1, 185
 functions, *see* subroutines

G

games, idea of, in Wittgenstein's later
 philosophy 337–41, 342–3, 346–7
 Gardner, Howard, and structuralism 249, 250

general will
 conformism as reflection of 851, 853, 874
 as idea, circularity of 851, 852
 as natural human qualities 851, 874
 generations, of programming languages 464–5,
 503
 generative grammar
 definition of 251
see also Chomskyan linguistics
 geometrical method, and hierarchical
 structures 88, 89, 208
 geometry, as hierarchical structures 87
 Ghost in the Machine, concept of 149
 GOTO statement, *see* structured programming:
 GOTO statement in
 grammar
 and Chomskyan linguistics 251, 257–8,
 260–1, 262–3
 formal, in computer science 261–2
 as hierarchical structures 112, 262
 Greenbaum, Joan M., and the deskilling of
 programmers 35 n. 2, 498 n. 13
 growth, delusions of 22–3
 gurus, *see* software theorists

H

Harris, Roy, and language mechanism 838–9
 Hayek, F. A.
 and central economic planning 862 n. 6
 and scientism 129, 132
 hierarchical metaphor
 and mathematics 88
 and mechanism 86
 hierarchical structures
 and abstraction 95–6, 636–7, 651–2
 in artificial intelligence 145–6
 as basis of mechanism 8–9, 69
 as basis of software theories, *see*
 software structures: perceived as
 simple structures
 as brain process 271–2
 and classification systems 8, 93–4, 98–9,
 100, 103–4, 643–4
 definition of 85, 92–3, 94–5, 554–5, 636–7
 embodying reductionism and atomism 85–6
 and engineering 86–7
 examples of 8–9, 93–4, 95–6, 98, 100, 153–4,
 554, 643–4
 and the geometrical method 88, 89, 208
 geometry as 87
 grammar as 112, 262
 incorrect, when depicting multiple attributes
 101–2, 338–41, 356–9, 360–2, 659
 language as 260, 262, 300–2, 309, 640–1

- hierarchical structures (*cont.*)
 logic systems as 708, 732
 and logical deduction 87–8, 310, 708
 and manufacturing 86–7, 359
 mathematics as 9–10, 87, 309–10, 728–9,
 735–6
 as mechanistic models 85–6, 90–1, 93, 108,
 127, 277–8, 283, 828–31
 as mind models, *see* mind models: based
 on hierarchical structures
 and object-oriented programming 636–7,
 642–3, 645–6, 652–4, 658–9, 666
 phenomena perceived as 9–10, 80
 software perceived as, *see* software struc-
 tures: perceived as simple structures
 and structured programming 525–8, 547–8,
 550–2, 561–2, 580–1, 618
see also simple structures
- holistic knowledge, and skill levels 157–60, 272
- homunculus fallacy
 definition of 150
 and mechanism 151–2, 153
 and mind models 150–2
 and the phenomenon of vision 150–1
- human beings
 perceived as deterministic systems
 in the new pseudosciences 181–3, 277–8,
 437–8, 827–8, 833–4, 854–5
 in software-related affairs 834–5
 under totalitarianism 836, 852, 861–2
 perceived as operators of software devices
 486–7, 489, 490, 493, 500
- human phenomena
 indeterministic nature of 10, 21, 89, 127–8,
 277–8, 283–7, 713, 828, 833–4
 as interacting structures 128, 133, 161–2,
 207, 277–8, 308
 software phenomena as 10, 89, 127, 289–90,
 501
- human potential, defending 422
- Hume, David, and the problem of induction
 213
- humour
 and creativity 177–8
 and the GOTO debate 614–15
 as interacting structures 177–8, 186–7
 as a means to mask irrationality 614, 616
- hypothesis, false, *see* false assumption
- I
- iatromechanics, and scientism 136–7
- IBM Corporation
 and nested relational databases 792–4
 and software irresponsibility 818–19
- ideal language, *see* perfect language
- ideologies
 academic, totalitarian nature of 846–7,
 849–50, 854–5
 business, totalitarian nature of 846–7,
 849–50, 856–8
 mechanistic, conforming to 853–4, 855,
 857–8, 862, 868–9, 873
 mechanistic software, conforming to 199,
 411–13, 437–8, 490–1, 842–3, 868–9,
 874–5
 political, and mind models 280–2
 promoted through deception 846, 865
- IEEE Computer Society, *see* computer
 associations
- ignorance
 causing primitive beliefs 42, 43–4, 59–60,
 819
 causing the software bureaucracy 29, 31–2,
 421, 818–19
 causing the software myth 28, 31–2
 fostered by the software elites 65–7, 291, 422,
 423–5, 427–9, 489–90, 493, 500, 819–20
 promoted through propaganda 66–7
- immunizing stratagems, as tricks to avoid
 falsifications 225, 226–7
- impoverishment
 due to abstraction 97, 114–16
 due to both mechanistic fallacies 122–4,
 285, 296, 418, 831
 due to reification 120–1, 257–8, 259, 418
- of knowledge
 through acronyms 373–4, 395–6, 403–4,
 864
 through language mechanism 124–5,
 366, 400–1, 402–5, 429–30, 870
 through slogans 372–3, 385–6, 394, 395–6,
 398, 863–4
 through software mechanism 367, 847–8,
 871
- of language 124, 292, 366, 399–401, 402–5,
 406–7, 429–31, 468, 863–4, 870
- of software 126, 290–2, 367, 832, 867–8, 870–1
- of software-related affairs 115–16, 126, 291–2,
 296
- incompetence
 in programming, *see* programming
 incompetence
 in software-related affairs 31–2, 41, 427–8
 spread of 41, 823
- inconsistencies, data, *see* relational model: data
 redundancy and inconsistencies in
- indeterminism
 attempting to reconcile with mechanism
 825–6, 827–8, 833–4

- indeterminism (*cont.*)
 degrading the concept of 827–8, 829–32, 833–4
 denied by deterministic social theories 840–1
 in human phenomena, denied by
 mechanistic theories 10, 21, 89, 127–8, 277–8, 283–7, 713, 827–8, 833–4
 misinterpreted as large number of alternatives 828–31
 and reification 121
 in software phenomena
 denied by software theories 89–90, 127, 287, 288–9, 292–4
 replacing with determinism 292–5, 833–5
see also determinism
- indeterministic models, *see* non-mechanistic models
- indeterministic phenomena, *see* non-mechanistic phenomena
- index files
 and the current index 691
 definition of 688–9
 multiple 689, 701, 754
 used to define a sorting sequence 689, 696
 used to identify data records 688–9, 693–4, 700
 used to scan data records 688–9, 694–700, 701
see also data files
- indexes, *see* index files
- individuality, replacing with conformism 853, 857–8, 862, 873
- indoctrination, mechanistic, *see* mechanistic indoctrination
- induction
 based on regularities 211–12
 as basis of pseudosciences and superstitions 211–12
 as basis of science 211–12
 and chicken's quandary 212–13
 in discovering theories 215
 and fallacious decision-making 212–13
 fallacy of 211–12
 problem of 211–12
 and the problem of demarcation 211–12, 213–14
- inefficiency
 in corporations 37–8, 200, 487–9
 in programming 32–3, 126, 199, 483–4, 486
 in society 30, 38–9, 483, 490–1, 504
- informal reductionism 76
- information technology
 as abstract term 395, 398
 based on the mechanistic myth 73
- information technology (*cont.*)
 definition of term 395
 misuse of term
 through the acronym "IT" 395–6
 analysis of 395–6, 397–8
 compared with other fields 397–8
 examples of 395–6, 397
see also technology: misuse of term
 term used as slogan 395–6, 397
- inheritance, in object-oriented programming,
see object-oriented programming:
 inheritance in
- innate faculties, idea of
 arguments against 170, 270–3
 and linguistics 76, 170, 252–5, 269–73, 280–1, 303, 838–9
 and the new pseudosciences 284–5, 828
- integrity functions, *see* relational model:
 integrity functions in
- Intel Corporation, and the term "technology" 388–90
- intelligence
 artificial, *see* artificial intelligence
 attempting to represent mechanistically 170–1
 and intuitive knowledge 144
 and *knowing how* vs. *knowing that* 148–9, 164
 and non-mechanistic knowledge 142, 145, 181–3, 194
- interactions, of language structures
 and Chomskyan linguistics 268, 272–3
 and deception 371
 and the idea of a perfect language 305, 308–9, 641
 impaired in schizophrenia 179–80
 with knowledge structures
 and Chomskyan linguistics 256–7, 258–9, 266
 and deception 371
 and the idea of a perfect language 305, 308–9
 and Newspeak 124–5, 401, 403–5
 and non-mechanistic knowledge 122, 166–7
 and the rationalist fallacies 311–12
 and similarity to software 292, 300–2, 326–7, 351, 366, 368–9, 429–30
 and software totalitarianism 870
 and Wittgenstein's later philosophy 346
 and Newspeak 401
 and the rationalist fallacies 311–12
 and similarity to software 326–7, 351
 in stories 350–1, 357–9, 401, 442–3
 and structuralism 244

- interactions, of language structures (*cont.*)
 and totalitarianism 863–4
 and Wittgenstein's later philosophy 342, 345–6
 and word meanings 112, 350–1
- interactions, of software structures
 and the delusion of mathematical programming 89–90, 597–9, 600, 681–2
 and the delusion of the main structure 345–6, 350, 363–4
 and file operations 702
 and flow-control constructs 559–60, 593–6, 597–8, 666
 ignored by programming aids 125–6, 193, 364–5
 ignored by software devices 296, 832
 ignored by software theories 12, 89–90, 363–5, 472–4, 514, 550–2
 with knowledge structures
 ignored by programming aids 125–6
 ignored by software devices 296
 and similarity to language 300–2, 305, 326–7, 367, 368–9
 and software totalitarianism 847, 871
 and the need for low levels 445–8, 469, 628–30, 648–9, 683, 703–4, 705, 724, 737–8, 800, 803–4
 and object-oriented programming 642–3, 648–50, 652, 653–4, 659, 662
 and the relational model 227–9, 724, 734–5, 737–8, 761
 requiring programming expertise 54, 121, 193, 294–5, 365, 415, 419, 441–3, 568, 596–7, 702
 and similarity to language 326–7, 442–3, 551–2, 628–9
 and software processes 347–8, 350, 352, 355, 360–3, 365, 440–1, 542, 559–60, 702
 and software totalitarianism 867, 871
 and structured programming 521, 537, 567–8, 591–2, 593–7
 and Wittgenstein's later philosophy 345–7, 360–3
- interactions, of structures
 causing complexity 87, 105
 in complex structures 90, 99, 116–19, 122, 164–5, 173, 642, 643–4, 648–9, 652
 and creativity 175, 177–8, 183–6
 and distinguishing chicks 169
 and face recognition 111, 147–8
 in human phenomena 128, 133, 161–2, 207, 277–8, 308
 and humour 177–8, 186–7
 impaired in autism 180–1, 185
 impaired in schizophrenia 179–80, 185
- interactions, of structures (*cont.*)
 in language, *see* interactions, of language structures
 loss of 120–1, 123, 205
 and mechanistic mind models 154, 156–7, 246–7
 and non-mechanistic knowledge 148, 161–2, 165–7, 316, 323, 417–18, 652, 832–3
 and non-mechanistic mind models 170–1, 181, 270–3
 in software, *see* interactions, of software structures
 and tacit knowledge 161–2, 164–5
 in a three-body system 109–10
 and voice recognition 147–8
see also complex structures
- intuition, *see* intuitive knowledge
- intuitive knowledge
 definition of 159
 and intelligence 144
 as manifestation of tacit knowledge 171
 and non-mechanistic capabilities 164, 271–3, 495
 required for expert performance 157–60, 168–9, 432
 and skill levels 157–60
- irrationality
 blending with rationality 12–13, 46, 49–51, 53–6
 due to software mechanism 31–2, 53–4, 614
 in looking for confirmations 220–2
 masked through humour 614, 616
- irresponsibility
 causing software failures 486, 821–2
 in corporations 468–9, 487–9, 819–20
 due to language mechanism 468, 838–9
 due to mechanism 40–1, 274–5, 818, 834, 836–8
 due to software devices 468, 486–9, 839
 due to software mechanism 40–1, 818, 834–5, 872
 due to the software myth 29, 410, 839
 fostered by the software elites 468, 620–1, 624–5, 819–20
 justified by the idea of social forces 841, 843
 justified by the idea of software forces 842–3
 in programming 486–7, 498, 620–1, 624–5, 822–3
 in software-related affairs 41, 834–5, 839, 842
 spread of 40–1, 823
see also responsibility
- isomorphism, *see* mapping
- IT, *see* information technology
- iterative constructs, *see* software constructs,
 flow-control: iterative

J

- Jacopini, Giuseppe, *see* Böhm and Jacopini, work of
- Jensen, Randall W.
 and Böhm and Jacopini's work 575
 and the GOTO debate 612
 and non-standard constructs 589
 and software engineering 496
- jokes, *see* humour
- jumps, in the flow of execution
 definition of 517–18
 explicit
 vs. implicit 520, 529, 543–4, 564–5, 567–8, 591, 602–3, 616
 necessary 568, 586
 prohibited 538–41, 565
 and systems of jumps and labels 622–5
 using GOTO 518, 538–41, 543, 577, 602–3, 621
 and flow-control attributes 543–4, 553, 564–5, 567, 593, 594–5
 implicit, in flow-control constructs 520, 529, 543–4, 553, 558, 564–5
 overriding the nesting scheme 547–8, 553, 558, 564–5, 590–1, 593, 602

K

- Kernighan, Brian W., and the GOTO debate 610
- keys
 in data files
 ascending vs. descending 689
 and the current key 691
 definition of 688–9
 duplicate 689
 in the relational model 714–15
- knowing how*
 and complex structures 164
 definition of term 148–9
- knowing that*
 definition of term 148–9
 and simple structures 164
- knowledge
 and abstraction 155
 approximating with simple structures 165, 418
 and artificial languages 314–15
 as complex structures 148, 154–5, 164, 165–7, 176, 270–3, 315–17, 323, 417–18, 641, 651–2
 degrading, *see* impoverishment: of knowledge
 developed through experience, *see* personal experience

- knowledge (*cont.*)
 and the fallacy of abstraction 155, 418
 growth of, through trial and error 208–9, 218
 impoverishment of, *see* impoverishment: of knowledge
 intuitive, *see* intuitive knowledge
 mechanistic, *see* mechanistic knowledge
 new, uncertainty of 206–7
 non-mechanistic, *see* non-mechanistic knowledge
 perceived as simple structures 143, 315–17, 322–3, 641, 651–2
 and reification 121, 154–5, 301, 315–16, 371, 400–1, 418
 tacit, *see* tacit knowledge
 unspecifiable, *see* tacit knowledge
- knowledge structures, interacting with
 language structures, *see* interactions, of language structures: with knowledge structures
- knowledge substitutes
 as culture 190, 874–5
 dependence on, *see* dependence: on knowledge substitutes
 failure of 27, 143, 160, 200–1, 296
 and the idea of a perfect society 17
 as mechanistic delusions
 causing irresponsibility 487–9
 and the delusion of high levels 295–6, 433, 453–5, 461–2, 831–2
 justified by the software myth 26–7, 493
 vs. non-mechanistic capabilities 143, 417–18, 459, 832–3
 preventing non-mechanistic knowledge 419–20, 423–5, 427–8
 promising to solve non-mechanistic problems 65–7, 295–6
 mind models as 143, 146–7, 188–9, 417–18
 perceived as software power 62, 819
 promoted by the elites
 causing irresponsibility 487–90
 and the delusion of high levels 434–5, 449–52
 justified by the software myth 41
 vs. non-mechanistic capabilities 190, 296–7, 532–3
 preventing non-mechanistic knowledge 419–20, 427–8, 874–5
 promising to solve non-mechanistic problems 65–7, 195–7
 and skill levels 160
 spread of 41, 200–1, 295–6
- Knuth, Donald E.
 and the GOTO debate 609–10, 613, 615
 and non-standard constructs 592 n. 17

