

# SOFTWARE AND MIND

Andrei Sorin

EXTRACT

Index

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

**Copyright © 2013, 2019 Andrei Sorin**

**The free digital book and extracts are licensed under the  
Creative Commons Attribution-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 free at the book's website.

[www.softwareandmind.com](http://www.softwareandmind.com)



SOFTWARE  
AND  
MIND

The Mechanistic Myth  
and Its Consequences

Andrei Sorin

ANDSOR BOOKS

Copyright © 2013, 2019 Andrei Sorin  
Published by Andsor Books, Toronto, Canada (www.andsorbooks.com)  
First edition 2013. Revised 2019.

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.

The free digital book is a complete copy of the print book, and is licensed under the Creative Commons Attribution-NoDerivatives International License 4.0. You may download it and share it, but you may not distribute modified versions.

For disclaimers see pp. vii, 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

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 philosopher 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

	<b>Preface</b>	<b>xiii</b>
<b>Introduction</b>	<b>Belief and Software</b>	<b>1</b>
	Modern Myths	2
	The Mechanistic Myth	8
	The Software Myth	26
	Anthropology and Software	42
	Software Magic	42
	Software Power	57
<b>Chapter 1</b>	<b>Mechanism and Mechanistic Delusions</b>	<b>68</b>
	The Mechanistic Philosophy	68
	Reductionism and Atomism	73
	Simple Structures	90
	Complex Structures	96
	Abstraction and Reification	111
	Scientism	125
<b>Chapter 2</b>	<b>The Mind</b>	<b>140</b>
	Mind Mechanism	141
	Models of Mind	145

	Tacit Knowledge	155
	Creativity	170
	Replacing Minds with Software	188
<b>Chapter 3</b>	<b>Pseudoscience</b>	<b>200</b>
	The Problem of Pseudoscience	201
	Popper's Principles of Demarcation	206
	The New Pseudosciences	231
	The Mechanistic Roots	231
	Behaviourism	233
	Structuralism	240
	Universal Grammar	249
	Consequences	271
	Academic Corruption	271
	The Traditional Theories	275
	The Software Theories	284
<b>Chapter 4</b>	<b>Language and Software</b>	<b>296</b>
	The Common Fallacies	297
	The Search for the Perfect Language	304
	Wittgenstein and Software	326
	Software Structures	345
<b>Chapter 5</b>	<b>Language as Weapon</b>	<b>366</b>
	Mechanistic Communication	366
	The Practice of Deceit	369
	The Slogan "Technology"	383
	Orwell's Newspeak	396
<b>Chapter 6</b>	<b>Software as Weapon</b>	<b>406</b>
	A New Form of Domination	407
	The Risks of Software Dependence	407
	The Prevention of Expertise	411
	The Lure of Software Expedients	419
	Software Charlatanism	434
	The Delusion of High Levels	434
	The Delusion of Methodologies	456
	The Spread of Software Mechanism	469
<b>Chapter 7</b>	<b>Software Engineering</b>	<b>478</b>
	Introduction	478
	The Fallacy of Software Engineering	480
	Software Engineering as Pseudoscience	494

Structured Programming	501
The Theory	503
The Promise	515
The Contradictions	523
The First Delusion	536
The Second Delusion	538
The Third Delusion	548
The Fourth Delusion	566
The <i>GOTO</i> Delusion	586
The Legacy	611
Object-Oriented Programming	614
The Quest for Higher Levels	614
The Promise	616
The Theory	622
The Contradictions	626
The First Delusion	637
The Second Delusion	639
The Third Delusion	641
The Fourth Delusion	643
The Fifth Delusion	648
The Final Degradation	655
The Relational Database Model	662
The Promise	663
The Basic File Operations	672
The Lost Integration	687
The Theory	693
The Contradictions	707
The First Delusion	714
The Second Delusion	728
The Third Delusion	769
The Verdict	801
<b>Chapter 8 From Mechanism to Totalitarianism</b>	<b>804</b>
The End of Responsibility	804
Software Irresponsibility	804
Determinism versus Responsibility	809
Totalitarian Democracy	829
The Totalitarian Elites	829
Talmon's Model of Totalitarianism	834
Orwell's Model of Totalitarianism	844
Software Totalitarianism	852
<b>Index</b>	<b>863</b>



# Preface

This revised version (currently available only in digital format) incorporates many small changes made in the six years since the book was published. It is also an opportunity to expand on an issue that was mentioned only briefly in the original preface.

*Software and Mind* is, in effect, several books in one, and its size reflects this. Most chapters could form the basis of individual volumes. Their topics, however, are closely related and cannot be properly explained if separated. They support each other and contribute together to the book's main argument.

For example, the use of simple and complex structures to model mechanistic and non-mechanistic phenomena is explained in chapter 1; Popper's principles of demarcation between science and pseudoscience are explained in chapter 3; and these notions are used together throughout the book to show how the attempts to represent non-mechanistic phenomena mechanistically end up as worthless, pseudoscientific theories. Similarly, the non-mechanistic capabilities of the mind are explained in chapter 2; the non-mechanistic nature of software is explained in chapter 4; and these notions are used in chapter 7 to show that software engineering is a futile attempt to replace human programming expertise with mechanistic theories.

A second reason for the book's size is the detailed analysis of the various topics. This is necessary because most topics are new: they involve either

entirely new concepts, or the interpretation of concepts in ways that contradict the accepted views. Thorough and rigorous arguments are essential if the reader is to appreciate the significance of these concepts. Moreover, the book addresses a broad audience, people with different backgrounds and interests; so a safe assumption is that each reader needs detailed explanations in at least some areas.

There is some deliberate repetitiveness in the book, which adds only a little to its size but may be objectionable to some readers. For each important concept introduced somewhere in the book, there are summaries later, in various discussions where that concept is applied. This helps to make the individual chapters, and even the individual sections, reasonably independent: while the book is intended to be read from the beginning, a reader can select almost any portion and still follow the discussion. In addition, the summaries are tailored for each occasion, and this further explains that concept, by presenting it from different perspectives.



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.

The book's epigraph, about Newspeak, will become clear when we discuss the similarity of language and software (see, for example, pp. 409–411).

Throughout the book, the software-related arguments are also supported with ideas from other disciplines – from the philosophies of science, of mind, and of language, 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.

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, and their long history. 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.

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 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). The plural, “elites,” is used when referring to several entities within such a body.