Koestler, Arthur
 and behaviourism 239, 240
 and creativity 177, 187

Kraft, Philip, and the deskilling of
 programmers 35 n. 2, 498 n. 13

Kurtz, Paul
 and myths 7
 and pseudoscience 203

L

labels, in the flow of execution
 definition of 518, 622–3
 and systems of jumps and labels 622–5

language
 artificial, *see* artificial languages
 aspects of 256
 and contexts 122, 256–7, 259
 controlled by the elite in imaginary society
 366, 429–31, 468
 deception through
 by coining new terms 369
 corrupting society 382
 and doublespeak 382–4
 due to language mechanism 369–70
 examples of 373–4, 375, 376–9, 380–4
 as a form of violence 377
 preventing 385
 principles of 372–5, 376–82, 384–5
 by restricting alternatives 371–2, 384,
 385, 394, 398
 by turning words into abstract terms
 374–5, 376–9, 383–4
 by using abstract terms as disclaimers
 376–82
 by using acronyms 373–4, 395–6, 864
 by using slogans 372–3, 385–6, 394, 395–6,
 398, 863–4
 deficiency in autism 180–1
 deficiency in schizophrenia 180
 degraded under totalitarianism 124–5,
 405–7, 863, 869–70
 emulating mathematics 309–12
 formal, in computer science 261–2
 ideal, *see* perfect language
 in imaginary society 366, 429–31, 468
 impoverishment of 124, 292, 366, 399–401,
 402–5, 406–7, 429–31, 468, 863–4, 870
 as innate faculty 76, 252–5, 269–70, 303,
 838–9
 as innate faculty, arguments against 170,
 270–3
 as interacting structures, *see* interactions,
 of language structures
 as machine in the mind 838–9

language (*cont.*)
 manipulated by the elites 124, 370, 398,
 399–400, 406–7, 408–9, 863–4, 865,
 870–1
 as means to represent the world 299–300,
 368–9, 408–9, 870
 mechanistic, *see* mechanistic language
 and the mechanistic fallacies, *see*
 mechanistic fallacies: in language
 mechanistic theories of 76, 112, 244, 251,
 260–1, 268, 269–70, 308–9, 322–3,
 330–3, 838–9
 and non-mechanistic knowledge 122, 166–7,
 256–7, 258–9, 270–3, 406, 411–12, 865
 as non-mechanistic phenomenon
 and Chomskyan linguistics 268, 270–3
 and the idea of a perfect language 304–5,
 308–9
 and Newspeak 124–5
 and the rationalist fallacies 311–12
 and reification 120, 122, 244
 and similarity to software 326–7, 628–9
 and Wittgenstein's later philosophy
 336–7, 343–4
 and word meanings 12, 112
 as organ, in Chomskyan linguistics 254,
 265, 269–70
 perceived as mapping system 300, 459,
 640–1
 perfect, *see* perfect language
 programming, *see* programming languages
 and the rationalist fallacies 308–12, 313–18,
 651–2
 and responsibility 252, 838–9
 similar to software, *see* software: similar to
 language
 and the structuralist phoneme theory 244
 and Swift's language machine
 fallacies of 458–9
 mechanistic basis of 318, 458–9
 and similarity to programming
 substitutes 318, 459–61
 software version of 460–1
 in Wittgenstein's later philosophy 336–7,
 341–2, 343–4
see also linguistics

language games, idea of, in Wittgenstein's
 later philosophy 342, 343

language mechanism
 and analytic philosophy 319–21
 causing irresponsibility 468, 838–9
 circularity of 320
 degrading minds through 124–5, 366, 399–
 401, 404–5, 406–7, 408–9, 429–30, 863,
 865, 869–70

- language mechanism (*cont.*)
 impoverishing knowledge 124–5, 366,
 400–1, 402–5, 429–30, 870
 and mind control 124–5, 366, 399–401,
 404–5, 406–7, 408–9, 429–30, 863,
 865, 869–70
 promoted by the elites 251, 838–9, 870
 in the seventeenth century 308–12, 313–15,
 317–18, 458–9, 651–2
- language structures
 interacting, *see* interactions, of language
 structures
 perceived as simple structures 260, 262,
 300–2, 309, 640–1
- Laplacean determinism 72, 569, 824–5
- Leach, Edmund, and binary operations 247,
 249
- learning
 and context-dependent knowledge 158
 and context-free knowledge 158
 through generic mental processes 170, 271–2
 and holistic knowledge 157–60
 of interacting concepts 165–7
 and intuitive knowledge 157–60
 and personal involvement 158–9
 process of 157–60
 and progression from novice to expert
 157–60, 416–17
 and rule-following 157–8, 159, 272
 and skill levels 157–60
see also knowledge
- Leibniz, Gottfried
 and the alphabet of thought 314–15, 316–17
 and artificial languages 314–15
 and hierarchical structures 306–7
 and material world vs. mental world 306–7
 and monadology 307
 and the universal characteristic 315
- levels
 in language structures
 and acronyms 373–4, 395–6, 403–4
 and deception 374–5, 384–5
 and the idea of a perfect language 322–3,
 330–3
 and Newspeak 124–5, 402–4
 and similarity to software 292, 300, 366,
 369, 429–30, 448, 706
 and slogans 372–3, 385–6
 and software totalitarianism 870
 and the term “technology” 386, 389,
 390, 394
 and totalitarianism 863–4
 and Wittgenstein’s later philosophy 341–2
 and word meanings 374–5, 384
see also levels: in structures
- levels (*cont.*)
 in software structures
 and the delusion of high levels 433,
 446–8, 449–52, 466–7, 469, 470–1,
 626, 628–30, 705–6, 800–1
 emulating manufacturing concepts 289,
 449–50
 and file operations 684, 687, 703–4
 and high-level database operations
 678–9, 702–3, 800–1
 and high-level devices 295–6, 434–5, 456
 and high-level programming aids 115–16,
 125–6, 193, 293, 462–3, 626, 648–9, 675–6
 low in high-level systems 449–55, 464–5,
 466–7, 628–30, 648–9, 685, 704, 800–1
 and the need for low levels 445–8, 469,
 628–30, 648–9, 683, 702–4, 705, 724,
 737–8, 800, 803–4
 and ready-made applications 126
 and the relational model 227–9, 676–7,
 678, 682–3, 702–3, 736–8
 and software mechanism 290, 292, 419,
 434–5, 444, 461–2, 831–2
 and software totalitarianism 866–7
see also levels: in structures
- in structures
 definition of 92–3, 96, 554, 636–7
 embodying reductionism and atomism
 85–6
 and the fallacy of abstraction, *see*
 abstraction, fallacy of
 loss of 97, 114–16
 in mathematical systems 728–9, 735–7
 in mind models 151, 153–4
see also levels: in language structures;
 levels: in software structures
- levels, of abstraction, *see* levels
 levels, of complexity, *see* levels
 levels, of programming languages, *see*
 programming languages: levels of
- levels, of skill 157–60, 416–17
- Lévi-Strauss, Claude
 and analysis of myths 244–5
 and binary operations 247, 249
 and classification of social customs 243
 denying creativity and free will 280
- liberal democracy
 definition of 849, 873
 and liberal software concepts 873–5
- life cycles, of software applications, delusion
 of 478–9
- Linger, Richard C.
 and Böhm and Jacopini’s work 576
 and error-free programming 532
 and mathematical programming 501–2, 531

linguistics

Chomskyan, *see* Chomskyan linguistics
and scientism 76, 112, 233, 243, 251, 838–9
theories in, as myths 838

little man, *see* homunculus fallacy

logic, formal, *see* logic systems

logic systems

benefits of 708
Boolean logic as 711
description of 707, 708–10
as hierarchical structures 708, 732
informal aspects of 710
limitations of 708, 710, 713, 732–3
logical operations in 709–10, 712, 730
as mathematical systems 707, 728–9
as models of phenomena 732–3
predicate calculus as 711–12, 732
premises in 710
propositional calculus as 711
syllogistics as 707–8
used to explain non-mechanistic phenomena mechanistically 708, 712–13
see also mathematical systems

logical atomism, Russell's

description of 321, 322–3
and the idea of a perfect language 321,
322–3, 405
as mechanistic theory 322–3

logical atomism, Wittgenstein's, *see*

Wittgenstein's early philosophy

logical positivism

description of 323–4
and the idea of a perfect language 324–6
and the idea of unity of science 325
and meaningfulness 324
as mechanistic project 324
and reductionism 325–6
and verifiability 323–4

logical thinking, difficulty of 206–8

loop, *see* file scanning loops; software
constructs, flow-control: iterative

Lutz, William, and language abuse 373–4,
377–8, 382–4

M

magic

in agriculture 47–8
as belief system 44
blending with skills 47–9, 50–1
in canoe building and sailing 48–9
compared with mana 57
definition of 44
as extension to skills 45–6, 50–1
function of 45

magic (*cont.*)

in modern society 45
as prescientific thinking 44
as pseudoscience 44
similar to science 46
term used in software names 42–3
see also software magic

magic beliefs, due to false assumption 49–50, 52
main structure

delusion of 345–6, 348, 350, 353–4, 363–4,
550–2

the flow-control structure as 363–4, 529,
544–5, 550–2

and flowcharts 345, 364

perceived as the application's logic 346,

351, 353–4, 363–4, 529

and structured programming 363–4, 529,

544–5, 550–2

maintenance, of software applications

as continuous development 478

and the delusion of life cycles 478–9

distorted attitude toward 476–8

and preference of new applications 415,

476–7, 478–9, 520

and programming incompetence 415, 476–7,

519–20

Malinowski, Bronislaw

and magic 45, 46, 47–9, 53, 57

and myths 2–3

mana

compared with magic 57

complexity of the idea 60–1

definition of 57, 60–1

depicting the sacred 61

examples of 57

perceived as tool 62

similar to software power 58–60, 61–2

as substitute for personal knowledge 59–60

term used in various grammatical roles 63–5

as universal potency 62

see also power; software power

managers

as answer to programming incompetence

34–5, 498, 679–80

deskilling of 200, 295–6

similar to primitives 819

as software bureaucrats 34–5

manufacturing

based on mechanism 18

emulated in all activities 18

emulated in object-oriented programming

632, 633, 635, 661

emulated in software theories 27, 32, 34,

52, 72–3, 77, 289, 364, 443, 632, 633,

635, 866–7

- manufacturing (*cont.*)
 emulated in structured programming 531,
 545–6, 561–2
 and hierarchical structures 86–7, 359
 progress in 18
- mapping
 through language 300, 459, 640–1
 as one-to-one correspondence to the actual
 things 300, 303–4, 310, 318, 322–3,
 324–6, 328, 330–3, 335, 344–5, 459, 640–1
 through a perfect language 303–5, 319,
 322–3, 324–7, 328, 344–5
 through software 300, 304–5, 326–7, 329,
 335, 345–6, 459, 642, 652
- Marcuse, Herbert, and language abuse 373
- Martin, James
 and CASE 536
 and denormalization 796
 and the GOTO debate 606, 620
 and the idea of software reuse 640
 and mathematical programming 599 n. 18
 and non-standard constructs 585
 and object-oriented programming 632,
 645, 647, 648
 and the relational model 741
 and software engineering 495
- Marxism
 as myth 4–5
 as pseudoscience 210, 226
- Maslow, Abraham H., and mechanism 275
- mathematical functions, as high-level
 operations 684, 703
- mathematical logic, definition of 707
- mathematical systems
 definition of 728–9
 emulated with language 309–12
 as hierarchical structures 9–10, 87, 309–10,
 728–9, 735–6
 levels of abstraction in 728–9, 735–7
 limitations of 735–6
 logic systems as 707, 728–9
 manipulation in
 as formal aspect 730–1, 735–7, 738
 as main aspect 738
 vs. translation 738
 as models of phenomena 9–10, 89, 105,
 109–10, 156, 600, 728–9
 simple vs. complex 736
 translation in
 as informal aspect 730–1, 738
 vs. manipulation 738
 as minor aspect 738
see also logic systems
- mathematics
 based on tacit knowledge 163–4
- mathematics (*cont.*)
 as basis of science 87–8
 delusion of, in software theories 89–90,
 501–2, 523–4, 531–2, 597–9, 599 n. 18,
 600, 632–3, 680–3, 725, 731–7, 738–9,
 741, 765–7, 791–2
 and determinism 89
 and empirical science 209
 equivalent to mechanism 9–10, 88–9
 ideas in, theoretical vs. practical 569, 598–9
 the world identified with 88–9, 301–2,
 309–12
- meaning
 of sentences 256, 259, 311–12, 319–20
 of words 12, 112, 256, 259, 311–12, 316, 350,
 401, 641
- meaningfulness
 and logical analysis 319–20
 and logical positivism 324
 and Wittgenstein's early philosophy 333–4
- mechanical metaphors
 in human sciences 134
 as method of explanation 71, 78–9, 85–6
 in programming 28, 52–3, 86, 134
 as source of confidence 78–9
- mechanical models
 in biology 136–7
 in economics 135–6
 as method of understanding 71
 in politics 137
 in psychology 135, 139–40
 in sociology 135, 139–40
- mechanical terms, used illegitimately 134, 138–9
- mechanics
 as basis of mechanism 9, 70
 perceived as ultimate science 69, 74–5, 76, 77
 reducing software to 28, 52–3, 72, 633 n. 12
- mechanism
 attempting to reconcile with
 indeterminism 825–6, 827–8, 833–4
 as bad myth 23–4
 based on mechanics 9, 70
 as basis of a perfect society 17, 137–8, 836,
 837, 850
 as basis of behaviourism 236, 238
 as basis of Chomskyan linguistics 261–3
 as basis of information technology 73
 as basis of manufacturing 18
 as basis of science 69
 as basis of software engineering 27–8,
 500–3, 507
 as basis of structuralism 243
 causing charlatanism 20–2, 92, 124, 713, 857
 causing irresponsibility 40–1, 274–5, 818,
 834, 836–8

mechanism (*cont.*)

- causing scientism 20, 127–8, 129–30, 134–8, 141, 161–2, 836–7, 850
- causing the new pseudosciences 13, 191, 204–5, 233–5, 242, 250, 268, 274, 512, 727
- causing totalitarianism, *see* totalitarianism: due to mechanism
- causing utopianism 133, 845, 850, 872
- as convention 71
- corrupting societies 437–8, 818, 859–61
- definition of 8, 9, 68, 70
- denying creativity and free will 278–80, 438, 827–8
- and determinism
 - equivalence of 80, 127, 824, 825–6, 827–8
 - equivalence of, denying 825–6, 827–8
 - and Laplace's claim 72, 569, 824–5
 - in models of phenomena 104–5, 106, 107
 - in software 292–4
- as dogma 7, 26, 91, 109, 233–5, 274–5, 503
- equivalent to hierarchical structures 8–9, 69, 86, 127
- equivalent to mathematics 9–10, 88–9
- failure of mechanists to adhere to 84, 284–6, 287–8
- fragmentary 83–4
- and the homunculus fallacy 151–2, 153
- as ideology, conforming to 853–4, 855, 857–8, 862, 868–9, 873
- in language, *see* language mechanism
- as method, vs. myth 25, 844
- as method of discovery 79–80
- as method of explanation 70–1
- as method of science 9, 69, 274–5, 513
- as myth, *see* mechanistic myth
- naivety of
 - in behaviourism 238
 - in Chomskyan linguistics 262
 - in formal reductionism 83–4, 825
 - as old myth 25–6
 - perceiving all phenomena as simple structures 80
 - reflected in the new pseudosciences 191, 208, 233–4
 - in software theories 73, 516, 547, 680–1, 799
 - in structuralism 246
- as philosophy 9–10, 68–9, 143, 825–6
- promoted by corporations 844, 845, 856–7
- promoted by universities
 - causing corruption 273–5
 - causing totalitarianism 16, 837–8, 854–5
 - degrading the concept of research 233–5, 742
 - denying creativity and free will 277–80

mechanism (*cont.*)

- promoted by universities (*cont.*)
 - through indoctrination 845
 - through mechanistic software concepts 39, 742
 - as myth 7
 - through the new pseudosciences 202, 242, 484–5
 - through propaganda 844
- promoted through indoctrination, *see* mechanistic indoctrination
- promoted through propaganda, *see* mechanistic propaganda
- as psychological support 81–2, 85
- as religion 12, 79, 317
- similar to political ideologies 845, 865–6
- in software, *see* software mechanism
- spread of 437–8
- success of 69
- as trap 130, 419
- see also* mechanistic theories
- mechanistic approximation
 - adequate for mechanistic phenomena 91, 359, 826
 - causing loss of alternatives 829–31
 - classification systems as 11, 103–4, 106, 359, 646
 - and determinism 826, 828–31
 - inadequate for non-mechanistic phenomena 91, 109, 130, 307–8, 359, 826–7
 - of knowledge 165, 418
 - and mind models 154, 156, 172
 - through reification
 - and application definitions 473–4, 477
 - in artificial intelligence 145
 - in behaviourism 238–9
 - characteristic of the new pseudo-sciences 107, 127, 204–5, 233–4, 512
 - and face recognition 110–11
 - and human phenomena 128, 285, 307–8
 - and language 112, 308–9
 - and mind models 154
 - in object-oriented programming 646
 - in the relational model 724–5
 - in structured programming 550–2, 581
 - simple structures as 11, 106, 145, 148, 233–4, 238–9, 474, 550–2, 581, 826, 828–31
 - through software 105, 118–19
 - theories as 106–7, 828–31
- mechanistic capabilities
 - and non-mechanistic problems 23, 25, 66–7
 - and simple structures 164
- mechanistic delusions
 - application definitions as 472–4, 477, 479–82