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](http://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 have published, in source form, some of the software 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, 372–3, 374–6, 382
- abstraction
  - benefits of 94
  - in classification systems 93, 112, 622–3, 637–8
  - definition of 93, 622–3
  - examples of 93–4
  - as general programming concept 655–6
  - in hierarchical structures 93–4, 622–3, 637–8
  - levels of, *see* levels
  - in object-oriented programming 622–3, 643, 655–6
  - in the relational model 668, 669, 689, 692, 704, 709, 774, 786
  - in structured programming 511, 517, 578, 599
- abstraction, fallacy of
  - causing loss of alternatives 95, 111–14, 288
  - consequences of 95–6, 112–13
  - definition of 94–5, 111
  - and the elites 112, 113–14
  - examples of 95, 111–12, 113–14
  - in human phenomena 283, 817, 847
  - in knowledge structures 153, 416
- abstraction, fallacy of (*cont.*)
  - and mind models 153
  - and reification 120–1
  - see also* alternatives, loss of; mechanistic fallacies
- academia, *see* universities
- academic mechanism, *see* mechanism:
  - promoted by universities
- ACM, *see* computer associations
- acronyms
  - definition of 371 n. 1
  - used as new words 371–2, 401–2
  - used to impoverish knowledge 371–2, 393–4, 401–2, 850
- advertising
  - deception in 22, 218–19, 831, 843, 853–4
  - and the falsification principle 218–19
  - as mechanistic indoctrination 831, 843
  - using confirmations to deceive in 218–19
- agnosia, and novice performance 158
- agriculture, expertise in, vs. magic 47–8
- alphabet of thought, as basis of all knowledge 312–13, 314
- alternatives
  - in complex structures 117, 281–4, 815–17
  - in simple structures 91, 111–13, 281–4

- alternatives (*cont.*)  
 in software phenomena 286, 817–18, 853
- alternatives, large number of  
 and mechanistic models 281–4, 814–17  
 misinterpreted as all alternatives 113, 288, 294, 816–17, 818–19  
 misinterpreted as creativity and free will 278, 279, 281–4, 286, 290–2, 814, 817–18  
 misinterpreted as indeterminism 814–17
- alternatives, loss of  
 due to abstraction 95, 111–14, 288  
 due to both mechanistic fallacies  
 by degrading the concept of indeterminism 817, 818–19  
 in the new pseudosciences 283  
 by simplifying complex structures 121–2  
 through software devices 416, 428–9, 439  
 through software mechanism 124, 287–8, 363, 365, 368, 436–8, 614–16  
 and software totalitarianism 853, 856  
 due to mechanistic approximation 815–17  
 due to mechanistic language  
 and deception 369–70, 382, 383  
 and Newspeak 122–3, 398  
 and similarity to software 290, 364–5, 367–8, 614–15  
 and software totalitarianism 856  
 and the term “information technology” 396  
 and the term “technology” 392  
 and totalitarianism 849–50
- due to reification 118–19
- in software applications  
 and the delusion of high levels 288–90, 294, 436–8, 614–16, 818–19  
 due to abstraction 113–14  
 due to emulating manufacturing concepts 439  
 due to programming incompetence 124, 289  
 and similarity to language 368, 614–15, 692  
 and software totalitarianism 853, 856
- analogies  
 and formal theories 76  
 as metaphors 79
- analysis, in simple structures 92–3
- analytic philosophy, and language mechanism 317–19
- Andreski, Stanislav, and structuralism 248
- animals, used to explain human behaviour 235, 237, 239
- applications, *see* software applications
- approximation  
 mechanistic, *see* mechanistic approximation
- approximation (*cont.*)  
 non-mechanistic, complex structures as 146, 168–9
- art, and creativity 173–4
- artificial intelligence  
 based on mechanism 143–4  
 failure of 143–4, 181–4  
 and mechanistic mind theories 143–4  
 naivety of 143, 259–60, 429  
 philosophical arguments against 142 n. 4
- artificial languages  
 based on simple structures 110, 307, 311–15, 626–7  
 as classification systems 312, 313–14, 626–7  
 as means to attain new knowledge 312–13  
 as means to express knowledge 312  
*see also* perfect language
- artisans, *see* craftsmen
- aspects  
 of software applications 344, 346, 417, 434, 435–6, 438, 458, 507, 528, 614, 666–8  
 of stories 349, 355, 399
- assumption, false, *see* false assumption
- astrology  
 and the fallacy of software engineering 496  
 as pseudoscience 13, 229  
 resembling rational pursuits 496
- atomism  
 ancient 73–4  
 as basis of understanding 80–1  
 causing the mechanistic myth 82  
 in corpuscular theories 74–5, 79  
 definition of 9, 68  
 in hierarchical structures 83–4
- atomistic philosophy, ancient 73–4
- atomists, ancient 73–4
- atoms, in the atomistic philosophy 73–4
- attributes  
 in classification systems  
 and complex structures 101–2, 628–30, 638, 645  
 and the idea of a perfect language 314, 627  
 and mechanistic approximation 11, 104, 357  
 and simple structures 93, 623, 632  
 and Wittgenstein’s later philosophy 336–9, 340–1, 354–7, 358–61
- combining in one structure 99–100, 336–9, 354–7, 358–61, 628–30, 645
- in complex structures  
 causing interactions 97, 103, 170–1, 627, 628–30  
 and the idea of a perfect language 314, 627

attributes (*cont.*)  
 in complex structures (*cont.*)  
 and Newspeak 399  
 and Wittgenstein's later philosophy  
 335-9, 340-1, 344-5, 354-7, 358-61  
 constant vs. changeable 171  
 depicted as one within another 99-100,  
 336-9, 354-7, 358-61, 628-30  
 determining contexts 171  
 examples of 11-12, 93-4, 96-7, 98, 335-9  
 flow-control, *see* flow-control attributes  
 in language structures 110, 399  
 multiple 98, 99, 335-9, 340-1, 354-7, 358-61,  
 528-9, 549-50, 628-30, 638, 645  
 multivalued vs. two-valued 336 n. 11  
 and non-mechanistic capabilities 102, 314,  
 335-9, 354-7  
 in object-oriented programming, *see* object-  
 oriented programming: attributes in  
 and reification 105, 115-16, 152-3  
 repeated 99-100, 337-9, 354-7, 358-61, 628,  
 645  
 shared by several elements 99-100, 336-9,  
 340-1, 354-7, 358-61, 628  
 in simple structures 11-12, 93-4, 96-7, 98,  
 104, 360  
 of software entities 12, 28, 345, 358-61, 528,  
 580-1  
 and software processes 346, 358-61, 528,  
 549-50  
 in software structures, causing interactions  
 358-61, 528, 549-50, 579-82, 644, 645  
 attributes, in the relational model, *see* relational  
 model: fields (columns, attributes) in  
 autism, as failure to combine knowledge  
 structures 178-9, 183  
 automata, and formal languages 259-60  
 automotive technology, analysis of term 395

## B

Babel, and the idea of a perfect language 301-2  
 Bacon, Francis, and utopianism 129-30  
 basic file operations, *see* file operations  
 begging the question, *see* circularity  
 behaviour  
 illogical, creativity as 181-2  
 logical, lack of creativity as 181-2  
 normal, as unspecifiable quality 181-2,  
 186-7  
 science of, behaviourism perceived as 233,  
 234  
 behaviourism  
 based on animal behaviour 235, 237, 239  
 based on mechanism 234, 236

behaviourism (*cont.*)  
 definition of 233-4  
 failure of 236, 238-40  
 as means to create a perfect society 277  
 perceived as science of behaviour 233, 234  
 as pseudoscience 233, 236-7, 238-9, 277  
 and reification 236  
 reinstating the traditional concepts of  
 mind 238-9  
 and simple structures 275  
 stimulus-response process in 234, 236-7,  
 238-40  
 and the traditional concepts of mind 233-4,  
 238-9  
 Berlin, Isaiah  
 and free will vs. determinism 826-7  
 and modern myths 5  
 and scientism 132-3  
 binary operations, in structuralism 242-5, 246  
 biology, and scientism 134-5  
 black-box principle, *see* structured  
 programming: black-box principle in  
 blocks, *see* software blocks  
 Boden, Margaret A., and structuralism 247  
 Boehm, Barry W., and software engineering  
 485-6, 494  
 Böhm, Corrado, *see* Böhm and Jacopini,  
 work of  
 Böhm and Jacopini, work of  
 misrepresented as practical concept 552-3,  
 554-5, 557-8, 591  
 misrepresented as saying three elementary  
 constructs 558-63, 571  
 misrepresented as the black-box principle  
 570  
 perceived as proof of the structured  
 programming theory 509, 552, 557-8,  
 563, 564  
 and the structure theorem 509  
 as theoretical concept 554, 557-8, 563-4  
 and Turing machines 557-8, 559  
 Boole, George, and language structures 310-11  
 Boolean logic  
 invoked in structuralism 244  
 as logic system 697  
 and the rationalist language fallacies 310-11  
 Brackett, Michael H., and denormalization  
 782-3  
 Brathwaite, Ken S.  
 and normalization 758-60  
 and the relational model 709  
 Bruner, Jerome S., and creativity 172, 173-4  
 bureaucracy  
 role of, under totalitarianism 30, 476-7,  
 830, 833

- bureaucracy (*cont.*)  
 in software, *see* software bureaucracy;  
 software bureaucrats  
 universities as 14, 16, 273, 841  
 business, *see* corporations  
 business mechanism 830, 831, 842–3
- C**
- C (programming language)  
 GOTO statement in 596, 603  
 as high-level language 452  
 as hybrid language, in object-oriented  
 programming 642  
 statements in, vs. subroutines 452  
 canoe building, expertise in, vs. magic 48–9  
 capabilities  
 mechanistic, *see* mechanistic capabilities  
 non-mechanistic, *see* non-mechanistic  
 capabilities  
 Carnap, Rudolf, and logical positivism 323  
 Carter, John, and normalization 762–3  
 CASE (Computer-Aided Software Engineering)  
 and application definitions 465–7  
 failure of 465, 467–9, 586, 806–7  
 and methodologies 467–9  
 promises of 465, 521–2  
 promotion of 522, 806–7  
 and software mechanism 467–8, 521–2, 586  
 CASE (software construct), *see* software  
 constructs, flow-control: CASE  
 Cassirer, Ernst  
 and artificial languages 306, 311  
 and mana 57, 61, 62  
 causality, fallacy of 209–10  
 centralized economy  
 as growing trend 846  
 totalitarian nature of 131, 846–8  
 century of genius, corpuscular fantasies in  
 74–5, 76, 78–80  
 Channell, David F., and scientism 133, 134  
 charlatanism  
 due to mechanism 20–2, 90, 122, 699, 843  
 in promotional work 21–2, 832, 843  
 in research, *see* corruption: in universities  
 in software, *see* software charlatanism  
 chick sexing  
 through intuitive knowledge 165–7  
 as non-mechanistic phenomenon 167  
 similar to face recognition 167  
 chicken's quandary  
 and the problem of induction 210–11  
 similar to human problems 210–11  
 Chomsky, Noam  
 and artificial intelligence 260  
 Chomsky, Noam (*cont.*)  
 and the concept of freedom and creativity  
 279–81  
 and formal grammars 260–1  
 and the humanist cause 280  
 and informal models 76  
 and linguistics, *see* Chomskyan linguistics  
 and Skinner's views on language 240  
 Chomskyan linguistics  
 and arguments against innate faculties  
 168, 268–71  
 based on mechanism 259–61  
 circularity of 253, 268  
 facts adduced to support theory 251–2  
 failure of 249–50, 257, 262–3, 266, 271  
 fallacies of 253–6, 260–1, 269–71  
 fifth model (minimalist program) 264 n. 23  
 first model 258–9  
 as formal system 249, 255–6, 258–9, 260–1,  
 262–3  
 fourth model (principles and parameters)  
 263–6  
 and grammar 249, 255–6, 258–9, 260–1  
 harmful effects of 250, 261, 279–81  
 as informal system 76, 252  
 and innate faculties 76, 250–3, 267–8,  
 278–9, 301  
 and mind models 259–60, 266, 268, 279–80  
 and modularity 266  
 and programming languages 261  
 as project in artificial intelligence 260  
 as pseudoscience 249–50, 257–8, 261–4,  
 271, 278  
 and rationalism vs. empiricism 278–80  
 and reification 255, 256–7, 266  
 reinstating the traditional concepts 265  
 as revolution 249, 263  
 second model (standard theory) 262  
 and semantics 255–6, 261, 262–3  
 and simple structures 275  
 third model (extended standard theory) 263  
 transformations in 258–9, 260–1  
 and universal grammar 252–3, 271  
 circular definition, software engineering  
 definition as 481–2, 494–5  
 circularity  
 in Chomskyan linguistics 253, 268  
 in the idea of a general will 837, 838  
 in the idea of social forces 826  
 in the idea of software engineering 481–3,  
 485, 491, 494–5  
 in the idea of software forces 828–9  
 in the idea of totalitarianism 837–8, 839  
 in ignoring falsifications 498  
 in language mechanism 318

- circularity (*cont.*)  
 in mechanistic theories 132, 150, 152, 205,  
 253, 268, 318, 837–8  
 in mind models 152  
 in scholastic explanations 70, 149
- classes, in object-oriented programming, *see*  
 object-oriented programming:  
 classes in
- classification systems  
 and abstraction 93, 112, 622–3, 637–8  
 artificial languages as 312, 313–14, 627  
 attributes in, *see* attributes: in classification  
 systems  
 and hierarchical structures 8, 91–2, 96–7,  
 98, 101–2, 629–30  
 as mechanistic approximation 11, 101–2,  
 104, 357, 632  
 and object-oriented programming 622–3,  
 624–5, 628–9, 631–2, 638–40, 644–5,  
 646  
 and reification 313–14, 627, 631–2  
 and software structures 347–8, 350, 352,  
 358–61, 528  
 and Wittgenstein's later philosophy 336–9,  
 340–1, 354–7, 358–61
- COBOL (programming language)  
 file operations in 672, 676–9, 680, 683,  
 684–5, 691  
 GOTO statement in 603, 609 n. 47  
 as high-level language 452  
 as hybrid language, in object-oriented  
 programming 642  
 statements in, vs. subroutines 452
- Codd, E. F.  
 as inventor of the relational model 699  
 and the mathematics of normalization 752
- Cohen, I. Bernard, and scientism 132, 133, 135
- collective thinking, myths as 3–4
- columns, in the relational model, *see*  
 relational model: fields (columns,  
 attributes) in
- combinations, of attributes, in one structure  
 99–100, 336–9, 354–7, 358–61, 628–30,  
 645
- combinations, of file relationships 740
- combinations, of knowledge structures  
 and creativity 172–4, 175–6, 181–5  
 and mental disorders 176–9, 183
- comic strips, as myths 6
- Communism  
 as myth 5  
 as totalitarian democracy 835  
 and utopianism 130
- 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 117, 281–4, 815–17  
 attributes in, *see* attributes: in complex  
 structures  
 and contexts 170–1, 182–3, 185, 186–7  
 and creativity 173, 179, 180, 182–4  
 definition of 88, 98  
 examples of 107, 108, 110, 145–6  
 knowledge as 146, 152–3, 162, 163–5, 174,  
 268–71, 313–15, 321, 415–16, 627, 637–8  
 and the mind 162, 192, 245, 268–71, 414–16  
 as models, *see* non-mechanistic models  
 as non-mechanistic approximation 146,  
 168–9  
 and non-mechanistic capabilities 108–9,  
 110, 146, 162, 192–3, 415–16, 487, 490,  
 579–80  
 and non-mechanistic phenomena 11–12,  
 102–3, 106–7, 125, 202–3, 231–2, 236,  
 246–7, 275–6, 847  
 non-mechanistic qualities of 102–3, 146  
 simplification of, *see* abstraction, fallacy of;  
 mechanistic approximation;  
 mechanistic fallacies; reduction: of  
 complex to simple structures;  
 reification, fallacy of  
 and tacit knowledge 158–62  
 and Wittgenstein's later philosophy 334–41,  
 343–5  
*see also* interactions, of structures;  
 non-mechanistic phenomena
- complexity  
 levels of, *see* levels  
 in structures 91, 91 n. 1, 94, 97–8, 106–7,  
 119, 622, 695 n. 2, 715 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 208, 214  
 looking for  
 as illogical method 208–9, 213–14,  
 218–21, 247–8, 497, 660  
 and old theories 220  
 as popular method 216–17, 218–20, 660