- mechanistic delusions (*cont.*)
 causing software delusions
 and the fallacy of software engineering
 32-3, 72-3, 495-6, 500-3, 505-6
 and inadequacy of software devices
 191-2, 194-5, 417-18, 419-20, 437-8
 and object-oriented programming 632-3,
 640-2
 and the relational model 680-1, 779-81
 and similarity to language 299-300,
 302, 304-5, 326-7, 640-2
 and structured programming 531-3,
 544-7, 550-2, 561-2, 568-9, 597-9
 and Wittgenstein's early philosophy 335
 consequences of 21, 32-4, 81, 206, 274-5, 368
 exploited by software companies 190,
 419-20, 421, 427-9
 knowledge substitutes as, *see* knowledge
 substitutes: as mechanistic delusions
 mechanism as cause of 69, 81, 91, 107-8, 513
 method to guard against 207
 methodologies as 472-3, 475-6, 479-81
 mind models as, *see* mind models: as
 mechanistic delusions
 object-oriented programming as 642-3,
 645-6, 652-4, 667
 perfect languages as 301-2, 304-5, 319-20,
 323, 326-7, 641-2
 in programming 32-3, 36, 194-5, 290-2,
 453-5, 483-5, 501-4
 programming substitutes as 191-2, 302,
 411-12, 415, 419, 427-8, 459-60, 461
 ready-made products as 856-7
 relational model as 680-1, 713, 727, 748-9,
 779
 and scientific thinking 75
 software devices as 27, 190, 444, 628-30
 software ease of use as 432-4, 449-50
 software power as 432-4, 449-50
 software theories as, *see* software theories:
 as mechanistic delusions
 structured programming as 529-30, 544-7,
 550-2, 561-2, 567-8, 580-1, 590-1,
 599-600, 617, 667
 totalitarianism as 21, 24, 276, 852, 859-60
- mechanistic explanation
 based on mechanical metaphors 71, 78-9,
 85-6
 based on reductionism and atomism 80
 definition of 70
 by separating structures 121
- mechanistic fallacies
 abstraction, *see* abstraction, fallacy of
 causing loss of alternatives, *see* alternatives,
 loss of
- mechanistic fallacies (*cont.*)
 committed together 122-3, 285, 296, 418,
 831, 861
 in language
 and acronyms 373-4, 396, 403-4, 864
 and deception 371-2, 384-5
 and the idea of a perfect language 308-9,
 315-17, 323
 and linguistics 120, 257, 258-9
 and Newspeak 124-5, 400-1, 402-6
 and similarity to software 292, 299-302,
 366, 369-70, 408-9, 429-31, 459-60,
 838-9, 869-70
 and slogans 372-3, 863-4
 and Swift's language machine 458-9
 and the term "technology" 394
 and totalitarianism 863, 865
 and mechanistic models 123-4, 126-7
 and mind models 154-5
 and non-mechanistic models 123-4, 126-7
 reification, *see* reification, fallacy of
 in software
 and application definitions 474
 causing loss of alternatives 125-6, 365,
 367, 418
 and the delusion of high levels 290-2,
 444, 446-7, 469, 626, 628-30, 650,
 675-7, 682-3, 800-1
 due to emulating manufacturing
 concepts 289, 410, 411, 561-2
 due to programming incompetence 121,
 126
 and high-level devices 440, 832-3
 and high-level programming aids 115-16,
 192-3, 419, 682-3
 and knowledge substitutes 190, 417-18
 and mechanistic theories 514-15, 626,
 675-6, 713
 and similarity to language 299-302,
 366-7, 408-9, 421, 642
 and software totalitarianism 866-7
- mechanistic indoctrination
 through advertising 845, 857
 through education 16, 845, 855
- mechanistic knowledge
 as novice level 157-8, 164-5, 169, 423-5,
 488, 504, 834
 restriction to, *see* non-mechanistic
 knowledge: prevention of
- mechanistic language
 causing loss of alternatives, *see* alternatives,
 loss of: due to mechanistic language
 as high levels of abstraction 366, 429-30,
 863-4, 865, 870
 prevalence of 406-7, 864

- mechanistic language (*cont.*)
 preventing non-mechanistic knowledge
 369–70, 406–7, 429–30, 865, 870
 and progression toward totalitarianism
 399, 405–7, 864, 869–70
see also language mechanism
- mechanistic models
 adequate for mechanistic phenomena 10, 69,
 80, 89, 90–1, 104–5, 108, 359, 500–1,
 580–1
 as approximation, *see* mechanistic
 approximation
 in artificial intelligence 145–6, 262
 hierarchical structures as 85–6, 90–1, 93,
 108, 127, 277–8, 283, 828–31
 and large number of alternatives 283–6,
 828–31
 and the mechanistic fallacies 123–4, 126–7
 and non-mechanistic phenomena 91, 104–5,
 107–9, 204–5, 569
 and the three-body system 109–10
- mechanistic myth
 alternatives to 25
 as belief system 7, 14, 81–2
 causing programming incompetence 32–3
 dependence on 25, 26
 due to reductionism and atomism 81
 and non-mechanistic phenomena 10
 origins of 70
 software as 26, 28, 31, 33
see also software myth
- mechanistic performance
 as novice level 157–8, 164–5, 169, 416
 restriction to 423–5, 833, 834–5, 836, 855,
 857–8
- mechanistic phenomena 10, 69, 80, 89, 90,
 104–5, 108, 359, 500–1, 580–1
- mechanistic problems 66, 195–7, 419–20
- mechanistic propaganda
 conducted by corporations 844, 856–7
 conducted by universities 844, 854
- mechanistic software
 restriction to 29, 66–7, 275, 291, 411–13, 438,
 866–8, 870–1, 872, 874–5
see also software mechanism
- mechanistic solutions
 products as 856
 and the promise to solve non-mechanistic
 problems 21, 65–7, 422–5, 854, 856–7,
 865, 868, 872
 software devices as, *see* software devices
- mechanistic theories
 and analogies 77, 246
 as approximation, *see* mechanistic
 approximation
- mechanistic theories (*cont.*)
 in artificial intelligence 145–6
 based on the idea of innate faculties 284–5,
 828
 circularity of 134, 152, 154, 207, 255, 270,
 320, 851–2
 as delusions, *see* mechanistic delusions
 failure of
 due to misinterpreting creativity 283–6,
 288–90
 due to perceiving minds as machines
 277–8, 854
 masked through informal analogies 77
 in the new pseudosciences 191, 234–5,
 569, 727–8
 as frauds 81, 513
 for human phenomena 128, 129–30, 134–40,
 277–8, 828–31
 and language 76, 112, 244, 251, 260–1, 268,
 269–70, 308–9, 322–3, 330–3, 838–9
 and mechanistic phenomena vs. non-
 mechanistic phenomena 10, 69, 207,
 233–4
 of mind, *see* mind models
 and non-mechanistic capabilities 143, 194–5,
 326–7, 417–18, 443, 596–7, 662, 834–5
 for non-mechanistic phenomena
 based on reification, *see* mechanistic
 approximation: through reification
 in behaviourism 236, 238–9, 278–9
 in Chomskyan linguistics 256–8, 263–5,
 280–2
 as delusions 89, 108–9, 133, 207, 513,
 828–31
 in economics 861
 inadequacy of 10–12, 104–5, 107–8, 161–2,
 233–5
 in the new pseudosciences 191, 233–5,
 277–8, 283–7, 436–8, 727–8
 in software 194–5, 501–3, 713, 725
 in structuralism 244–7, 248–50, 280
 as poor approximations
 of human phenomena 128, 130, 139–40,
 307–8, 828
 of non-mechanistic phenomena 825,
 826–7, 829–31
 of software phenomena 474, 550–2, 581,
 681
 popularity of 81
 similarity of 85
 in software, *see* software theories
 ways to identify 127
see also mechanism
- mechanistic understanding, *see* mechanistic
 explanation

- media
 employing confirmations rather than falsifications 221–2, 627, 674, 822
 and software propaganda 222, 536–7, 627, 674, 822
- medical technology, analysis of term 397–8
- messages, in object-oriented programming, *see* object-oriented programming: messages in
- metaphors
 and creativity 176
 hidden 86, 87
 hierarchical 86, 88
 immediate 86
 mechanical, *see* mechanical metaphors
- method, geometrical, and hierarchical structures 88, 89, 208
- method of science, mechanism as 9, 69, 274–5, 513
- methodologies
 and application definitions 472–4
 and CASE 481–3
 as frauds 475–6
 as low levels in high-level systems 476
 as mechanistic delusions 472–3, 475–6, 479–81
 requiring non-mechanistic knowledge 480–1
 as substitutes for programming expertise 146–7, 197–9, 475–6
see also programming substitutes
- methods, for logical thinking 207–8
- methods of inquiry, vs. magic practices 44
- Microsoft Corporation
 fostering ignorance and dependence 489
 fostering programming incompetence 486–7
- Mills, Harlan D.
 and Böhm and Jacopini's work 576, 605
 and error-free programming 532, 534–5
 and the GOTO debate 605
 and mathematical programming 501–2, 531
 and structured programming 529–30, 603 n. 1
- mind
 and artificial intelligence 145–6, 183–6
 and body, perceived as separate entities 149, 306–8, 310
 and complex structures 164, 194, 247, 270–3, 416–18
 controlling, *see* mind: degrading
 degrading
 through language mechanism 124–5, 366, 399–401, 404–5, 406–7, 408–9, 429–30, 863, 865, 869–70
- mind (*cont.*)
 degrading (*cont.*)
 through software devices 190, 411–13, 423–5, 433–5, 832–3, 834–5
 through software domination 413, 427–8, 868
 through software mechanism 367, 411–13, 429–31, 437–8, 870–1
 disorders, and creativity 178–81, 185
 and the Ghost in the Machine 149
 mechanistic capabilities of, *see* mechanistic capabilities
 mechanistic theories of, *see* mind models
 as natural phenomenon 156–7
 non-mechanistic capabilities of, *see* non-mechanistic capabilities
 and non-mechanistic problems 23, 25, 66–7, 142, 419–20, 857–8
 and partial reductionism 84
 potential of, as myth 25
 represented with hierarchical structures 143–4
 and simple structures 164, 193–4, 416–18
- mind control, *see* mind: degrading
- mind mechanism, *see* mind models
- mind models
 based on analogy to machines 181–2
 based on hierarchical structures
 in artificial intelligence 145–6, 262
 due to the mechanistic philosophy 143–4
 and knowledge substitutes 295–7, 832
 as mechanistic delusions 149–50, 153–4
 and programming substitutes 293–4
 in structuralism 245–7
 based on mechanistic theories
 in artificial intelligence 145–6
 in behaviourism 236–7
 in Chomskyan linguistics 261–2, 268, 270, 281–2
 circularity of 153–4
 failure of 145–6, 172, 834
 fallacies of 149–50, 154–5
 as reduction projects 84
 in structuralism 245–7
 as basis of software devices 190, 293–4, 832
 as basis of software theories 146–7, 192, 293–4, 834
 limitations of 183–7, 188–9
 as mechanistic approximation 154, 156
 as mechanistic delusions
 in artificial intelligence 153–4, 262
 in Chomskyan linguistics 268, 271–3
 and contexts 188–9
 and creativity 172, 181–3
 and the mechanistic fallacies 154–5

- mind models (*cont.*)
 non-mechanistic 170, 181, 270–3
 and political ideologies 280–2
 programming substitutes as 192, 293–4
 and rationalism vs. empiricism 280–2
 as substitutes for knowledge 143, 146–7,
 188–9, 417–18
- Minsky, Marvin L., and artificial intelligence
 145
- mirroring, *see* mapping
- MMX technology, analysis of term 388–90
- models
 complex structures as, *see* non-mechanistic
 models
 deterministic, *see* mechanistic models
 indeterministic, *see* non-mechanistic models
 mechanistic, *see* mechanistic models
 of mind, *see* mind models
 non-mechanistic, *see* non-mechanistic
 models
 simple structures as, *see* mechanistic models
- modules, *see* software modules
- modus tollens, and Popper's criterion of
 demarcation 214
- mythology, *see* myths
- myths
 ancient 2–4
 bad 15–16, 23–4
 based on false assumption 14
 as basis of totalitarianism 30
 as collective thinking 3–4
 and elites 14–15, 23–4
 function of 2–4
 good 14–15
 harmful effects of 7
 as human need 7
 mechanism as, *see* mechanistic myth
 modern 3–7
 potential of minds as 25
 religious 15–16, 24
 scientific 24
 software as, *see* software myth
 and structuralism 244–5
 transition between 15, 24–6

N

- Nagel, Ernest, and mechanism 69, 72
- naivety
 of artificial intelligence 145, 261–2, 435
 of determinism 72
 of the early scientists 79–80, 83–4
 of mechanism, *see* mechanism: naivety of
 of modern scientists 80, 83–4, 145, 191,
 233–4, 238, 246, 261–2
- naivety (*cont.*)
 of the software theorists 516, 547, 577, 592,
 617–18, 664, 681–2, 727, 748–9, 799
- natural order, idea of
 as scientism 850
 used to justify totalitarianism 851
- Nazism
 degrading the concept of responsibility 836
 as myth 6
 as totalitarian democracy 849
 and utopianism 132
- nesting
 of file scanning loops 697–700, 701, 788, 793
 of files 697–700, 702
 of files, in the relational model, *see*
 relational model: file nesting in
 of flow-control constructs 525–8, 582, 697
 of messages, in object-oriented
 programming 664, 666, 667
- nesting scheme
 in object-oriented programming, *see*
 object-oriented programming:
 nesting scheme in
 in structured programming, *see* structured
 programming: nesting scheme in
- new pseudosciences
 definition of 204, 512, 727
 due to mechanism 13, 191, 204–5, 233–5,
 242, 250, 268, 274, 512, 727
- Newspeak
 and abstraction 402–5
 and acronyms 403–4
 eliminating words and meanings 400–1
 and the GOTO debate 615
 and the idea of a perfect language 405–6
 inventing new words 402–3
 and mind control 124–5, 399–400, 430–1
 and reification 400–1
 and similarity of language and software
 411–12, 430–1
 simplifying the grammar 404–5
- Newton's theory of gravitation
 contradicting mechanism 77–8
 as deterministic theory 71–2
 as inspiration in human sciences 134
 interpreted as mechanistic 79
 as mathematical model 728–9
 as mechanistic approximation 106
- non-mechanistic approximation, complex
 structures as 148, 170–1
- non-mechanistic capabilities
 and complex structures 110–11, 112, 147–8,
 164, 194–5, 417–18, 501, 504, 593–4
 and creativity 172, 174–6, 177–8, 277–8, 434–5
 and evolution 142, 154, 271

- non-mechanistic capabilities (*cont.*)
 and free will 277–8
 and intuitive knowledge 164, 271–3, 495
 and mechanistic theories 143, 194–5, 326–7,
 417–18, 443, 596–7, 662, 834–5
 and multiple attributes 104, 316, 337–41,
 356–9
 neglecting 25, 142, 190, 423–5, 437–8, 439,
 456, 832–3, 858, 865–6
 and non-mechanistic mind models 170–1,
 181, 270–3
 and non-mechanistic problems 23, 25, 66–7,
 142, 419–20, 857–8
 and non-mechanistic theories 128, 130
 and recognizing contexts 173, 184–5
 vs. software devices 66–7, 188–9, 190, 196–7,
 290–2, 417–18, 423–5, 437–8, 439, 440,
 456, 833
- non-mechanistic knowledge
 appreciation of 439
 craftsmanship as 497, 504, 505
 development of 148, 157–60, 161–3, 164–5,
 166–7, 168–9, 171, 270–3, 416–17, 423–5,
 439
 and intelligence 142, 145, 181–3, 194
 as interacting structures 148, 161–2, 165–7,
 316, 323, 417–18, 652, 832–3
 and language 122, 166–7, 256–7, 258–9,
 270–3, 406, 411–12, 865
 prevention of
 due to denying creativity and free will
 832–3, 855
 due to mechanism 23, 25
 due to mechanistic language 369–70,
 406–7, 429–30, 865, 870
 due to software devices 65–7, 290–2,
 411–13, 419–20, 422–5, 434–5, 436,
 437–8, 439, 456–7
 due to software mechanism 66–7, 370,
 437–8, 488, 489, 504, 532–3, 594, 599–
 600
 due to totalitarianism 857–8, 865–6,
 868, 870–1, 874–5
- in programming
 and the delusion of high levels 455, 470,
 675–6, 831–2
 and the fallacy of software engineering
 495, 501, 504, 507, 515
 and interacting structures 441–3, 567–8,
 593–7, 599–600, 662, 681–2
 and methodologies 480–1
 prevention of 194–5, 294–5, 411–13, 432,
 505–6
 and reification 121, 154–5, 301, 315–16, 371,
 400–1, 418
- non-mechanistic knowledge (*cont.*)
 in software-related affairs 66–7, 193, 427–8,
 480–2, 488, 489
 and stories 357–9, 442–3
see also tacit knowledge
- non-mechanistic models
 interacting structures as 90
 and the mechanistic fallacies 123–4, 126–7
 of mind 170, 181, 270–3
 and non-mechanistic phenomena 91
- non-mechanistic phenomena
 and charlatanism 65–7, 857
 and determinism 91, 234, 283–6, 826–7
 examples of 109, 110, 112, 147–8, 829–30
 language as, *see* language: as non-
 mechanistic phenomenon
 and mechanistic models 91, 104–5, 107–9,
 204–5, 569
 and the mechanistic myth 10
 and mechanistic naivety 80, 191, 233–4,
 238, 246
 mechanistic theories for, *see* mechanistic
 theories: for non-mechanistic
 phenomena
 non-mechanistic theories for 206
 in programming 194–5, 198, 501
 and reification 116–18, 233–4, 238, 244, 257,
 861
 software as, *see* software: as non-
 mechanistic phenomenon
 and software theories 125–6, 194–5, 206,
 867
see also complex structures
- non-mechanistic problems
 and inadequacy of software devices 65–6,
 195–7, 417–18, 419–20, 422–3, 469
 and mechanistic capabilities 23, 25, 66–7
 and the mind 23, 25, 66–7, 142, 419–20,
 857–8
 and non-mechanistic capabilities 23, 25,
 66–7, 142, 419–20, 857–8
 and the promise of mechanistic solutions
 21, 65–7, 422–5, 854, 856–7, 865, 868,
 872
- non-mechanistic theories
 and human phenomena 128, 129–30
 required for non-mechanistic phenomena
 206
- normal forms, *see* normalization, in the
 relational model
- normalization, in the relational model
 1NF (first normal form) in
 annulment of 786–7, 789–90
 definition of 718, 720, 786
 examples of 718–19, 786–7