- confirmations (*cont.*)  
 looking for (*cont.*)  
 and software testing 221  
 sought by pseudoscientists 236–7, 247–8,  
 497, 500, 613  
 used to deceive  
 in advertising 218–19  
 in the media 219–20, 613, 660, 808  
*see also* falsification principle; falsifications
- conformism  
 vs. freedom 839  
 to a mechanistic ideology 839–40, 841,  
 843–4, 848, 854–5, 859  
 to the mechanistic software ideology 197,  
 409–11, 431–2, 476–7, 828–9, 854–5,  
 860–1  
 perceived as education 837, 840, 859  
 as reflection of the general will 837, 839, 860  
 replacing individuality 839, 843–4, 848, 859
- construction, emulated in software theories,  
*see* manufacturing: emulated in  
 software theories
- constructs, software, *see* software constructs,  
 flow-control
- context-dependent knowledge  
 and contexts 171  
 and skill levels 156
- context-free knowledge  
 and contexts 171  
 and mechanistic capabilities 162, 186–7  
 and skill levels 156
- contexts  
 as complex structures 170–1, 182–3, 185,  
 186–7  
 definition of 170–1  
 failure to recognize 177–9, 182–3  
 in humour 175–6, 185  
 and language 120, 254–5, 257  
 recognizing 182–3, 185, 186–7
- Copi, Irving M., and artificial languages 317
- corporations  
 corruption in, *see* corruption: in  
 corporations  
 as elite 830, 832, 835–6, 842–4  
 inefficiency in 37–8, 198, 473–5  
 irresponsibility in 473–5, 805–6  
 promoting mechanistic concepts 830, 831,  
 842–3  
 promoting totalitarian concepts 829–30,  
 832, 839, 843–4  
 software, *see* software companies
- corpuscular theories  
 as fantasies 75, 78–80  
 and mechanism 78–80  
 as metaphors 78–80
- corpuscular theories (*cont.*)  
 similar to the ancient theories 74–5
- correspondence, one-to-one, *see* mapping
- corroborating evidence, failed tests as 217, 276
- corruption  
 in corporations  
 allowing dependence on the software  
 elites 37, 461–2, 473–5, 778, 802–3,  
 805–6, 807–8, 857–8  
 deskilling workers 198, 293–4  
 following pseudoscientific software  
 theories 29, 469, 471, 473–5, 522, 779  
 preventing programming expertise 29,  
 34–5, 41, 124, 195–7, 780, 805–6, 808–9,  
 857  
 suppressing software failures 779, 808–9
- in society  
 apathy toward 804, 809, 858  
 due to language abuse 380  
 due to mechanism 433, 804, 845–7  
 due to software totalitarianism 409–11,  
 476–7, 833–4, 857–8  
 due to totalitarianism 832, 851–2, 855, 857
- in universities  
 accepting blindly the mechanistic  
 ideology 21, 127–8, 272–3, 499, 823,  
 840–1, 845–6  
 assuming that minds are like machines  
 261, 268–71, 275–8, 280, 319–20, 403–4,  
 432, 821, 823, 824–5, 840  
 causing totalitarianism 16, 840–1, 845–6  
 degrading the concept of responsibility  
 821, 823–4  
 misrepresenting the principles of  
 demarcation 228–30  
 promoting pseudoscientific notions  
 202, 271–3, 277, 499, 728, 780, 857  
 pursuing pseudoscientific theories 77,  
 139, 200, 227–8, 230, 233, 240–1,  
 249–50, 470–1, 780  
 struggling to save fallacious theories  
 232–3, 262–4, 267, 509–10, 535, 637, 658  
 studying non-mechanistic phenomena  
 mechanistically 189, 202–3, 240,  
 242–4, 254–6, 271–3, 281–4, 698–9  
 using stratagems to avoid falsifications  
 238–40, 246–8, 261–4, 499, 840
- cost  
 of programming incompetence 32–3, 198,  
 414, 472, 476, 506, 806 n. 2  
 of software mechanism 33, 124, 446–7
- craftsmanship  
 definition of 483  
 misrepresented by the software theorists  
 483–5, 491, 621–2, 661

craftsmanship (*cont.*)  
 as non-mechanistic knowledge 483, 490, 491  
 programming expertise as 491–2, 501  
 craftsmen  
 programmers compared to 480, 483–5,  
 491, 621–2, 661  
*see also* craftsmanship  
 creativity  
 in art 173–4  
 as capacity for low levels 428–9  
 as combining knowledge structures 172–4,  
 175–6, 181–5  
 and complex structures 173, 179, 180, 182–4  
 degrading the concept of 275–6, 279–81,  
 813–14, 817–18, 819–20, 821  
 denied by mechanistic theories 276–8, 432,  
 813–14  
 denied by software theories 290–2, 294–5,  
 828  
 in humour 175–6  
 as illogical behaviour 181–2  
 and mental disorders 176–9, 183  
 in metaphoric thinking 174  
 misinterpreted as large number of  
 alternatives 278, 279, 281–4, 286,  
 290–2, 814, 817–18  
 and non-mechanistic capabilities 170,  
 172–4, 175–6, 275–6, 428–9  
 in normal behaviour 172, 176, 179–80  
 in recognizing objects 181–2  
 replacing with software devices 280, 291–2,  
 294–5, 817–18  
 crisis, in software, *see* software crisis  
 cultural fashions, as myths 6–7  
 culture of knowledge substitutes 188, 860–1  
 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 798, 799

## D

Dante Alighieri, and the idea of a perfect  
 language 301

data fields, *see* fields: in data files  
 data files  
 definition of 674  
 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 672–3  
 difficulty of 687–8, 689  
 vs. file operations 673  
 as high-level operations 664–5, 669, 673,  
 688–9, 786–7  
 database structures  
 definition of 673  
 description of 688  
 interacting with other software structures  
 226, 688, 710, 719, 720–1, 723–4, 743,  
 747, 754, 786  
*see also* file relationships  
 database systems  
 definition of 673  
 and the delusion of high levels 669, 688–9,  
 786–7  
 early models of 664–5  
 vs. file systems 673  
 as frauds 664–5  
 as management tools 664–5  
 motivation for 663, 672  
 and programming incompetence 664–5,  
 691, 745–6, 778, 803  
 as programming substitutes 664–6, 745–6,  
 803  
 promises of 665  
*see also* relational model  
 databases  
 and applications, integration of 691, 786,  
 797  
 aspects of 766–8  
 definition of 673  
 design of  
 based on field dependencies 747–9,  
 754–5, 758, 759–60, 761–3, 764–5,  
 768–9, 788  
 based on file relationships 741–3, 748–9,  
 758–9, 760–1, 762, 763–4, 767–8, 784  
 as informal procedure 746, 748–9, 750–1,  
 766, 767–9, 782–4  
 as non-mechanistic phenomena 766–8  
 relationships in, *see* file relationships  
 as software principle 349–51  
 as software processes 350, 359, 688  
 as structures 350, 359, 688  
 Date, C. J.  
 and denormalization 781–2  
 and the first normal form 775–6

- Date, C. J. (*cont.*)  
 and the mathematics of normalization 753  
 and normalization 748, 760–2, 763–5  
 and ordinary field dependencies 757  
 and the relational model 709
- deception  
 in advertising 22, 218–19, 831, 843, 853–4  
 through language, *see* language: deception  
 through  
 in promoting ideologies 832, 851  
 in selling 22, 831  
 through software, *see* software charlatanism
- deduction  
 and hierarchical structures 85–6, 308, 694  
 and the problem of demarcation 211–12  
 in refuting theories 213
- definitions, of software applications  
 and CASE 465–8  
 as mechanistic approximation 459–60, 463  
 as mechanistic delusion 458–60, 463, 465–8  
 requiring non-mechanistic knowledge  
 466–7
- definitions, of structures 83, 90–1, 92–3, 540–1,  
 622–3
- dehumanization, through totalitarianism  
 19–20, 841
- delusions, mechanistic, *see* mechanistic  
 delusions
- demarcation  
 Popper's criterion of 212, 214–16  
 principles of  
 as concept, explanation of 208–9, 212,  
 213–16, 221–3  
 generally ignored 216, 218–20, 227–8  
 misinterpreted in universities 228–30  
 misrepresented in universities 228–30  
 and philosophical subtleties 228–9  
 and testing criteria 217, 230  
 used to expose pseudosciences 206,  
 228, 498  
 problem of 206, 207, 208–9  
 and the problem of induction 209–10,  
 211–12
- democracy  
 liberal, *see* liberal democracy  
 and progression toward totalitarianism  
 844–5, 858  
 totalitarian, *see* totalitarian democracy
- denormalization, *see* normalization, in the  
 relational model: denormalization in
- dependence  
 fostered by the software elites  
 causing irresponsibility 472–6, 819  
 and the delusion of high levels 438–42,  
 451, 454–6, 614–16, 636, 661–2, 691
- dependence (*cont.*)  
 fostered by the software elites (*cont.*)  
 due to programming incompetence  
 414, 461–2, 612, 661–2, 778, 802–3,  
 805–6  
 preventing non-mechanistic knowledge  
 289–90, 421–3, 425–6, 430, 479, 856–7,  
 860–1  
 and the promise to solve non-mechanistic  
 problems 193–5, 418, 419
- on knowledge substitutes  
 causing irresponsibility 473–6  
 vs. non-mechanistic capabilities 188  
 preventing non-mechanistic knowledge  
 198–9, 417–18, 425–6, 860–1  
 and the promise to solve non-mechanistic  
 problems 65–7, 193–5, 293–4
- on the mechanistic myth 25, 26
- on programming substitutes  
 causing irresponsibility 472–3  
 and the delusion of high levels 445–7,  
 454–6, 612, 614–16, 636, 661–2, 691
- due to programming incompetence  
 32–3, 38–9, 54–5, 196–7, 292–3, 409–11,  
 413, 805–6
- on ready-made products 842, 844
- on software 296–7, 407–8, 409–11, 431–2
- on software devices  
 causing irresponsibility 473–6, 819,  
 820–1, 825  
 and the delusion of high levels 438–42,  
 451  
 vs. non-mechanistic capabilities 418–19,  
 431–2  
 preventing non-mechanistic knowledge  
 198, 421–3, 430, 433, 860–1  
 and the promise to solve non-mechanistic  
 problems 193–5, 420–1
- dependencies, field, *see* relational model: field  
 dependencies in
- dependency theory, *see* normalization, in the  
 relational model
- Descartes, René  
 and artificial languages 311  
 and hierarchical structures 86, 87, 141–2,  
 206, 304–5  
 and language structures 308  
 and material world vs. mental world 304–5
- deskilling  
 of managers and workers 198, 293–4  
 of programmers 34–5, 35 n. 2, 472–3, 479,  
 483–6, 484 n. 13, 604–7, 611–12, 661–2
- determinism  
 and academic fashion 813–14  
 definition of 810

determinism (*cont.*)  
 vs. free will, and responsibility 809–10,  
 813–14, 819–20, 822–3, 826–9  
 and hierarchical structures 97, 275–6  
 Laplacean 72, 555, 810–11  
 and mathematics 87  
 and mechanism, *see* mechanism: and  
 determinism  
 and mechanistic approximation 812, 814–17  
 naivety of 72  
 in the new pseudosciences 275–6, 279–80,  
 281–4, 814  
 and non-mechanistic phenomena 89, 232,  
 281–4, 812–13  
 and scientific theories 89–90  
 and sociology 826–7  
 and software theories 286, 290–2, 817–18,  
 827–9  
*see also* indeterminism  
 deterministic models, *see* mechanistic models  
 deterministic phenomena, *see* mechanistic  
 phenomena  
 deterministic systems  
 human beings perceived as  
 in the new pseudosciences 179–81, 275–  
 6, 431–2, 813–14, 819–20, 840–1  
 in software-related affairs 820–1  
 under totalitarianism 822, 838, 847–8  
 societies perceived as  
 in the new pseudosciences 819–20,  
 840–1  
 under totalitarianism 838  
 development environments  
 as low levels in high-level systems  
 dishonest nature of 438–42, 451, 452–3,  
 454–6  
 and fourth-generation languages (4GL)  
 452–3, 690  
 and object-oriented programming  
 614–16, 634–5, 636, 661–2  
 and the relational model 662–3, 671,  
 690, 786–7  
 and structured programming 612  
 reversing the principles of programming  
 451, 455–6, 615–16, 634–5, 671  
*see also* software devices  
 devices, software, *see* software devices  
 die rolling  
 as analogy of human phenomena 815–16  
 as non-mechanistic phenomenon 815–16  
 Dijksterhuis, E. J.  
 and corpuscular theories 75, 79  
 and mechanism 71, 80, 86, 87  
 Dijkstra, E. W.  
 and the GOTO debate 508–9, 594, 606