- normalization, in the relational model (*cont.*)
- 1NF (first normal form) in (*cont.*)
 - as formal principle 721, 743, 747–8, 784, 791
 - importance of 789–90
 - inefficiency of 785, 787, 787 n. 2, 792–3
 - 2NF, etc. (higher normal forms) in
 - annulment of 794–5
 - definition of 718, 720, 745
 - examples of 772–9
 - inefficiency of 746, 769, 785, 794
 - as informal principle 721, 748, 784
 - and optimal normal forms 769
 - and updating operations 721, 743, 747–8, 769, 770, 801
 - based on fallacious arguments 780–1
 - definition of 718, 743, 784
 - denormalization in
 - definition of 795
 - as falsification of the relational model 785, 794–6, 797–9
 - as informal procedure 796–8
 - misrepresented by the theorists 795–7
 - perceived as exact method 796–8
 - procedure of 796–7
 - as reversal of normalization 743, 785, 794–6
 - vs. traditional design principles 795, 797–8
 - and field dependencies 719, 761–3, 765–7, 768–71, 772, 773–4, 775–7, 778–9, 802
 - as fraud 726, 743, 762, 771, 783
 - as game 767
 - as independent theory 746, 779
 - inefficiency of 746, 785
 - as informal procedure 721–2, 726, 738, 748, 761, 762–3, 764–5, 768–9, 772–9, 797–8, 801–2
 - manipulation in, vs. translation 766
 - mathematical claims of 743, 748, 764, 766–7
 - as mechanistic delusion 742–3, 748–9, 764–5, 771, 779, 783, 801–2
 - non-1NF (non-first-normal-form) in
 - as annulment of 1NF 786–7, 789–90
 - definition of 787–8
 - as falsification of the relational model 788, 789–90, 791, 794
 - as informal principle 791
 - promotion of 788–9, 792–4
 - as replacement of JOIN operations 788–9
 - vs. normalization in traditional databases 720, 745–6, 757, 758–9
 - and ordinary field dependencies 770–1, 802
 - overnormalization as formal procedure of 769
- normalization, in the relational model (*cont.*)
- perceived as formal design principles 726, 742–3, 748–51, 761, 762–3, 764–5, 772–9, 780–1, 782–3, 797–8, 801–2
 - perceived as substitute for expertise 760, 761–2, 764
 - procedure of 720, 745, 761–2, 769, 785
 - and similarity to the traditional file relationships 758–9, 762–3
 - as specious mathematics 765–7
 - as system of representation 766–7
 - vs. traditional design principles 762–3, 764–5, 772–9, 781–3, 798
 - translation in, vs. manipulation 766
 - as unnecessary theory 726, 749–52, 757, 759, 762–3, 764–5, 768, 772
 - novice performance, characteristics of 157–60, 164–5, 416
- O
- object-oriented programming
- and abstraction 636–7, 651, 657, 669–70
 - attributes in
 - and complex structures 642–3
 - definition of 636
 - and the modified concept of inheritance 658–9, 660
 - classes in
 - as basis of software reuse 638–9
 - definition of 637
 - as hierarchical structures 637, 645–6
 - as templates 638
 - used as subroutines 656, 657
 - and classification systems 636–7, 638–9, 642–3, 645–6, 652–4, 658–9, 660
 - delusions of, summarized 650–1
 - difficulties of 647–8, 662
 - downgraded to informal theory 656, 660, 662
 - emulating manufacturing concepts 632, 633, 635, 661
 - encapsulation in 670
 - expanded in order to suppress falsifications
 - by allowing any attributes 661
 - by allowing informal principles 669–72, 673
 - by allowing interacting hierarchies 649–50, 651, 655–6
 - by allowing multiple flow-control structures 667
 - failure of 649, 661, 667, 671–2, 675–6
 - fallacies of
 - allowing interacting hierarchies 655–6, 658, 661, 662

object-oriented programming (*cont.*)
 fallacies of (*cont.*)
 attempting to represent applications
 mathematically 632–3
 claiming both simplicity and difficulty
 646–8
 ignoring the flow of execution 662–4
 ignoring the significance of hybrid
 languages 656
 ignoring the structures generated by
 messages 666, 667, 668
 misinterpreting the problem of multiple
 hierarchies 655–6
 misinterpreting the problem of one
 hierarchy 652–3
 perceiving inheritance as modifiable
 principle 657–8, 659–60
 perceiving software mechanism as
 substitute for expertise 675–6
 promising automatic application
 development 633
 reinstating the traditional programming
 concepts 648–9, 655–6, 660, 661–2,
 668, 670
 replacing the original promises with
 trivial concepts 669–72, 673
 restricting relations to one hierarchy
 642–3, 645–6, 648–9, 651–4
 reversing the principles of
 programming 629–30, 648–9
 using classes as subroutines 656, 657
 flow of execution in
 between objects 665–6, 667
 as complex structure 666, 667
 ignored by the theory 662–4
 within objects 666
 same as in structured programming
 665–6, 666 n. 3
 and state transition diagrams 665 n. 1
 as fraud 650, 651, 668
 and hierarchical structures 636–7, 642–3,
 645–6, 652–4, 658–9, 666
 information hiding in 670
 inheritance in
 allowed to change and omit attributes
 658–9
 allowed to take attributes from other
 classes 659–60
 benefits of 637–8
 definition of 637, 657
 modifying the concept of, as falsifica-
 tion of the theory 657–8, 659–60
 multiple 660
 obviated by using classes as subroutines
 656, 660

object-oriented programming (*cont.*)
 inheritance in (*cont.*)
 as reverse of abstraction 637, 657
 and interacting software structures 642–3,
 648–50, 652, 653–4, 659, 662
 as low levels in high-level systems 648–9,
 650, 675–6
 as mass madness 631–2
 mathematical claims of 632–3
 as means to control programming
 practices 675–6
 as mechanistic delusion 642–3, 645–6,
 652–4, 667
 messages in
 definition of 636
 determining the flow of execution 663–4,
 665–6, 667
 generating structures 668
 methods in 663–4, 666
 and misuse of the terms “object” and
 “object-oriented” 633–4, 669, 672–3
 nesting scheme in
 generated by messages 664, 666, 667
 same as in structured programming 666
 objects in
 definition of 631, 636
 degrading the concept of 673
 as hierarchical structures 636–7, 663–4
 as instances of classes 638, 663–4
 similar to subroutines 636, 666, 667–8
 as software principle 667–8
 as software processes 667–8
 origins of 631
 overloading in 671
 perceived as a new paradigm 631, 635
 perceived as a revolution 630–1, 640, 646
 polymorphism in 670–1
 preventing interactions between structures
 653–4
 and programming incompetence 635–6,
 675–6
 promises of 630–2, 634–5, 639–40, 644–5,
 661, 663
 promotion of 630–2, 634–5, 646–7, 673–4
 as pseudoscience 649, 651, 655, 661, 667, 672
 and reification 642, 645–6, 652
 requiring non-mechanistic knowledge
 662, 676
 requiring traditional languages 655–6, 657
 and software reuse
 through one class hierarchy 638
 principles of 631, 638–9
 through several class hierarchies 639
 and software wrapping 673
 and structures, static vs. dynamic 664

objects, in object-oriented programming, *see*
 object-oriented programming:
 objects in

one-to-one correspondence, *see* mapping

operationalism, and scientism 139

operations, between elements of structure
 92–3, 548, 554, 637

operations, logical
 in logic systems 709–10, 712, 730
 in predicate calculus 712, 715–17, 732

operations, software
 on files, *see* database operations; file
 operations
 low-level vs. high-level, *see* levels: in
 software structures
 in programming languages, definition of
 517 n. 1
 in the relational model, *see* relational
 operations
 repeated, as software processes 354–5
 in Turing machines 569–70

Orwell, George
 and central economic planning 860, 862
 and corruption of the intellectuals 859
 and mechanistic language 400–1, 402–7,
 863, 864, 870
 and mind control through language 124–5,
 398–401, 402–5, 430–1, 862–3
 and origins of *Nineteen Eighty-Four* 398–9,
 858–9, 862–3
 and progression toward totalitarianism
 399, 405–7, 858–9, 860, 864, 865, 871
 totalitarian model of, *see* totalitarianism:
 Orwell's model of

P

packages, software, *see* software applications:
 ready-made

particles, used to explain phenomena 9, 74–5

parts, elementary 78

Pascal, Blaise, and intuitive knowledge 144

perfect language
 idea of 302, 319–21, 344–5, 405–6, 640–1
 idea of, in software 302, 326–7, 345–6, 642
 and logical positivism 324–6
 as mapping system 303–5, 319, 322–3, 324–7,
 328, 344–5
 as mathematical language 320, 322–3
 as means to represent the world 301–2,
 303–4, 319–21, 325–6, 330–3, 640–1
 as mechanistic delusion 301–2, 304–5,
 319–20, 323, 326–7, 641–2
 and Russell's theory of logical atomism
 321, 322–3, 405

perfect language (*cont.*)
 and the story of Babel 303–4
see also artificial languages

perfect society
 as aim of totalitarianism 16–17, 18–19, 133–4,
 836
 based on knowledge substitutes 17
 based on mechanism 17, 137–8, 836, 837, 850
 based on science and conformism 849, 866
 through behaviourism 279
 idea of 17–18, 131–2, 133–4
 and liberal democracy vs. totalitarian
 democracy 849

personal experience
 based on whole knowledge structures
 434–5
 as basis of non-mechanistic knowledge
 148, 149, 164–5, 416–17
 impossible for mind models 156, 182–3, 188
 and language acquisition 272–3
 prevented by the software elites 423–5,
 427–8
 in programming
 prevented by software devices 456–7
 required for expertise 193, 198
 and skill acquisition 157–60, 161–2

Pettit, Philip, and structuralism 248

phenomena
 deterministic, *see* mechanistic phenomena
 explained through particles 9, 74–5
 human, *see* human phenomena
 indeterministic, *see* non-mechanistic
 phenomena
 interacting, *see* non-mechanistic phenomena
 mechanistic 10, 69, 80, 89, 90, 104–5, 108,
 359, 500–1, 580–1
 non-mechanistic, *see* non-mechanistic
 phenomena
 perceived as hierarchical structures 9–10, 80
 represented mathematically 9–10, 89, 105,
 109–10, 156, 600, 728–9
 software, *see* software phenomena
 supernatural, software as, *see* software
 magic; software power

philosophy, mechanism as 9–10, 68–9, 143,
 825–6

Piaget, Jean
 and binary operations 245–7
 denying creativity and free will 280

Plato, and totalitarianism 131

Poincaré, Henri, and creativity 174–5

pointer, *see* file operations: pointer in
 Polanyi, Michael
 and the learning process 167
 and tacit knowledge 160–4

- politics
 and scientism 137–8, 243, 836, 850
 turned into morbid preoccupation 874
- Popper, Karl R.
 and the criterion of demarcation 213–14,
 216–18, 224–6, 232
 and determinism 91–2, 826
 and empiricism 216
 and Freudianism 224
 and the growth of knowledge 208–9, 215
 and the idea of falsifiability 216–18
 and the idea of verifiability 324, 334
 and Marxism 226
 and the Popper legend 232
 and the problem of demarcation 208, 209,
 210–11
 and the problem of induction 211, 213–14
 and reductionism 81
 and testing criteria 219
 and totalitarianism 131
- power
 as abstract term 58
 definition of term 58
 term used in software names 59
 term used in the sense of *mana* 58–60, 62–5
 term used in various grammatical roles
 63–5
see also mana; software power
- power tools
 term used in book titles 60
 term used in software names 60
- practices
 business, definition of 348
 software, definition of 348
 as software principle 348–50, 542–3
 as software processes 348–50, 361, 440–1
 as structures 348–9, 361–2, 440–1
- predicate calculus
 as basis of the relational model 713–14,
 715–16, 721, 725, 729–30, 732, 746–7,
 765, 791
 description of 711–12
 examples of 711–12
 limitations of 733–4
 as logic system 711–12, 732
 operations in 712, 715–17, 732
 relations in 711, 732
 tuples in 711, 732, 746–7
- primitive beliefs, due to ignorance 42, 43–4,
 59–60, 819
- primitive society
 magic in 45–6, 47–9, 50–1
mana in 57, 60–1
 myth in 2–3
 similar to modern society 43–4
- principles, software, *see* software principles
- problems
 mechanistic 66, 195–7, 419–20
 non-mechanistic, *see* non-mechanistic
 problems
 software-related, *see* software-related
 problems
- processes, software, *see* software processes
- products
 dependence on 856, 858
 as mechanistic solutions 856
 software perceived as 42 n. 1, 429, 866–8
- programmers
 application vs. system 36, 199
 avoiding programming 198, 419, 456–7,
 486, 819–20
 compared to craftsmen 494, 497–9, 505,
 635–6, 675
 dependent on programming substitutes
 32–3, 54–5, 198–9, 294–5, 411–13, 415,
 456–7, 465–8, 486–7, 626, 675–6, 792,
 819–20
 deskilling of 34–5, 35 n. 2, 486–7, 493, 497–
 500, 498 n. 13, 618–21, 625–6, 675–6
 education of 32, 197, 414–15, 497, 505–6,
 619, 621
 incompetence of, *see* programming
 incompetence
 perceived as magicians 56
 similar to primitives 54–5, 65–7, 819
 as software bureaucrats 32, 35–6, 414–15,
 468, 486, 493, 498–500, 625
 spurious activities of 54–5, 199, 483–5
 as unskilled workers 34–5, 198–9, 497–8,
 618–19, 620–1, 675, 678–9
- programming
 avoided, as principle 198, 419, 456–7, 486,
 819–20
 expertise in, *see* programming expertise
 incompetence in, *see* programming
 incompetence
 inefficiency in 32–3, 126, 199, 483–4, 486
 irresponsibility in 486–7, 498, 620–1, 624–5,
 822–3
 mechanistic delusions in 32–3, 36, 194–5,
 290–2, 453–5, 483–5, 501–4
 non-mechanistic knowledge in, *see* non-
 mechanistic knowledge: in
 programming
 non-mechanistic phenomena in 194–5,
 198, 501
 reversing the principles of 462, 465, 467–8,
 629–30, 648–9, 685
 science of, software theories perceived as
 516, 523, 617–18