Dijkstra, E. W. (*cont.*)  
 and mathematical programming 487–8  
 and structures, static vs. dynamic 532  
 doctrine, mechanistic, *see* mechanism  
 dogma  
 mechanism as 7, 26, 89, 107, 231–3, 272–3, 489  
 software mechanism as 26, 489, 490  
 domination, through software, *see* software  
 domination  
 doublespeak  
 definition of 380  
 examples of 380–2  
 Dreyfus, Hubert L., and the philosophy of  
 artificial intelligence 142 n. 4  
 Dreyfus, Hubert L. and Stuart E.  
 and artificial intelligence 144  
 and mind substitutes 188  
 and skill acquisition 155–8  
 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 300–2, 307  
 economics  
 and reification 847  
 and scientism 131, 133–4, 847  
 economy  
 centrally-planned, *see* centralized economy  
 as non-mechanistic phenomenon 847  
 education  
 conformism perceived as 837, 840, 859  
 as mechanistic indoctrination 16, 831, 841  
 of programmers 32, 195, 412–13, 483, 491–2,  
 605, 607  
 Einstein's theory of relativity, and mechanistic  
 approximations 104  
 elements, of structure  
 abstract concepts as 112  
 atomic and independent 78, 153–4, 283–4,  
 285–6, 294, 309–10, 313–14, 321  
 classes as, in object-oriented programming  
 623–4  
 concrete things as 112  
 definition of 90–1, 97 n. 1  
 objects as, in object-oriented programming  
 623–4  
 shared by several structures 97, 103, 436–8,  
 614  
 starting 78, 91, 93, 540  
 terminal 90–1

- elements, of structure (*cont.*)  
 top 90–1, 93, 540
- 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 364  
 corporations as 830, 832, 835–6, 842–4  
 good 15, 23–4  
 justified by a mechanistic ideology 830, 851  
 manipulating language 122, 368, 396, 397–8, 404–5, 406–7, 849–50, 851, 856–7  
 and myths 15–16, 23–4  
 promoting language mechanism 249, 824–5, 856  
 role of, under totalitarianism 30, 476–7  
 software, *see* software elites  
 in totalitarian democracy 835, 837–8  
 universities as 830, 832, 835–6, 840–1, 845–6  
 weakness of 833  
*see also* corporations; universities
- empiricism  
 explaining the world through 207–8, 214  
 and mind models 278–80  
 in Popper's criterion of demarcation 214  
 and the problem of induction 211
- engineering  
 emulated in software theories 84, 480–2, 489–90, 492–3, 494–5, 619  
 and hierarchical structures 84–5  
 misinterpreted by the software theorists 84, 492–3  
 non-mechanistic aspects of 492–3
- entities, of structure  
 definition of 97 n. 1  
*see also* elements, of structure
- environment, natural, destruction of 19–20
- environment, software, *see* development environments
- Epicureanism, as atomistic philosophy 74
- ethics, and the idea of responsibility 809–10
- Euclid's geometry, based on hierarchical structures 85, 87
- evolution, and non-mechanistic capabilities 140, 152, 269
- experience, personal, *see* personal experience
- expert systems, as mechanistic delusion 429
- expertise  
 characteristics of 155–8, 270  
 degrading the concept of 272–3, 414, 430, 821, 831, 854  
 due to intuitive knowledge 155–8, 166–7
- expertise (*cont.*)  
 in natural skills 157–8, 269  
 prevention of 65–7, 194–5, 409–11, 417–18, 473–6, 479  
 in programming, *see* programming expertise  
 substitutes for, *see* knowledge substitutes; programming substitutes
- experts  
 as charlatans 22  
 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 108–9, 117–18, 159
- failure, due to mechanism  
 in artificial intelligence 143–4, 181–4  
 of knowledge substitutes 27, 141, 158, 198–9, 294  
 of mechanistic theories, *see* mechanistic theories: failure of  
 of programming substitutes 27, 144–5, 158, 196, 292–3, 807–8  
 of software applications, *see* software failures  
 of software theories, *see* software theories: failure of
- fallacies, mechanistic, *see* mechanistic fallacies
- false assumption  
 causing invalid theories 203  
 in magic 49–50, 52  
 in myths 14  
 in pseudosciences 203, 498  
 in software theories 32, 49–50, 52
- falsifiability  
 as basis of testability 214–16  
 lacking in pseudoscientific theories 215–16, 222–4, 232–3  
 as misunderstood concept 216, 229  
 as quality of scientific theories 214–16, 267  
 vs. verifiability, as criterion of demarcation 212, 214–15, 322
- falsification principle  
 vs. confirmations 213–15, 236–7, 247–8, 500, 613, 660  
 vs. defending the theory 213–14, 232–3, 267, 326 n. 1, 497  
 definition of 212  
 generally ignored 216, 218–20  
 and old theories 220  
 used in decision-making 218–20

- falsification principle (*cont.*)  
 used in testing software 221  
 used to expose deception in advertising  
 218–19
- falsifications  
 avoiding, stratagems for 222–4, 236–7,  
 246–8, 261–4, 342, 658, 840  
 ignored by pseudoscientists 219, 237, 248,  
 497, 500, 613  
 ignoring, circularity of 498  
 looking for  
 as logical method 209, 213–16, 218–21,  
 497, 660  
 and old theories 220  
 and software testing 221  
 suppressed by expanding the theory  
 in behaviourism 238–40  
 in Chomskyan linguistics 257–8, 261–4,  
 267  
 in the new pseudosciences 232–3, 342,  
 499, 713–14, 813  
 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 441, 612, 787  
 in software theories 500, 612  
 as stratagem, explanation of 222–5,  
 229–30, 497  
 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 335–6, 340–1
- field dependencies, *see* relational model: field  
 dependencies in
- field relationships, and file relationships  
 740–2
- fields  
 in data files  
 alphanumeric, definition of 676  
 binary, definition of 676  
 date, definition of 676  
 definition of 674  
 numeric, definition of 676  
 relating files through 683–6, 737–8,  
 739–42, 743–4, 754–5  
 used as record key 674–5, 679–86, 687,  
 739–40
- fields (*cont.*)  
 in the relational model, *see* relational model:  
 fields (columns, attributes) in
- file maintenance  
 as low-level operations 435–8, 669  
 using the basic file operations 669, 680
- file operations  
 for accessing individual records 679–80  
 for adding records 679  
 and applications, integration of 691, 786, 797  
 basic, definition of 672, 677–8  
 as correct level of abstraction 689–90, 691  
 current index in  
 definition of 677  
 uses of 677–8  
 current key in  
 definition of 677  
 uses of 677–8  
 current pointer in  
 definition of 677  
 uses of 678  
 for data analysis 682–3  
 vs. database operations 673  
 definition of 672
- DELETE  
 definition of 678  
 vs. SQL operations 796–7, 798  
 uses of 679, 682
- for deleting records 679, 682
- for displaying fields 679
- for file maintenance 669, 680  
 and file scanning loops 680–6, 687, 689  
 flexibility of 690  
 and flow-control constructs 676, 680–3,  
 684–5, 690, 691, 724, 799
- as high-level operations 670
- and interacting structures 688
- as low-level operations 670, 673, 688–90, 786
- for modifying records 679, 682
- as necessary and sufficient operations 672,  
 679, 689, 786, 796–7
- and nesting of file scanning loops 683–6,  
 687, 774, 779
- pointer in  
 definition of 677, 798  
 uses of 678
- for queries and reports 682, 724
- READ  
 definition of 678  
 vs. SQL operations 796–7  
 uses of 679–80, 683, 686, 687
- READ NEXT  
 definition of 678  
 vs. SQL operations 796–7, 798, 800  
 uses of 680, 683, 684–5