- programming (*cont.*)
see also software engineering; software theories
- programming aids, *see* programming substitutes
- programming expertise
 as capacity for complex structures 121, 193, 294-5, 365, 415, 419, 421, 441-3, 505-6, 519, 567-8, 596-7
 as capacity for low levels 367, 419, 421, 702
 as craftsmanship 505-6, 515
 definition of 414
 degraded, *vs.* traditional expertise 414-15, 439, 484, 504, 505-6, 621, 626
 degrading the concept of, *see* programming substitutes: replacing expertise
 prevention of 34-5, 54-5, 197-9, 294-5, 411-13, 414-15, 487, 493, 497-500, 620-1
 replacing with substitutes, *see* programming substitutes: replacing expertise
 standards for 34, 197, 621, 822, 822 n. 3
- programming incompetence
 accepted by society 34, 621
 causing dependence on the software elites 456-7, 475-6, 626, 675-6, 792, 816-17
 causing poor application maintenance 415, 476-7, 519-20
 causing software failures 32-3, 483-5, 821-2
 causing the software bureaucracy 34-5, 37-8, 198-9, 411-12, 679-80
 and the concept of expertise 432, 498, 505-6
 cost of 32-3, 200, 416, 486, 490, 520, 820 n. 2
 and database systems 678-9, 705, 759-60, 792, 817
 due to software mechanism 32-3, 126, 197, 291, 294-5, 456-7, 483-5, 504, 625-6, 675-6
 fostered by software companies 35-6, 456-7, 466, 468-9, 819-20, 820 n. 2
 fostered by the software elites 413, 416, 431, 486-7, 618-19, 620-2, 624-6, 675-6
 fostered by universities 39, 65, 275, 676, 794
 GOTO prohibition perceived as answer to 618-20
 and object-oriented programming 635-6, 675-6
 and the relational model 685, 759-60, 764, 792, 817
 and software magic 43-4, 54-5, 819, 821
 software mechanism perceived as answer to 515, 520-1, 618-19, 625-6, 675-6, 760, 764
- programming incompetence (*cont.*)
 and structured programming 516, 618-22, 625-6, 674-6
 turned into management challenge 34-5, 498, 678-80
- programming languages
 and formal grammars 261-3
 fourth-generation, *see* fourth-generation languages (4GL)
 generations of 464-5, 503
 hybrid, in object-oriented programming 656
 levels of 464-5, 503
 non-procedural, as low levels in high-level systems 452-3, 809-10
 in relational database systems
 as means to override the relational principles 806-8
see also SQL (database language)
 traditional, as correct level of abstraction 462-3, 464-5, 628-30, 655-6, 703-4, 705
see also C (programming language); COBOL (programming language)
- programming profession
vs. craftsmanship 497-9, 505
 destruction of 27, 32-4, 37, 197, 200, 294-5, 486-7, 493, 676
- programming substitutes
 as consumer products 38-9
 database systems as 678-80, 759-60, 817
 and the delusion of high levels
 dishonest nature of 444, 448, 449-52, 456-7, 461, 465-8
 in object-oriented programming 628-30
 promoted by the elites 290-2, 293, 867
 in ready-made applications 470-1
 in the relational model 676-7, 702-3, 800-1
 dependence on, *see* dependence: on programming substitutes
 failure of 27, 146-7, 160, 198, 294-5, 821-2
 as mechanistic delusions 191-2, 302, 411-12, 415, 419, 427-8, 459-60, 461
 as mind models 192, 293-4
 perceived as magic systems 54-5, 819, 821
 promoted by the elites 38-9, 147, 192, 449-52, 486-7, 531-3, 618-19, 650, 759-60
 replacing expertise
 as mechanistic delusions 32-4, 146-7, 191-2, 194-5, 294-5, 431, 599-600
 and methodologies 475-6
 and object-oriented programming 628-30, 640, 675-6

programming substitutes (*cont.*)
 replacing expertise (*cont.*)
 and primitive beliefs 54–5, 56, 65–6
 and the relational model 677–8, 759–60, 761–2
 and the software bureaucracy 197–9, 411–12, 415, 416
 and structured programming 516, 520–1, 531–3, 567–8, 625–6
 programming theories, *see* software theories
 programs, structured 528, 537, 577, 587
 promotion, *see* advertising
 propaganda
 mechanistic, *see* mechanistic propaganda
 software, *see* software propaganda
 propositional calculus, as logic system 711
 pseudosciences
 astrology as 13
 based on false assumption 205, 512
 based on induction 211–12
 behaviourism as 235, 238–9, 240–1, 279
 causing totalitarianism 279, 846, 859–60
 characteristics of 191, 203–5, 233–5, 511
 Chomskyan linguistics as 251–2, 259–60, 263–6, 273, 280
 definition of 203, 511
 examples of 203, 224, 226, 227–30, 235, 242, 251, 509–10
 exposed through the principles of demarcation 208, 231, 512
 Freudianism as 210, 224
 ignoring falsifications 221, 239, 250, 511, 514, 627
 as logical activities within a delusion 205, 509–10, 512–13, 514
 magic as 44
 making vague statements 217, 224, 241, 247–8
 Marxism as 210, 226
 the new, *see* new pseudosciences
 object-oriented programming as 649, 651, 655, 661, 667, 672
 perceiving human beings as deterministic systems 181–3, 277–8, 437–8, 827–8, 833–4, 854–5
 perceiving societies as deterministic systems 833–4, 854–5
 popularity of 230–1, 279
 relational model as 227–30, 676–7, 706, 725, 727, 784, 790–1, 795, 801
 resembling science 204, 205, 512–13
 software engineering as 492, 509, 627
 software theories as 29, 206, 209–10, 436–8, 451, 510–11, 514–15, 549, 626–7, 651, 674
 as spreading belief systems 52, 511–12

pseudosciences (*cont.*)
 structuralism as 242, 247, 280
 structured programming as 549, 552, 579, 590, 601, 626–7, 667, 672
 totalitarianism as 846, 859
 unfalsifiable nature of
 and behaviourism 240–2
 and Chomskyan linguistics 259–60, 265–6, 267, 269
 as concept, explanation of 210, 217–18, 224–6, 512, 724–5
 and the new pseudosciences 234–5, 344, 513, 727–8, 827
 and object-oriented programming, *see* object-oriented programming:
 expanded in order to suppress falsifications
 and the relational model, *see* relational model:
 expanded in order to suppress falsifications
 and software devices 451, 626, 801
 and software theories 514, 626
 and structuralism 248–50
 and structured programming, *see* structured programming:
 expanded in order to suppress falsifications
 using stratagems to avoid falsifications 224–6, 238–9, 248–50, 263–6, 344, 672, 854
 psychology, and scientism 135, 139–40, 233, 235–6, 243

Q

query systems, as low levels in high-level systems 454–5

R

Ralston, Anthony
 and Böhm and Jacopini's work 576, 605
 and the first normal form 790
 and the GOTO debate 604, 605
 and non-standard constructs 581–2
 and the relational model 741
 and software engineering 508
 and structured programming 578–9
 rationalism
 and hierarchical structures 306
 and the language fallacies 308–12, 313–18, 651–2
 and material world vs. mental world 306–8
 and mind models 280–2
 rationality, blending with irrationality 12–13, 46, 49–51, 53–6

- recognition, as non-mechanistic phenomenon 110–11, 119–20, 147–8, 161, 169, 183–4
- record area, *see* file operations: record area in
- record keys, *see* keys
- records
 in data files
 definition of 688
 duplicate 689
 and the record area 691
 in the relational model, *see* relational model: records (rows, tuples) in
- reduction
 of complex to simple structures
 and application definitions 473–4
 causing loss of alternatives 123–4, 125–6, 285–6, 289–90, 446–8, 829–31, 832–3, 867
 and economics 861
 and face recognition 110–11
 and language 112, 268, 308–9, 321
 as mechanistic delusion 104–5, 108–9, 207, 321, 417–18
 and object-oriented programming 645–6
 and reification 117–18, 119–20, 121, 122, 127, 512
 and the relational model 724–5, 727
 and structured programming 544–5, 550–2, 561–2, 581
 and the three-body system 110
 of *knowing how to knowing that* 149
 to mechanics 74–5, 76, 77, 82–3, 134–40
 of one discipline to another 82–4
 of software elements to simpler ones 531, 533–4
 of software to mechanics 28, 52–3, 72, 633 n. 12
- reductionism
 as basis of understanding 80
 causing the mechanistic myth 81
 in corpuscular theories 78
 definition of 9, 68
 failure of 81, 83–4
 formal 9, 76, 82–4
 in hierarchical structures 85–6
 informal 76
 and logical positivism 325–6
 modern 83
 multistage 82–4
 partial 9, 83–4, 284–6, 323, 831
- redundancy, data, *see* relational model: data
 redundancy and inconsistencies in
- regularities, as basis of induction and causality 211–12
- reification, fallacy of
 and abstraction 122–3
- reification, fallacy of (*cont.*)
 in behaviourism 238
 causing loss of alternatives 120–1
 in Chomskyan linguistics 257, 258–9, 268
 in classification systems 315–16, 641, 645–6
 definition of 116
 in economics 861
 examples of 119–20, 257
 in human phenomena 128, 133, 285, 307–8, 831, 861
 and indeterminism 121
 in knowledge structures 121, 154–5, 301, 315–16, 371, 400–1, 418
 and mind models 154–5
 and the new pseudosciences 107, 127, 204–5, 207, 233–4, 512
 in object-oriented programming 642, 645–6, 652
 in the relational model 682, 724–5, 779–81, 800
 in structuralism 244
 in structured programming 544–5, 550–1, 581
see also alternatives, loss of; mechanistic fallacies
- Reilly, Edwin D., *see* Ralston, Anthony
- relational model
 as abstract concept 228, 723, 799
 and abstraction 682, 683, 703, 706, 718, 723, 788, 800
 and automatic application development 683
 data redundancy and inconsistencies in
 definition of 719, 743
 due to incorrect relationships 752, 756–7, 762–3
 examples of 719, 744
 identical to traditional concepts 749, 751–2
 impossible to formalize 749–50, 757, 761, 762–3, 764–5
 misrepresented as difficult problem 749–50, 759, 764
 and data validity
 delusion of 721–2, 734–6
 guarantee of 717, 733
 and the delusion of high levels 683, 702–3, 705–6, 723–4, 736–8, 799–801
 delusions of, summarized 726–7
 denormalization in, *see* normalization, in the relational model: denormalization in
 dependency theory in, *see* normalization, in the relational model
 and the effort to discredit the traditional file systems 816

relational model (*cont.*)

- expanded in order to suppress falsifications
 - by allowing low-level operations 725-6, 801, 804
 - by allowing non-1NF 786-7, 790-1, 793-4
 - by allowing programming 228-9, 685, 806-8
 - by allowing traditional database concepts 686, 706, 727, 783-4
 - by allowing unnormalized files 228, 725-6, 794-5
- and extended relational databases 786, 791, 792-4
- failure of 706, 725, 783, 795, 799, 804, 816
- fallacies of
 - annulling the exact principles 725-6, 727, 748, 784, 793-4
 - annulling the mathematical benefits 740, 790, 791, 814-15, 816
 - annulling the restriction to high-level operations 724, 725-6, 740, 783-4, 799-800, 804, 806-8, 812-14
 - annulling the restriction to normalized files 724, 725-6, 743, 785, 786-7, 794-6, 798-9
 - annulling the restriction to whole files 737-8, 740, 800, 812-13, 814
 - confusing cause and effect in database design 781
 - defining data as valid by default 714, 733, 734
 - destroying the integration of the database and the application 705-6, 800, 811, 815
 - expanding SQL into a programming language 808-10, 812-14
 - ignoring the significance of the informal aspects 722, 734
 - misinterpreting the aspects of a database 780-1, 782-3
 - misrepresenting the concept of denormalization 795-7
 - misrepresenting the non-1NF feature 788-90, 791
 - perceiving denormalization as exact method 796-8
 - perceiving normalization as formal design principles 726, 742-3, 748-51, 761, 762-3, 764-5, 772-9, 780-1, 782-3, 797-8, 801-2
 - perceiving normalization as important theory 726, 748, 749-51, 764-7, 771, 772
 - perceiving software mechanism as substitute for expertise 682-3, 759-60, 761-2, 764, 792, 817

relational model (*cont.*)

- fallacies of (*cont.*)
 - perceiving the integrity functions as high-level operations 804, 805
 - perceiving the mathematical aspects as important quality 680-1, 721-2, 726, 728, 730, 735-7, 738-9, 741, 759-60, 791-2, 793-4
 - recognizing only one type of field dependencies 769-71, 802
 - recognizing only the database structures 680-2, 733, 734-5, 764, 768
 - reinstating programming capabilities 735, 800, 804, 806-8, 810-14, 815-17
 - reinstating the traditional database concepts 676-7, 706, 725-6, 727, 784, 785, 788, 795, 814, 815-17
 - replacing design principles with correction principles 742-3, 746, 749-51, 759, 760-3, 772-9, 798
 - replacing design principles with dependency principles 719-20, 745, 748-9, 759, 760-3, 764-5, 780-3
 - restricting database operations to whole files 682, 717, 737-8, 740
 - reversing the principles of programming 629-30, 685
 - separating the integrity functions from the application 802, 804, 805
 - separating the operations from their implementation 227-8, 722-3, 798
- field dependencies in
 - database design based on 761-3, 768-9, 772, 773-4, 775-7, 778-9, 782-3, 802
 - definition of 719, 744-5
 - examples of 719, 768, 802
 - identical to traditional concepts 720, 758-9
 - impossible to formalize 749-50, 762-3, 768-9, 772-9, 782-3
 - irrelevance of 747, 759
 - and normalization 719, 761-3, 765-7, 768-71, 772, 773-4, 775-7, 778-9, 802
 - as ordinary dependencies 770-1, 802
- fields (columns, attributes) in 714, 715, 715 n. 4, 718-19, 730, 733, 791
- file nesting in
 - definition of 788
 - as falsification of the relational model 788, 789-90, 791, 794
 - as high-level operations 788
 - as informal aspect 791
 - promotion of 788-9, 792-4
 - as replacement of JOIN operations 788-90, 793

relational model (*cont.*)file nesting in (*cont.*)

similar to traditional file nesting 788,
789–90, 793

files (tables, relations) in 682, 714, 715,
715 n. 4, 717–19, 730, 733, 735–7, 791

formal aspects of

data format as 721, 735–6

first normal form as 743, 747–8, 784

vs. informal aspects 721–2, 724–5, 731,
738–9, 748–9, 765

manipulation as 731, 735–6, 738, 760, 765

query operations as 721, 734, 738, 740, 799
restricting the model's usefulness 739,
740, 760

founded on a convention 714, 733

as fraud 685, 686, 704–5, 728, 742, 791, 800,
814, 817

as high-level operations 678, 683, 702–3,
718, 723, 736–8, 790, 799, 800–1, 812–13

as illogical idea 685, 721, 722, 738–9, 740, 760

as inappropriate use of mathematics 721,
722, 733–4, 738–9

inefficiency of 723–4, 789, 792–3, 815–16

informal aspects of

data integrity as 722, 734, 738

database design as 760, 762–3, 764–5,
780, 781–3, 796–8

database language as 722, 738

field dependencies as 748–9, 762–3, 768–9
file nesting as 791

and the need for programming 739, 760

normalization as 721–2, 726, 738, 748,
761, 762–3, 764–5, 768–9, 772–9, 784,
801–2

replacing the formal aspects 725–6, 748,
784, 785, 793–4

translation as 731, 738, 760, 765

updating operations as 721, 738, 747–8,
799

integrity functions in

as built-in operations 805

and database languages 806–7

definition of 803

examples of 802, 803, 804–5

as extension to normalization 802

as fraud 804, 806–7

identical to traditional concepts 803, 805

as low-level operations 803–4, 805–6, 807

as part of the application's logic 803

and triggers 805, 807

as unnecessary feature 802, 804

and updating operations 801, 802

and interacting software structures 227–9,
681–2, 724, 734–5, 737–8, 761, 779–80

relational model (*cont.*)

levels of abstraction in 736–8

as low levels in high-level systems 676–7,
685, 704–5, 725–6, 798, 799–801, 804,
813–14

manipulation in

definition of 730–1

as formal aspect 731, 738

as minor aspect 732, 738, 760, 765

vs. translation 731, 738–9

mathematical claims of 227–8, 680–1, 682,
717–18, 721, 725, 728, 741, 761, 791, 816

as mechanistic delusion 680–1, 713, 727,
748–9, 779

motivation for 677–8, 686

and nested relational databases 788, 791,
792–4

non-1NF in, *see* normalization, in the
relational model: non-1NF (non-first-
normal-form) in

normalization in, *see* normalization, in the
relational model

operations in, *see* relational operations

predicate calculus as basis of 713–14, 715–16,
721, 725, 729–30, 732, 746–7, 765, 791

and the problem of performance 723, 785,
789, 798

and programming incompetence 685,
759–60, 764, 792, 817

as programming substitute 678, 759–60,
761–2, 792, 817

promises of 227–8, 680, 682–3, 717–18,
759–60, 799, 816

promotion of 723, 741, 816–17

as pseudoscience 227–30, 676–7, 706, 725,
727, 784, 790–1, 795, 801

record keys in 714–15

records (rows, tuples) in 714, 715, 715 n. 4,
718–19, 730, 733, 735–7, 747, 791

and reification 682, 724–5, 779–81, 800
and similarity to the traditional data files

in file concepts 715, 720, 751–2

in file relationships 751–2, 757, 758–9,
762–3

as simple mathematical system 730, 731,
735–7, 738–9

SQL in, *see* SQL (database language)