- file operations (*cont.*)
  - record area in
    - definition of 677
    - uses of 677-8
  - and record selection 680-5, 687, 722
  - for relating files 683-6
  - requiring expertise 663, 672
  - REWRITE
    - definition of 677
    - vs. SQL operations 796-7, 798
    - uses of 679-80, 682, 683
  - for scanning records 680-3
  - simplicity of 687, 689
  - for sorting records 682
  - START
    - definition of 678
    - vs. SQL operations 796-7, 798
    - uses of 680, 683, 684-5
  - as traditional operations 672, 673, 691, 796-7
  - WRITE
    - definition of 677
    - vs. SQL operations 796-7
    - uses of 679-80, 682, 683
- file relationships
  - combinations of 740
  - as complex structures 350 n. 3, 688, 747
  - database design based on 741-3, 748-9, 758-9, 760-1, 762, 763-4, 767-8, 784
  - and field relationships 740-2
  - through field values 683-6, 687, 737-8, 739-42, 743-4, 754-5
  - through file nesting 683-6, 687, 739-40, 774
  - as hierarchical structures 688, 740
  - many-to-many
    - definition of 686, 739
    - examples of 686-7, 739-40, 758-9, 763-4
  - many-to-one
    - definition of 686, 739
    - examples of 739, 741-2, 754
  - one-to-many
    - definition of 686, 739
    - examples of 684-5, 739, 741-2, 760-1, 762
  - one-to-one
    - definition of 686, 738
    - examples of 738-9, 741-2, 754, 758-9, 760-1
  - as reflection of field value restrictions 743-5
  - as reflection of the application's requirements 742-4, 745, 747-9, 750, 754-5, 758-65, 766-7, 783-4
- file scanning loops
  - as correct level of abstraction 689, 796, 798
  - definition of 680-1
  - file scanning loops (*cont.*)
    - examples of 680-3, 684-5, 687
    - implicit, in the relational model 704, 774, 796-7
    - nesting of 683-6, 687, 774, 779
    - in SQL 796-7, 800, 801-2
- file systems
  - vs. database systems 673
  - definition of 673
- files
  - data, *see* data files
  - index, *see* index files
  - in the relational model, *see* relational model: files (tables, relations) in
  - relative 674-5
  - sequential 674-5
  - service, *see* service files
  - see also* file operations; file relationships
- Fleming, Candace C.
  - and denormalization 781
  - and the first normal form 776
  - and the relational model 709, 727
- flow-control, as software principle 529
- flow-control attributes
  - and flow-control constructs 580-1
  - and flow-control structures 530, 533-4, 544, 546-7, 579-80
  - and the GOTO statement 529
  - and jumps 529-30, 539, 550-1, 553, 579, 580-1
  - multiple 530, 544-5, 550-1, 579-80
  - and the nesting scheme 531, 544, 546-7, 550-1, 580
- flow-control constructs, *see* software constructs, flow-control
- flow-control features
  - low-level vs. high-level 504
  - as software processes 529
- flow-control structures
  - as complex structures 534, 538, 539, 542-5, 546-7, 548-9, 553-4, 576, 577-8, 583-4, 652
  - and the delusion of the main structure 361-2, 515, 530-1, 536-8
  - and flow-control attributes 530, 533-4, 544, 546-7, 579-80
  - multiple 539-40, 542-5, 576-7, 579-80, 588-9, 649, 652
  - perceived as one structure 515, 530-1, 533-4, 538, 553-4, 649
  - and software processes 545-6
  - as structures in time, vs. structures in space 532, 533, 540, 542
  - as variable structures, vs. fixed structures 541, 544

flow diagrams  
 examples of 513, 524–7, 559  
 ignored in object-oriented programming  
 649, 651  
 reflecting the nesting scheme 512–14, 531,  
 544, 577, 649  
 structured 514, 523–7, 563, 573

flow of execution  
 definition of 503  
 jumps in, *see* jumps, in the flow of  
 execution  
 in object-oriented programming, *see*  
 object-oriented programming: flow  
 of execution in  
 and programming expertise 504–5, 553–4,  
 605  
 in structured programming, *see* structured  
 programming: flow of execution in

flowcharts  
 vs. classification-style diagrams 362  
 and the delusion of the main structure  
 343, 362  
*see also* flow diagrams

formal logic, *see* logic systems

formal reductionism 9, 76, 77–8

fourth-generation languages (4GL)  
 as frauds 441, 452–3, 690–1  
 as low levels in high-level systems 441,  
 452–3, 690

fraud, in research, *see* corruption: in  
 universities

free will  
 degrading the concept of 275–6, 813–14,  
 819–20, 821  
 denied by deterministic social theories  
 826–7  
 denied by mechanistic theories 276–8, 432,  
 813–14  
 denied by software theories 828  
 vs. determinism, and responsibility  
 809–10, 813–14, 819–20, 822–3, 826–9  
 misinterpreted as large number of  
 alternatives 278, 279, 281–4, 286, 814  
 and non-mechanistic capabilities 275–6

freedom  
 vs. conformism 839  
 defending 420  
 degrading the concept of 279–81, 839  
 in software, *see* software freedom  
*see also* free will

Frege, Gottlob, and the idea of a perfect  
 language 318

French Revolution  
 and totalitarian democracy 835  
 and utopianism 130

Freudianism  
 as myth 4–5  
 as pseudoscience 208, 222

Frith, Uta, and contexts 178–9, 183

functions, *see* subroutines

## G

games, idea of, in Wittgenstein's later  
 philosophy 335–9, 340–1, 345

Gardner, Howard, and structuralism 247, 248

general will  
 conformism as reflection of 837, 839, 860  
 as idea, circularity of 837, 838  
 as natural human qualities 837, 860

generations, of programming languages  
 452–3, 489

generative grammar  
 definition of 249  
*see also* Chomskyan linguistics

geometrical method, and hierarchical  
 structures 86, 87, 206

geometry, as hierarchical structures 85

Ghost in the Machine, concept of 147

GOTO statement, *see* structured programming:  
 GOTO statement in

grammar  
 and Chomskyan linguistics 249, 255–6,  
 258–9, 260–1  
 formal, in computer science 259–60  
 as hierarchical structures 110, 260