stored procedures in

and database languages 808

definition of 808

as low-level operations 808

as means to bypass the relational
principles 808

translation in

definition of 730–1

- relational model (*cont.*)
 translation in (*cont.*)
 as informal aspect 731, 738
 as main aspect 732, 738, 760, 765
 vs. manipulation 731, 738–9
 treating files as tables in 682, 717–18, 790,
 799, 812–13
- relational operations
 benefits of 717
 definition of 715–17
 DIFFERENCE, definition of 716
 and file nesting 788
 and formal definitions 717
 as high-level operations 678, 683, 702–3,
 718, 723, 736–8, 790, 799, 800–1, 812–13
- JOIN
 definition of 716
 file nesting as replacement of 788–90
 inefficiency of 746, 785, 789, 792–3
 uses of 716–17, 746, 785
- as logical operations 715–16, 730
- PRODUCT
 definition of 716
 uses of 716–17
- PROJECTION
 definition of 716
 uses of 716–17
- SELECTION
 definition of 716
 uses of 716–17
 simplicity of 718
 as subset of the basic file operations 791,
 792 n. 9
 treating files as tables 682, 717–18, 790,
 799, 812–13
- UNION, definition of 716
- relations
 in predicate calculus 711, 732
 in the relational model, *see* relational model:
 files (tables, relations) in
- relations, between elements
 in software structures
 reflecting jumps in execution 538, 543–4,
 564–5, 567–8, 602–3
 reflecting practices 348–9
 reflecting shared data 352, 540–1, 566,
 594–5
 reflecting shared operations 355, 441,
 539–40, 566, 594–5, 667–8
 reflecting the flow of execution 543–4,
 546, 550–1, 558, 564–5, 566, 594–5,
 597–8, 602
 reflecting various processes 550–1, 563,
 655
see also software processes
- relations, between elements (*cont.*)
 in structures 92–3, 98–9, 106, 112, 164, 637
see also attributes
- relative files, definition of 688–9
- requirements, for software applications
 and automatic application development
 479–80, 481–2, 536, 600, 683
 perceived as exact specifications 293, 472–4,
 477
 reflected in the file relationships 756–8,
 759, 761–3, 764, 768–9, 772–9, 780–1,
 797–8
 specified by selecting options 453–5
- research, charlatanism in, *see* corruption: in
 universities
- responsibility
 degrading the concept of 834–6, 837–8, 845
 denied by deterministic social theories
 840–1
 denied by software theories 842
 and free will vs. determinism 823–4, 827–8,
 833–4, 836–7, 840–3
 and language 252, 838–9
see also irresponsibility
- Ritchie, Dennis M., and the GOTO debate 610
- Rousseau, Jean-Jacques, and the idea of a
 general will 851
- rows, in the relational model, *see* relational
 model: records (rows, tuples) in
- rule-following, in novice performance 157–8,
 159, 272
- Russell, Bertrand
 and empiricism 213
 and the humanist cause 321–2
 and the idea of a perfect language 301, 327,
 405
 and mechanism 70
 and the philosophy of language 321
 and the problem of induction 212
 and the rationalist philosophy 306
 and the theory of logical atomism 321,
 322–3
- Ryle, Gilbert, and mind models 148–50, 156

S

- S-R units, *see* stimulus-response units
- sailing, expertise in, vs. magic 48–9, 50
- Salomon, Albert, and scientism 132
- Sampson, Geoffrey, and Chomsky's views on
 freedom and creativity 282–3
- scanning loop, *see* file scanning loops
- scanning sequence, in data files
 ascending vs. descending 689
 definition of 689, 690

- schizophrenia
 as failure to combine knowledge structures 179–80, 185
 manifestations of 179–80
- scholasticism, as scientific philosophy 12, 68, 70, 151
- science
 based on conventions 91–2
 based on induction 211–12
 based on mechanism 69
 based on tacit knowledge 163–4
 of behaviour, behaviourism perceived as 235, 236
 empirical
 and mathematics 209
 and the problem of induction 213
 vs. pseudoscience 209
 vs. software theories 209–10
 identified with mathematics 88–9
 identified with mechanism 7, 10, 14, 88–9, 233–5
 mechanism as method of 9, 69, 274–5, 513
 of programming, software theories perceived as 516, 523, 617–18
 as pursuit of sound theories 206
 similar to magic 46
- Scientific Revolution, and the mechanistic philosophy 15, 25, 68, 74–5, 81–2, 88, 130
- scientific theories, falsifiable nature of 210, 216–18
- scientism
 in biology 136–7
 causing the new pseudosciences 204
 causing utopianism 20, 130, 850
 definition of 129
 due to mechanism 20, 127–8, 129–30, 134–8, 141, 161–2, 836–7, 850
 in economics 133, 135–6, 861
 and the idea of natural order 850
 in linguistics 76, 112, 233, 243, 251, 838–9
 and operationalism 139
 in politics 137–8, 243, 836, 850
 in psychology 135, 139–40, 233, 235–6, 243
 in sociology 135, 139–40, 233, 243, 840–1, 850
 views on 129
- scientists, as charlatans, *see* corruption: in universities
- selling, deception in 22, 845
- sentences
 logical analysis of 319–20
 meaning of 256, 259, 311–12, 319–20
- sequence
 of execution, *see* flow of execution
 of operations, static vs. dynamic 518, 543
- sequential constructs, *see* software constructs, flow-control: sequential
- sequential files, definition of 688–9
- service files
 data files as, used to relate files 701, 753–4, 773, 777–8
 index files as, *see* index files
- shared data, as software principle 353, 542–3
 shared operations, as software principle 355, 542–3, 667–8
- Shlaer, Sally
 and mathematical programming 632–3
 and object-oriented programming 633
- Simon, Herbert A., and artificial intelligence 145
- simple structures
 alternatives in 93, 113–15, 283–6
 attributes in 11–12, 95–6, 98–9, 100, 106, 362
 definition of 90, 92
 determinism of 99, 277–8
 as mechanistic approximation 11, 106, 145, 148, 233–4, 238–9, 474, 550–2, 581, 826, 828–31
 and the mind 164, 193–4, 416–18
 as models, *see* mechanistic models
 used for both analysis and synthesis 94–5
see also hierarchical structures
- skills
 blending with magic 47–9, 50–1
 blending with software magic 51–2, 54–5
 diverse, capacity for 170, 271–2, 416–17, 428
 holistic knowledge in 157–60, 272
 levels of 157–60, 416–17
 natural 159–60, 166, 271
 supplemented with magic 45–6, 50–1
 supplemented with software magic 51–2, 54–5
see also knowledge; learning
- Skinner, B. F.
 and behavioural technology 279
 denying creativity and free will 279
 and high-level stimulus-response units 241–2
 and the Skinner box 237, 241
- slogans, used to impoverish knowledge 372–3, 385–6, 394, 395–6, 398, 863–4
- social forces
 as deterministic explanation of historical evolution 840–1
 as idea, circularity of 840
- society
 corrupted
 by mechanism 437–8, 818, 859–61
 by software mechanism 439, 507, 847–8

society (*cont.*)corrupted (*cont.*)

by software totalitarianism 411–13, 490–1, 847–8, 871–2

by totalitarianism 846, 865–6, 869, 871

inefficiency in 30, 38–9, 483, 490–1, 504

perceived as deterministic system

in the new pseudosciences 833–4, 854–5
under totalitarianism 852

perfect, *see* perfect society

primitive, *see* primitive society

sociology

and determinism 840–1

and the idea of social forces 840–1

and scientism 135, 139–40, 233, 243, 840–1, 850

software

controlled by the elites 409–10, 411–13, 421

deception through, *see* software
charlatanism

dependence on 298–9, 409–10, 411–13, 432, 437–8

easy to use

due to high levels 433, 449–50

as mechanistic delusion 432–4, 449–50

as environment, *see* development
environments; software devices

impoverishment of 126, 290–2, 367, 832, 867–8, 870–1

as interacting structures, *see* interactions,
of software structures

manipulated by the elites 370, 408–9, 870–1

as means to represent the world 256, 300–2,

345–6, 368–9, 408–9, 411–13, 542–3,

642–3, 652, 847, 868, 870

mechanistic, *see* mechanistic software

and the mechanistic fallacies, *see*

mechanistic fallacies: in software

as mechanistic myth 26, 28, 31, 33

as model of phenomena 105

as myth, *see* software myth

as non-mechanistic phenomenon

and the fallacy of software engineering

501–3, 515

and inadequacy of software devices 832, 867

and interacting structures 105, 125–6,

360–3, 364, 441–3

and similarity to language 304–5, 326–7

and software theories 287, 597–9, 642–3

and Wittgenstein's later philosophy 345–7

perceived as mapping system 300, 304–5,

326–7, 329, 335, 345–6, 459, 642, 652

perceived as mathematical concept 89,

501–2, 597–9, 600, 632–3, 725

software (*cont.*)

perceived as products 42 n. 1, 429, 866–8

perceived as supernatural phenomenon,
see software magic; software power

potency of 31

promoted through confirmations rather
than falsifications 222, 398, 514, 627,
674

reducing to mechanics 28, 52–3, 72, 633 n. 12

similar to language

and Chomskyan linguistics 256, 263 n. 17

as means to represent the world 298–9,

300–2, 345–6, 347, 368–9, 640–1, 642,
652

and the mechanistic fallacies 292, 299–

300, 326–7, 366–7, 370, 408–9, 421,

429–31, 459–60, 706

and non-mechanistic knowledge 67,

370, 411–13, 421, 431–2, 442–3

perceived as mapping system 300–2,

304–5, 318, 326–7, 329, 335, 345–6, 459

perceived as simple structures 300, 318,

344–5, 364, 642, 651–2

and responsibility 468, 839

and the search for a perfect system 302,

326–7, 344–6

and software totalitarianism 847, 868,

870, 872

used to control minds 125, 368, 370,

408–10, 411–13

and Wittgenstein's shift in philosophies

329, 344–6

tested using the falsification principle 223

turned into morbid preoccupation 874

software applications

aspects of 348, 355, 419, 440–1, 444, 445–6,
448, 472, 521, 542, 628, 680–2

created by their users 37–9, 487

and databases, integration of 705, 800, 811

and definitions, *see* definitions, of software
applications

and the delusion of life cycles 478–9

and the delusion of perfect applications

477–8

development of 470, 472–4

as interacting structures

and application definitions 472–4

and the delusion of mathematical pro-

gramming 89–90, 597–9, 600, 681–2

and the delusion of the main structure

345–6, 350, 363–4, 544–5

and file operations 702

and the need for low levels 445–8, 469,

628–30, 648–9, 683, 702–4, 705, 724,

737–8, 800, 803–4

- software applications (*cont.*)
- as interacting structures (*cont.*)
 - and object-oriented programming 642–3, 648–50, 652, 653–4, 659, 662
 - and the relational model 227–9, 724, 734–5, 737–8, 761
 - and similarity to language 364, 442–3, 551–2, 628–9
 - and software processes 347–8, 350, 352, 355, 360–3, 365, 440–1, 542, 559–60, 702
 - and structured programming 521, 537, 567–8, 591–2, 593–7
 - and Wittgenstein's later philosophy 345–7, 360–3
 - loss of alternatives in, *see* alternatives, loss of: in software applications
 - and maintenance, *see* maintenance, of software applications
 - and methodologies, *see* methodologies
 - perceived as simple structures, *see* software structures: perceived as simple structures
 - ready-made
 - and the delusion of high levels 470–1
 - inadequacy of 470–2
 - and the mechanistic fallacies 126
 - and software totalitarianism 868
 - and requirements, *see* requirements, for software applications
 - and software innovations 472, 476
- software blocks
- definition of 517 n. 1
 - as implicit jumps 520, 543–4
 - as software elements 542, 554–5
- software bureaucracy
- aspects of 40–1, 414–15
 - due to ignorance 29, 31–2, 421, 818–19
 - due to programming incompetence 34–5, 37–8, 198–9, 411–12, 679–80
 - due to software mechanism 29, 40–1, 200, 536, 676
 - due to the elite 36–7, 676, 819–20
 - spread of 34–5, 37–9, 490–1, 493, 847
 - and structured programming 516, 579, 601–2, 625
- software bureaucrats
- managers as 34–5
 - programmers as 32, 35–6, 414–15, 468, 486, 493, 498–500, 625
 - similar to primitives 819–20
 - software users as 37–9, 487–9
 - workers as 37–9, 487–9
- software charlatanism
- and the delusion of high levels 448, 449–53, 465–8, 628–30, 650
- software charlatanism (*cont.*)
- due to programming incompetence 37, 115–16, 444, 456–7, 817
 - due to software mechanism 65–7, 72, 190, 274, 369–70, 408–9, 440, 461, 741–2
- software claims, assessing 46, 208
- software companies
- contributing to the software bureaucracy 35–6
 - exploiting the mechanistic delusions 190, 419–20, 421, 427–9
 - fostering programming incompetence 35–6, 456–7, 466, 468–9, 819–20, 820 n. 2
 - justified by the software myth 36, 469–70, 867
 - see also* software elites
- software constructs, flow-control
- and application maintenance 519–20
 - built-in
 - vs. constructs using GOTO 564–5, 581–3, 591, 602–3, 611–13, 616–17, 618, 619
 - as language enhancements 611–13
 - perceived as substitute for expertise 619–21
- CASE
- definition of 581
 - as non-standard conditional construct 581, 589
- conditional
- definition of 517, 525
 - and the elimination of ELSE 615
 - equivalent to sequential constructs 556–7
 - non-standard 520–1, 538–9, 582–3
 - reducing to iterative constructs 572–4, 577
- definition of 517, 525–6
- elementary
- and Böhm and Jacopini's work 572–5, 576–7
 - and transformations 572–5, 576–7
- enhancing the nesting scheme 562, 565, 567
- and file operations 690, 694–7, 698–9, 704, 705, 738, 813
- and flow-control attributes 594–5
- implicit jumps in 520, 529, 543–4, 553, 558, 564–5
- and interacting structures 559–60, 593–6, 597–8, 666
- iterative
- definition of 517, 526
 - equivalent to sequential constructs 556–7
 - non-standard 520–1, 540, 581–3

- software constructs, flow-control (*cont.*)
 language-specific, *see* software constructs,
 flow-control: non-standard,
 permission of
 misinterpreted as operations between
 elements 548, 553-6, 560-1, 592-3,
 613, 619
 and multiple flow-control structures 548,
 553-4, 556-9, 560-1, 590-1, 593, 597-8,
 666
 nesting of 525-8, 582, 697
 and the nesting scheme, *see* structured
 programming: nesting scheme in
 non-standard, permission of
 to avoid GOTO statements 581-3, 586,
 589, 600, 602-3, 606, 611-13
 to avoid transformations 580, 581-3, 589
 justifying 581-2, 584-6, 589-90, 611-12
 perceived as higher levels 592, 619
 and similarity to standard constructs
 590-1, 593, 602
 and object-oriented programming 663, 666
 perceived as a minimal set of operations
 560-1, 566
 sequential
 definition of 525
 as operations in flow-control structures
 554-5, 593
 as source of programming errors 518-20
 standard, restriction to
 and absence of GOTO statements 529,
 564-5
 as basic principle 520-1, 525
 impracticality of 566-7, 568-9, 571
 inconvenience of 580-3, 586-8, 592
 mathematical claims of 531-2
 self-contradiction of 574-5
 and transformations 537-41, 562-3, 602-3
 software consumers, turning workers into 38-9
 software crisis
 aspects of 494, 632, 634
 due to software mechanism 504, 635
 due to the software theorists 504
 and the idea of software reuse 640
 origins of 504
 and the prevention of expertise 515, 521,
 618-19
 spread of 504
 software delusions, due to mechanistic
 delusions, *see* mechanistic delusions:
 causing software delusions
 software devices
 adequate for mechanistic problems 66,
 195-7, 417-18, 419-20
 based on mind models 190, 293-4, 832
 software devices (*cont.*)
 degrading minds through 190, 411-13,
 423-5, 433-5, 832-3, 834-5
 dependence on, *see* dependence: on
 software devices
 emulating manufacturing concepts 289
 as frauds 65-7, 629-30
 human beings perceived as operators of
 486-7, 489, 490, 493, 500
 inadequate for non-mechanistic problems
 65-6, 195-7, 417-18, 419-20, 422-3, 469
 and irresponsibility 468, 486-9, 839
 as low levels in high-level systems
 examples of 451
 inducing dependence 449-52, 462-3,
 628-30, 705
 and non-procedural languages 452-3,
 809-10
 and object-oriented programming 648-9,
 650, 675-6
 and the relational model 676-7, 685,
 704-5, 725-6, 798, 799-801, 804, 813-14
 by selecting options 453-5
 as mechanistic delusions 27, 190, 444,
 628-30
 vs. non-mechanistic capabilities 66-7,
 188-9, 190, 196-7, 290-2, 417-18,
 423-5, 437-8, 439, 440, 456, 833
 replacing knowledge with selection of
 options 293, 296, 454-5, 460-1, 488,
 489, 832
 as substitutes for creativity 172, 282, 293-4,
 296-7, 832-3
 as substitutes for knowledge, *see* knowledge
 substitutes
 as substitutes for programming expertise,
see programming substitutes
 vs. traditional devices 418
 software diagrams
 classification-style vs. flowcharts 364
see also flowcharts
 software domination
 consuming people's time 421-2
 degrading minds through 413, 427-8, 868
 due to mechanistic delusions 31, 419, 456-7
 fostering ignorance and dependence 421,
 427-9, 449-52, 465-8, 475-6, 490-1, 493
 through high-level systems 449-52, 468-9
 as new phenomenon 410
 software elites
 causing the software bureaucracy 36-7,
 676, 819-20
 as charlatans 65-7, 115-16, 291, 456-7, 524,
 650, 674, 741-2, 817, 843
 consuming people's time 421-2

software elites (*cont.*)

- controlling software 409–10, 411–13, 421
- enforcing mechanistic knowledge, *see* non-mechanistic knowledge: prevention of
- exploiting society 65–7, 115–16, 411–13, 456–7, 490–1
- fostering dependence, *see* dependence: fostered by the software elites
- fostering ignorance 427–9, 489–90, 493, 500, 819–20
- fostering irresponsibility 468, 620–1, 624–5, 819–20
- fostering programming incompetence 413, 416, 431, 486–7, 618–19, 620–2, 624–6, 675–6
- justified by the software myth 26, 27, 31, 36–7, 847, 867
- manipulating software 370, 408–9, 870–1
- preventing non-mechanistic knowledge, *see* non-mechanistic knowledge: prevention of
- promoting knowledge substitutes, *see* knowledge substitutes: promoted by the elites
- promoting programming substitutes 38–9, 147, 192, 449–52, 486–7, 531–3, 618–19, 650, 759–60
- promoting software magic 43–4, 821
- promoting software mechanism 29, 32–4, 36, 290–4, 295–6, 500–3, 531–3, 630–2, 742, 870–1
- promoting totalitarian concepts 286, 838, 843, 847, 868–9, 871

software engineering

- based on mechanism 28, 500–3, 507
- definition of 494, 508
- as idea, circularity of 495–7, 499, 505, 508–9
- as pseudoscience 492, 509, 627
- resembling astrology 510
- resembling superstitions 509–10
- transition to 494–5, 499, 505
- as wish 496, 509
- see also* software theories

software evolution

- conforming to 842–3
- explained as deterministic process 842–3
- as justification for software devices 842–3

software experts, *see* software theoristssoftware exploitation, *see* software domination

software failures

- due to irresponsibility 486, 821–2
- due to programming incompetence 32–3, 483–5, 821–2
- due to software mechanism 33, 288–90, 484–5, 504, 627

software failures (*cont.*)

- manifestations of 821, 822–3
 - rationalized through software mechanism 483–5, 486
 - suppressed through propaganda 822
- software forces
- as deterministic explanation of software evolution 842–3
 - as idea, circularity of 842–3
- software freedom
- and the delusion of high levels 449–52, 462–3, 465–8, 629–30, 705
 - and the fallacy of abstraction 97–8
 - and similarity to language 411–13
 - and the software myth 410
- software magic
- blending with skills 51–2, 54–5
 - and the definition of software engineering 509 n. 4
 - emulating manufacturing concepts 52–3
 - as extension to skills 51–2, 54–5
 - and programming incompetence 43–4, 54–5, 819, 821
 - promoted by the elites 43–4, 821
 - see also* magic
- software mechanism
- causing charlatanism, *see* software charlatanism
 - causing irrationality 31–2, 53–4, 614
 - causing irresponsibility 40–1, 818, 834–5
 - causing programming incompetence 32–3, 126, 197, 291, 294–5, 456–7, 483–5, 504, 625–6, 675–6
 - causing software failures 33, 288–90, 484–5, 504, 627
 - causing software-related problems 195–7, 420, 426, 483–5, 487–8
 - causing spurious preoccupations 195–7, 199, 420, 421, 426–7, 483–5
 - causing the software bureaucracy 29, 40–1, 200, 536, 676
 - causing the software crisis 504
 - causing totalitarianism, *see* totalitarianism: due to software mechanism
 - causing utopianism 133, 874
 - as commitment 439
 - corrupting societies 439, 507, 847–8
 - cost of 33, 126, 457–8
 - degrading minds through 367, 411–13, 429–31, 437–8, 870–1
 - degrading the concept of responsibility 835–6
 - as delusion 194–5, 481–5, 627
 - denying creativity and free will 292–4, 296–7, 438, 831–2, 842

- software mechanism (*cont.*)
 as dogma 26, 503, 504
 early successes of 500–1, 503
 first stage in spread of 486–7, 490, 492
 as ideology, conforming to 199, 411–13,
 437–8, 490–1, 842–3, 868–9, 874–5
 impoverishing knowledge 367, 847–8, 871
 and mind control 367, 411–13, 429–31, 437–8,
 870–1
 perceived as answer to programming
 incompetence 515, 520–1, 618–19,
 625–6, 675–6, 760, 764
 preventing non-mechanistic knowledge
 66–7, 370, 437–8, 488, 489, 504, 532–3,
 594, 599–600
 promoted by the elites 29, 32–4, 36, 290–4,
 295–6, 500–3, 531–3, 630–2, 742, 870–1
 proven invalid by the failure of CASE 481–3
 second stage in spread of 487–9, 490, 492
 similar to traditional mechanism 274, 514,
 515
 spread of 483–5, 486–91, 492
 third stage in spread of 489–90, 492
 as trap 66, 190, 419, 424–5, 456, 833, 875
 used to rationalize failures 483–5, 486
see also software theories
- software metrics, as mechanistic delusion
 600 n. 19
- software modules
 perceived as independent 355–6
 as software elements 353–4, 542, 554–5
 used as subroutines 355–6
- software myth
 as basis of totalitarianism 31
 definition of 26
 due to ignorance 28, 31–2
 fallacies of 27–8, 410
 and irresponsibility 29, 410, 839
 as justification for elites 26, 27, 31, 36–7,
 847, 867
 as justification for software companies 36,
 469–70, 867
 and programming 27
 promoted through propaganda 31, 39–40,
 196, 415
see also mechanistic myth
- software names
 evoking magic 42–3
 evoking mana 59
- software phenomena
 alternatives in 288, 831–2, 867
 as human phenomena 10, 89, 127, 289–90,
 501
 indeterministic nature of 89–90, 127, 287,
 289–90, 831–2
- software phenomena (*cont.*)
 perceived as mechanistic 831–2, 866–7
 perceived as supernatural, *see* software
 magic; software power
- software power
 high levels perceived as 433, 449–50
 knowledge substitutes perceived as 62, 819
 as mechanistic delusion 432–4, 449–50
 similar to mana 58–60, 61–2
 as universal potency 62
see also mana; power
- software practitioners, *see* programmers
- software principles
 databases as 351–3
 definition of 347, 542–3
 flow-control as 543
 memory variables as 353
 objects as, in object-oriented programming
 667–8
 practices as 348–50, 542–3
 shared data as 353, 542–3
 shared operations as 355, 542–3, 667–8
 subroutines as 353–4, 355–6, 667
- software processes
 and attributes 348, 360–3, 542, 563–4
 databases as 352, 361, 440–1, 702
 definition of 348, 542
 and the delusion of high levels 193, 419,
 446–8
 external subroutines as 355
 flow-control features as 543
 and flow-control structures 559–60
 and interacting structures 192–3, 347–8,
 350, 352, 355, 360–3, 365, 440–1, 542,
 559–60, 702
 objects as, in object-oriented programming
 667–8
 practices as 348–50, 361, 440–1
 repeated operations as 354–5
 and structures 348, 360–3, 542
 subroutines as 353–4, 355–6, 361, 440–1
- software propaganda
 and the media 222, 536–7, 627, 674, 822
 promoting ignorance 66–7
 promoting the software myth 31, 39–40,
 196, 415
 redefining the meaning of expertise 56,
 415, 868
 suppressing failures 822
- software-related affairs
 impoverishment of 115–16, 126, 291–2, 296
 incompetence in 31–2, 41, 427–8
 irresponsibility in 41, 834–5, 839, 842
 non-mechanistic knowledge in 66–7, 193,
 427–8, 480–2, 488, 489

- software-related problems
 due to software mechanism 195–7, 420, 426, 483–5, 487–8
 spurious nature of 195–7, 199, 420, 421, 426, 483–5
- software structures
 and classification systems 349, 352, 354, 360–3, 441, 542
 databases as 352, 361, 440–1, 702
 flow-control, *see* flow-control structures
 interacting, *see* interactions, of software structures
 perceived as simple structures
 and the delusion of high levels 290
 and the delusion of mathematical programming 89–90, 501–2, 531–2, 680–3
 denying creativity and free will 292–4, 831–2
 due to software mechanism 514–15, 531–2, 550–2, 580–1
 emulating manufacturing concepts 27, 287, 289, 364, 561–2
 in object-oriented programming 636–7, 642–3, 645–6, 652–4
 in the relational model 680–2, 724–5, 727, 733, 737, 768, 779–80
 and similarity to language 300, 302, 551–2, 642, 651–2
 and software totalitarianism 866–7
 in structured programming 525–8, 531–2, 533–5, 536, 547–8
 and Wittgenstein's early philosophy 335
 and Wittgenstein's later philosophy 345–7
 practices as 348–9, 361–2, 440–1
 and software processes 348, 360–3, 542
 subroutines as 353, 361–2, 440–1
 software systems, honest vs. dishonest 462–3, 465–8
- software theories
 based on false assumption 32, 49–50, 52
 based on hierarchical structures, *see* software structures: perceived as simple structures
 based on mind models 146–7, 192, 293–4, 834
 and the delusion of the main structure 345–6, 353–4, 363–4, 550–2
 and determinism 288, 292–4, 831–2, 841–3
 emulating engineering concepts 86, 494–6, 503–4, 506–7, 508–9, 633
 emulating manufacturing and construction concepts 27, 32, 34, 52, 72–3, 77, 289, 364, 443, 632, 633, 635, 866–7
 failure of
 due to mind mechanism 146–7
- software theories (*cont.*)
 failure of (*cont.*)
 due to software mechanism 12, 287–90, 481–4, 514–15, 530, 599–600, 626, 627, 635, 675–6, 681–2
 as logical systems 209–10
 reinstating the traditional concepts 515, 626, 676
 as frauds 29, 469, 549, 651
 as informal concepts 506
 as mechanistic delusions
 and application definitions 472–4
 and the delusion of high levels 444, 503, 626, 628–30, 675–6
 and the delusion of mathematical programming 89–90, 501–2, 523–4, 531–2, 597–9, 599 n. 18, 600, 632–3, 680–3, 725, 731–7, 738–9, 741, 765–7, 791–2
 perceiving software as simple structures 335, 436–8
 pseudoscientific nature of 29, 514–15, 627, 649
see also object-oriented programming: as mechanistic delusion; relational model: as mechanistic delusion; structured programming: as mechanistic delusion
 and non-mechanistic phenomena 125–6, 194–5, 206, 867
 as pseudosciences 29, 206, 209–10, 436–8, 451, 510–11, 514–15, 549, 626–7, 651, 674
 reflected in software devices 293, 295–6
 similar to deterministic social theories 841–3
 similarity of 28, 72, 515
 as substitutes for programming expertise, *see* programming substitutes
 and Wittgenstein's early philosophy 329, 335, 344–5
see also object-oriented programming: relational model; software mechanism; structured programming
- software theorists
 causing the software crisis 504
 as charlatans 37, 147, 507, 549–50, 617, 651, 672, 741–2, 817
 and circularity of software engineering concepts 495–7, 499, 505, 508–9
 comparing programmers to craftsmen 497–9, 505, 635–6, 675
 and the delusion of software mechanism 501–4, 529–30, 534–5, 579, 597–9, 650, 680–2
 early contributions of 500–1, 507
 lacking engineering knowledge 86, 506
 lacking programming experience 500, 579

- software theorists (*cont.*)
 and methodologies 475
 naivety of 516, 547, 577, 592, 617–18, 664,
 681–2, 727, 748–9, 799
 perceiving software as simple structures
 86, 506–7, 514–15, 531–2, 550–2, 580–1,
 680–2
see also software elites
- software totalitarianism
 as conformism to the mechanistic software
 ideology 490–1, 868–9, 874–5
 corrupting societies 411–13, 490–1, 847–8,
 871–2
 extending the scope of software 874
 and Orwell's model 412–13, 848, 869–71,
 872–5
 similar to political totalitarianism 413,
 490–1, 847, 848, 869, 871, 872–5
 and Talmon's model 848, 866–9, 872–5
see also totalitarianism: due to software
 mechanism
- software users
 creating their own applications 37–9, 487
 dependent on software devices 56, 487–90
 similar to primitives 65–7
 as software bureaucrats 37–9, 487–9
 spurious activities of 483, 485
- Sorokin, Pitirim A., and scientism 138–41
- sorting sequence, in data files, definition of
 689
- space technology, analysis of term 397
- Spinoza, Baruch
 and hierarchical structures 306–7
 and language structures 310
 and material world vs. mental world 306–7
- SQL (database language)
 annulling the mathematical benefits 814–15
 and the delusion of high levels 809–10
 embedded, definition of 814
 expansion of
 through embedded SQL 814
 through explicit file scanning loops
 812–13, 814, 815–16
 as falsification of the relational model
 228–9, 800, 808–10, 812–14, 815–16
 through general programming features
 813–14, 815–16
 through SELECT features 809, 813, 814
 through updating operations 810
 and implicit file scanning loops 810–13
 as low levels in high-level systems 453,
 809–10, 812–13
 as means to suppress falsifications 228–9,
 800, 808
 as non-procedural language 809–10
- SQL (database language) (*cont.*)
 operations in
 and the cursor 812
 DELETE 810–11
 FETCH 812, 814
 INSERT 810–11
 SELECT 808–9, 810 n. 27, 811, 812–13
 similar to the traditional file operations
 811–12, 814, 815–16
 UPDATE 810–11
 as query language 808, 812–13
- standard constructs, *see* software constructs,
 flow-control: standard, restriction to
- standards, for programming expertise 34, 197,
 621, 822, 822 n. 3
- statements, in programming languages,
 definition of 517 n. 1
- static sequence, *see* sequence: of operations,
 static vs. dynamic
- static structure, *see* structures: static vs.
 dynamic
- Steiner, George
 and the idea of a perfect language 304
 and modern myths 4–5
- stimulus-response units
 definition of 236
 as elements of structure 238–9
 failure to reduce behaviour to 240–2
 modified to explain complex acts 241–2
- stories
 aspects of 351, 357, 401, 442–3
 as interacting structures 350–1, 357–9, 401,
 442–3
 and non-mechanistic knowledge 357–9,
 442–3
- structuralism
 analysis of art and literature in 245
 analysis of myths in 244–5
 based on mechanism 243
 binary operations in 244–7, 248
 computer analogies in 246
 definition of 242–3
 denying creativity 280
 failure of 250
 mathematical pretences of 246, 249, 250
 as myth 4–5
 as pseudoscience 242, 247, 280
 and reification 244
 and simple structures 277
 stemming from language mechanism 244
 transformations in 245–6, 248, 249–50
- structured design, *see* structured
 programming: top-down design in
- structured programming
 and abstraction 525, 531, 592, 613

- structured programming (*cont.*)
- black-box principle in
 - definition of 587 n. 14
 - as degradation of the original principles 584, 587–8, 589–90, 673
 - inappropriate use of 587 n. 14
 - as justification of non-standard constructs 584, 587–8, 591
 - and the nesting scheme 526–8, 587–8, 591
 - and Böhm and Jacopini's work, *see* Böhm and Jacopini, work of
 - and CASE 535–6, 600
 - causing the software bureaucracy 516, 579, 601–2, 625
 - and the delusion of the main structure 363–4, 529, 544–5, 550–2
 - delusions of, summarized 548–9
 - dominating programming practices 516, 579
 - emulating manufacturing concepts 531, 545–6, 561–2
 - expanded in order to suppress falsifications
 - by allowing GOTO statements 583, 603, 608–10
 - by allowing informal principles 578–9, 672
 - by allowing multiple flow-control structures 552–3, 667
 - by allowing non-standard constructs 581–3, 584–6, 590, 601
 - by allowing transformations 530
 - failure of
 - denied by the software theorists 523–4, 530, 552, 579, 580–1, 584–6, 599, 601, 603 n. 1, 616, 626–7
 - and the failure of object-oriented programming 667, 672, 675–6
 - fallacies of
 - attempting to represent applications mathematically 597–9, 599 n. 18, 600
 - blaming GOTO for bad programming 520–1, 522–3, 620
 - ignoring all systems of jumps and labels 621, 624–5
 - increasing the complexity of software 563, 568, 594–7
 - misinterpreting the inconvenience of the transformations 580–1, 584–7, 592–3
 - misinterpreting the nesting scheme 546, 547–8, 550, 552, 591–2
 - misinterpreting the role of flow-control constructs 548, 553–6, 560–1, 592–3, 613, 619
- structured programming (*cont.*)
- fallacies of (*cont.*)
 - misrepresenting Böhm and Jacopini's work 523, 566–7, 568–9, 571–7, 584–5, 605
 - perceiving built-in constructs as substitute for expertise 520–1, 567–8, 619–21
 - perceiving software mechanism as substitute for expertise 516, 531–3, 596–7, 599–600, 626, 675–6
 - promising error-free programming 534–5
 - recognizing only one flow-control structure 534–5, 559, 561, 567–8, 581, 591, 592, 663
 - recognizing only the flow of execution 521, 550–2, 581, 663
 - replacing the original principles with the black-box principle 584, 587–8, 589–90, 673
 - replacing the original principles with the GOTO preoccupation 603, 605, 606, 607, 608–10, 611–13, 616–17, 618–19
 - replacing the original promises with trivial concepts 578–9, 604–5, 606–7, 672
 - flow-control constructs in, *see* software constructs, flow-control
 - flow of execution in
 - as complex structure 556–8, 567–8, 590
 - and flow-control attributes 543–4, 564–5, 593–4
 - and the GOTO debate 602–3, 612–13, 616, 619
 - and the nesting scheme 528–9, 545–6, 550, 591–2
 - perceived as simple structure 537–8, 553–6, 663
 - see also* flow-control attributes; flow-control structures; software constructs, flow-control
 - formal theory of 525–8, 577, 578, 605, 616
 - as fraud 549, 577, 592, 601
 - GOTO statement in
 - as aspect of programming style 603, 616, 618, 621
 - and the attempt to replace it with other jumps 606, 607–8, 611–13
 - blamed for bad programming 520–1, 522–3, 620
 - vs. built-in constructs 564–5, 581–3, 591, 602–3, 611–13, 616–17, 618, 619
 - and the contradictory claims 607, 608–10
 - definition of 518
 - and flow-control attributes 543

structured programming (*cont.*)GOTO statement in (*cont.*)

- formal arguments against 529, 619
- informal arguments against 522-3, 603-5, 608-10, 611-13, 619-20
- and the informal theory vs. formal theory 616-17
- as means to control programming practices 621-2, 625
- needed for explicit jumps 518, 538-41, 543, 602-3
- as ongoing debate 586, 597, 600-1, 606
- permission of 583, 603, 608-10, 614
- preoccupation with, as substitute for the original principles 603, 605, 606, 607, 616-17, 618-19
- prohibition of 538-41, 577, 602-5, 611-13, 617, 618-20
- as superstition 507, 516, 614, 616
- and systems of jumps and labels 622-5
- and the use of humour 614-15
- and hierarchical structures 525-8, 547-8, 550-2, 561-2, 580-1, 618
- informal theory of, vs. formal theory 522-4, 577, 578, 616-17
- and interacting software structures 521, 537, 567-8, 591-2, 593-7
- as mass madness 601, 612-13
- mathematical claims of 523-4, 528, 531-2, 534, 565, 567, 587, 593-4, 597-9, 600
- as means to control programming practices 34-5, 35 n. 2, 621-2, 625-6, 675-6
- as mechanistic delusion 529-30, 544-7, 550-2, 561-2, 567-8, 580-1, 590-1, 599-600, 617, 667
- and misuse of the term "structured" 607
- and modularity 528, 533-4
- nesting scheme in
 - and the black-box principle 526-8, 587-8, 591
 - concept of, as convention 545, 558
 - and flow-control attributes 545, 558, 560-1, 564-5, 594
 - and nesting of constructs 517, 525-8
 - and nesting of modules vs. nesting of constructs 528
 - overridden by jumps 547-8, 553, 558, 564-5, 590-1, 593, 602
 - reflected in the flow diagram 526-8, 545, 558, 591, 663
 - static vs. dynamic 558-9, 663
 - and top-down design 526-8
- non-standard constructs in, *see* software constructs, flow-control: non-standard, permission of

structured programming (*cont.*)

- perceived as a new science 523, 577, 617-18
- perceived as a revolution 515, 536-7
- as programming discipline 531-2, 537-8
- and programming incompetence 516, 618-22, 625-6, 674-6
- promises of 521, 529, 531-2, 533-6
- as pseudoscience 549, 552, 579, 590, 601, 626-7, 667, 672
- and reification 544-5, 550-1, 581
- requiring non-mechanistic knowledge 593-7, 676
- and restriction to mechanistic knowledge 532-3, 593-4, 599-600
- standard constructs in, *see* software constructs, flow-control: standard, restriction to
- and the structure theorem 523
- and structured programs 528, 537, 577, 587
- and structures, static vs. dynamic 545-7, 558-9, 591, 597-8, 663-4
- top-down design in
 - benefits of 353, 528, 533-4, 537-8
 - definition of 526-7
 - practice of 526-8, 531, 533-4
- transformations in
 - to avoid GOTO statements 539-41, 565, 566, 600
 - and elementary constructs 572-5, 576-7
 - as falsification of the theory 530, 539-41, 563
 - and flow-control structures 562-3, 565-7, 594-5, 602-3
 - futility of 567-8
 - impracticality of 566-7, 568-9, 571, 594-6, 597
 - inconvenience of 580-3, 586-8, 592, 594-7, 599
 - as mechanistic fantasy 571
 - misrepresenting Böhm and Jacopini's work 566-7, 568-9, 571-7
 - motivation for 563-5
 - replacing flow-control attributes 561, 565-6, 594-5
 - by sharing data 540-1, 565-6, 595
 - by sharing operations 539-40, 565
- structures
 - complex, *see* complex structures
 - hierarchical, *see* hierarchical structures; simple structures
 - imaginary 117-18, 119-20, 121, 122, 357, 362
 - interacting, *see* interactions, of language structures; interactions, of software structures; interactions, of structures
 - in language, *see* language structures

structures (*cont.*)
 real 117–18, 119–20, 121, 122
 simple, *see* hierarchical structures;
 simple structures
 in software, *see* software structures
 static vs. dynamic 545–7, 558–9, 559 n. 1,
 591, 597–8, 663–4
 systems of, *see* complex structures;
 interactions, of structures

subroutines
 external
 as high-level operations 684
 as software processes 355
 as high-level elements 463
 as software principle 353–4, 355–6, 667
 as software processes 353–4, 355–6, 361,
 440–1
 as structures 353, 361–2, 440–1

substitutes, *see* knowledge substitutes;
 programming substitutes

supernatural forces
 belief in 59–60, 65
 evoking 43–4, 59–60

supernatural phenomena, software as, *see*
 software magic; software power

superstitions
 based on induction 211–12, 220–1
 and the fallacy of software engineering
 509–10
 and the GOTO debate 507, 516, 614, 616
 resembling rational pursuits 13, 509–10

Swanson, Guy E., and mana 57, 59, 62

Swift, Jonathan, and language mechanism
 317–18, 458–60

syllogistics, as logic system
 description of 707–8
 examples of 707

symbolic logic, definition of 707

synthesis, in simple structures 94–5

systems, deterministic, *see* deterministic
 systems

systems, of structures, *see* complex structures;
 interactions, of structures

T

tables, in the relational model, *see* relational
 model: files (tables, relations) in

tacit knowledge
 as basis of mathematics 163–4
 as basis of non-mechanistic mind models
 170–1
 as basis of science 163–4
 and complex structures 160–4
 definition of 160–1

tacit knowledge (*cont.*)
 examples of 161–4, 165–6, 167–9, 271–3
 and intuitive knowledge 171
 as normal human capability 169–70
see also non-mechanistic knowledge

Talmon, J. L.
 and liberal democracy vs. totalitarian
 democracy 848–9, 873
 and the paradox of freedom 852–3
 and scientism 138, 850
 totalitarian model of, *see* totalitarianism:
 Talmon's model of

Taylor, David A.
 and the industrialization of software 634–5
 and object-oriented programming 630,
 631, 634–5, 674

technology
 as abstract term 386
 definition of term 386
 misuse of term
 analysis of 388–9, 390–1, 394
 in company names 392–3
 deception through 390–1, 394
 and dictionaries 394
 examples of 386–90, 391, 392–3, 673 n. 2,
 792 n. 11
 in general discourse 393
 grammatical aspects of 390
 inflation in 386, 388–9, 633–4
 logical aspects of 390, 394
 as plural 392
see also information technology: misuse
 of term

term used as slogan 372, 385–6, 389, 394,
 395–6, 397, 633–4

testability, as quality of scientific theories
 216–18

tests
 through falsifications rather than
 confirmations 216–18, 219, 222–3
 genuine, definition of 219

theories
 based on false assumption 205
 based on particles 74–5
 corpuscular 74–5
 degraded by adding features 225–7, 228–9,
 590, 661, 672, 685, 783–4
 doubting vs. defending 208–9, 226–7, 269,
 328 n. 1, 511, 514
 expanding in order to avoid refutation, *see*
 falsifications: suppressed by
 expanding the theory
 falsifiable, *see* falsifiability
 identifying as mechanistic 84–5
 mechanistic, *see* mechanistic theories

theories (*cont.*)

- as models 91–2, 106–7, 209
- old, and the falsification principle 222
- pseudoscientific, unfalsifiable nature of, *see* pseudosciences: unfalsifiable nature of
- scientific, falsifiable nature of 210, 216–18
- in software, *see* software theories
- as systems based on conventions 91–2
- as systems that permit or forbid events 217–18, 217 n. 17, 226
- unfalsifiable, *see* falsifiability
- theorists, *see* software theorists
- three-body system, as complex structure 109–10, 119
- top-down design, *see* structured programming: top-down design in
- totalitarian democracy
 - definition of 849, 873
 - similar to academic ideologies 849–50
 - similar to business ideologies 849–50
 - and totalitarian software concepts 873–5
 - see also* totalitarianism
- totalitarianism
 - based on democratic ideas 848–9
 - based on the idea of natural order 851
 - and central economic planning 860–2
 - characteristics of 16–17, 130–2, 133–4, 854, 865, 869
 - as conformism to a mechanistic ideology 853–5, 857–8, 862, 868–9, 873
 - corrupting societies 846, 865–6, 869, 871
 - degrading language 124–5, 405–7, 863, 869–70
 - due to mechanism
 - based on mechanistic theories 16, 20, 202, 852
 - based on the idea of a perfect society 16–17, 133–4, 836, 837–8
 - promoted by corporations 843–4, 845
 - promoted by universities 276, 279, 843–4, 845
 - due to pseudosciences 279, 846, 859–60
 - due to software mechanism
 - based on mechanistic theories 133, 276, 866–7
 - based on the idea of a perfect society 133, 838
 - as conformism to a mechanistic ideology 866–9
 - by controlling minds through software 411–13, 490–1, 871
 - promoted by the elites 286, 838, 843, 847, 868–9, 871
 - see also* software totalitarianism

totalitarianism (*cont.*)

- due to utopianism 20, 130–2, 133–4, 836, 845, 850–1, 859, 872–3
- embraced by intellectuals 859
- as enactment of a myth 30
- extending the scope of politics 874
- fragmentary, through many elites 847
- as idea, circularity of 851–2, 853
- inefficiency of 30
- justification of 19–20, 133–4, 851, 852, 853–4, 861–2
- as means to create a perfect society 16–17, 18–19, 133–4, 836
- as mechanistic delusion 21, 24, 276, 852, 859–60
- non-political, similar to political 413, 846–7, 848, 865–6, 869, 871, 872–5
- as non-political ideology 846–7, 865–6, 869, 871–5
- objections to 19–20
- Orwell's model of
 - and the academic elites 859–60
 - and central economic planning 860–2
 - and mechanistic language 398, 405–7, 862–4
 - and software totalitarianism 412–13, 848, 869–71, 872–5
- perceiving human beings as deterministic systems 836, 852, 861–2
- perceiving societies as deterministic systems 852
- progression toward 30, 40, 399, 490–1, 843–4, 846, 858–9, 863, 865–6
- promoted by corporations 843–4, 846, 853, 857–8
- promoted by universities 276, 279, 837–8, 843–4, 846, 853, 854–5, 859
- as pseudoscience 846, 859
- replacing individuality with conformism 853, 857–8, 862
- role of the bureaucracy under 30, 490–1, 844, 847
- role of the elite under 30, 490–1
- through software, *see* software totalitarianism; totalitarianism: due to software mechanism
- and the software myth 31
- Talmon's model of
 - and the academic elites 854–5
 - and the corporate elites 856–8
 - and the origins of totalitarianism 848–51
 - and software totalitarianism 848, 866–9, 872–5
- transformational grammar
 - definition of 251

transformational grammar (*cont.*)
 as extension of simpler grammars 262–3
 transformations in 260–1, 262–3
see also Chomskyan linguistics
 transformations, in structuralism 245–6, 248,
 249–50
 transformations, in structured programming,
see structured programming:
 transformations in
 transformations, in transformational grammar
 260–1, 262–3
 transition, between myths 15, 24–6
 trap, mechanistic 66, 130, 190, 419, 424–5, 456,
 833, 875
 truth functions
 definition of 332, 710
 in Wittgenstein's early philosophy 332
 truth tables, description of 332, 710
 truth values, definition of 332, 707, 709
 tuples
 in predicate calculus 711, 732, 746–7
 in the relational model, *see* relational model:
 records (rows, tuples) in
 Turing machines
 and Böhm and Jacopini's work 571–2, 573
 description of 569–70
 as mechanistic models 569–70
 as theoretical devices 569–70, 571–2
 two-body system, as simple structure 109–10,
 119

U

understanding, mechanistic, *see* mechanistic
 explanation
 unity of science, and logical positivism 325
 universal characteristic, as mathematical
 language 315
 universal grammar
 definition of 254
see also Chomskyan linguistics
 universality, of scientific theories 91–2
 universities
 as bureaucracy 14, 16, 275, 855
 corruption in, *see* corruption: in universities
 as elite 844, 846, 849–50, 854–5, 859–60
 fostering programming incompetence 39,
 65, 275, 676, 794
 promoting mechanistic concepts, *see*
 mechanism: promoted by universities
 promoting totalitarian concepts 276, 279,
 837–8, 843–4, 846, 853, 854–5, 859
 unspecifiable knowledge, *see* non-
 mechanistic knowledge; tacit
 knowledge

utopianism
 causing totalitarianism 20, 130–2, 133–4,
 836, 845, 850–1, 859, 872–3
 due to mechanism 133, 845, 850, 872
 due to scientism 20, 130, 850
 due to software mechanism 133, 874

V

variables, as software principle 353
 verifiability
 vs. falsifiability 214, 216–17, 324
 and logical positivism 324
 and Wittgenstein's early philosophy 334
 visual perception
 as acquired skill 160
 as non-mechanistic phenomenon 150–1
 von Halle, Barbara, *see* Fleming, Candace C.

W

Westfall, Richard S.
 and mechanism 79
 and scientism 136–7
 wholes
 as justification of irresponsibility 841, 842
 as unexplained social forces 840–1
 as unexplained software forces 841–2
 Whyte, Lancelot L., and atomism 78, 81
 Wittgenstein, Ludwig
 and the early philosophy, *see* Wittgenstein's
 early philosophy
 and the later philosophy, *see* Wittgenstein's
 later philosophy
 and the philosophy of language 321, 328–9,
 336–7
 and the shift in philosophies 328–9, 336–7,
 343–4
 Wittgenstein's early philosophy
 as abstract system 330–1, 334
 description of 330–3
 limitations of 334
 as mapping system 328, 330–3
 and meaningfulness 333–4
 popularity of 328–9, 334, 336
 and software theories 329, 335, 344–5
 structures and elements in 330–3
 and truth functions 332
 and verifiability 334
 Wittgenstein's later philosophy
 and classification systems 338–41, 342–3,
 356–9, 360–3
 and complex structures 336–43, 345–7
 and the idea of family resemblance 337–8,
 342–3

Wittgenstein's later philosophy (*cont.*)
 and the idea of games 337–41, 342–3, 346–7
 and the idea of language games 342, 343
 misinterpreted by mechanists 329
 as non-mechanistic system 328, 336, 343–4
 as repudiation of mechanism 328, 344
 and software applications 345–7, 360–3
 and word meanings 336–7, 343

words
 and deception through language 374–5,
 376–82
 meaning of 12, 112, 256, 259, 311–12, 316,
 350, 401, 641
 in Wittgenstein's later philosophy 336–7, 343

workers
 deskilling of 200, 295–6
 as software bureaucrats 37–9, 487–9
 turned into software consumers 38–9

world
 explained through empiricism 209–10, 216
 identified with mathematics 88–9, 301–2,
 309–12
 material vs. mental, in rationalism 306–8

world (*cont.*)
 perceived as logically perfect 301–2, 319–20,
 344–5
 perceived as simple 79, 80, 91–2
 represented with a perfect language 301–2,
 303–4, 319–21, 325–6, 330–3, 640–1
 represented with language 299–300, 368–9,
 408–9, 870
 represented with software 256, 302, 345–6,
 368–9, 408–9, 411–13, 542–3, 642–3,
 652, 847, 868, 870

X

X-ray pictures, learning to interpret 167

Y

Y2K (year 2000), software problems of 200
 Yourdon, Edward
 and Böhm and Jacopini's work 576
 and the GOTO debate 603–4, 606, 614
 and non-standard constructs 584