Greenbaum, Joan M., and the deskilling of  
 programmers 35 n. 2, 484 n. 13

growth, delusions of 22–3

gurus, *see* software theorists

## H

Harris, Roy, and language mechanism 824–5

Hayek, F. A.  
 and central economic planning 848 n. 6  
 and scientism 127, 130

hierarchical metaphor  
 and mathematics 86  
 and mechanism 84

hierarchical structures  
 and abstraction 93–4, 622–3, 637–8  
 in artificial intelligence 143–4  
 as basis of mechanism 8–9, 69  
 as basis of software theories, *see*  
 software structures: perceived as  
 simple structures  
 as brain process 269–70  
 and classification systems 8, 91–2, 96–7,  
 98, 101–2, 629–30

hierarchical structures (*cont.*)

- definition of 83, 90–1, 92–3, 540–1, 622–3
- embodying reductionism and atomism 83–4
- and engineering 84–5
- examples of 8–9, 91–2, 93–4, 96, 98, 151–2, 540, 629–30
- and the geometrical method 86, 87, 206
- geometry as 85
- grammar as 110, 260
- incorrect, when depicting multiple
  - attributes 99–100, 336–9, 354–7, 358–60, 645
- language as 258, 260, 298–300, 307, 626–7
- logic systems as 694, 718
- and logical deduction 85–6, 308, 694
- and manufacturing 84–5, 357
- mathematics as 9–10, 85, 307–8, 714–15, 721–2
- as mechanistic models 83–4, 88–9, 91, 106, 125, 275–6, 281, 814–17
- as mind models, *see* mind models: based on hierarchical structures
- and object-oriented programming 622–3, 628–9, 631–2, 638–40, 644–5, 652
- phenomena perceived as 9–10, 80–1
- software perceived as, *see* software structures: perceived as simple structures
- and structured programming 511–14, 533–4, 536–8, 547–8, 566–7, 604
- see also* simple structures
- holistic knowledge, and skill levels 155–8, 270
- homunculus fallacy
  - definition of 148
  - and mechanism 149–50, 151
  - and mind models 148–50
  - and the phenomenon of vision 148–9
- human beings
  - perceived as deterministic systems
    - in the new pseudosciences 179–81, 275–6, 431–2, 813–14, 819–20, 840–1
    - in software-related affairs 820–1
    - under totalitarianism 822, 838, 847–8
  - perceived as operators of software devices 472–3, 475, 476, 479, 486
- human phenomena
  - indeterministic nature of 10, 21, 87, 125–6, 275–6, 281–5, 699, 814, 819–20
  - as interacting structures 126, 131, 159–60, 205, 275–6, 306
  - software phenomena as 10, 87, 125, 287–8, 487
- human potential, defending 420
- Hume, David, and the problem of induction 211

## humour

- and creativity 175–6
  - and the GOTO debate 600–1
  - as interacting structures 175–6, 184–5
  - as a means to mask irrationality 600, 602
- hypothesis, false, *see* false assumption

## I

- iatromechanics, and scientism 134–5
- IBM Corporation
  - and nested relational databases 778–80
  - and software irresponsibility 804–5
- ideal language, *see* perfect language
- ideologies
  - academic, totalitarian nature of 832–3, 835–6, 840–1
  - business, totalitarian nature of 832–3, 835–6, 842–4
  - mechanistic, conforming to 839–40, 841, 843–4, 848, 854–5, 859
  - mechanistic software, conforming to 197, 409–11, 431–2, 476–7, 828–9, 854–5, 860–1
  - political, and mind models 278–80
  - promoted through deception 832, 851
- IEEE Computer Society, *see* computer associations
- ignorance
  - causing primitive beliefs 42, 43–4, 59–60, 805
  - causing the software bureaucracy 29, 31–2, 419, 804–5
  - causing the software myth 28, 31–2
  - fostered by the software elites 65–7, 289, 420, 421–3, 425–6, 475–6, 479, 486, 805–6
  - promoted through propaganda 66–7
- immunizing stratagems, as tricks to avoid falsifications 223, 224–5
- impoverishment
  - due to abstraction 95, 112–14
  - due to both mechanistic fallacies 120–2, 283, 294, 416, 817
  - due to reification 118–19, 255–6, 257, 416
- of knowledge
  - through acronyms 371–2, 393–4, 401–2, 850
  - through language mechanism 122–3, 364, 398–9, 400–3, 856
  - through slogans 370–1, 383–4, 392, 393–4, 396, 849–50
  - through software mechanism 365, 833–4, 857
- of language 122, 290, 364, 397–9, 400–3, 404–5, 849–50, 856

- impoverishment (*cont.*)  
 of software 124, 288–90, 365, 818, 853–4, 856–7  
 of software-related affairs 113–14, 124, 289–90, 294
- incompetence  
 in programming, *see* programming  
 incompetence  
 in software-related affairs 31–2, 41, 425–6  
 spread of 41, 809
- inconsistencies, data, *see* relational model: data  
 redundancy and inconsistencies in
- indeterminism  
 attempting to reconcile with mechanism 811–12, 813–14, 819–20  
 degrading the concept of 813–14, 815–18, 819–20  
 denied by deterministic social theories 826–7  
 in human phenomena, denied by  
 mechanistic theories 10, 21, 87, 125–6, 275–6, 281–5, 699, 813–14, 819–20  
 misinterpreted as large number of alternatives 814–17  
 and reification 119  
 in software phenomena  
 denied by software theories 87–8, 125, 285, 286–7, 290–2  
 replacing with determinism 290–3, 819–21  
*see also* determinism
- indeterministic models, *see* non-mechanistic models
- indeterministic phenomena, *see* non-mechanistic phenomena
- index files  
 and the current index 677  
 definition of 674–5  
 multiple 675, 687, 740  
 used to define a sorting sequence 675, 682  
 used to identify data records 674–5, 679–80, 686  
 used to scan data records 674–5, 680–6, 687  
*see also* data files
- indexes, *see* index files
- individuality, replacing with conformism 839, 843–4, 848, 859
- indoctrination, mechanistic, *see* mechanistic indoctrination
- induction  
 based on regularities 209–10  
 as basis of pseudosciences and superstitions 209–10  
 as basis of science 209–10  
 and chicken's quandary 210–11
- induction (*cont.*)  
 in discovering theories 213  
 and fallacious decision-making 210–11  
 fallacy of 209–10  
 problem of 209–10  
 and the problem of demarcation 209–10, 211–12
- inefficiency  
 in corporations 37–8, 198, 473–5  
 in programming 32–3, 124, 197, 469–70, 472  
 in society 30, 38–9, 469, 476–7, 490
- informal reductionism 76
- information technology  
 as abstract term 393, 396  
 based on the mechanistic myth 73  
 definition of term 393  
 misuse of term  
 through the acronym “IT” 393–4  
 analysis of 393–4, 395–6  
 compared with other fields 395–6  
 examples of 393–4, 395  
*see also* technology: misuse of term  
 term used as slogan 393–4, 395
- inheritance, in object-oriented programming, *see* object-oriented programming: inheritance in
- innate faculties, idea of  
 arguments against 168, 268–71  
 and linguistics 76, 168, 250–3, 267–71, 278–9, 301, 824–5  
 and the new pseudosciences 282–3, 814
- integrity functions, *see* relational model: integrity functions in
- Intel Corporation, and the term “technology” 386–8
- intelligence  
 artificial, *see* artificial intelligence  
 attempting to represent mechanistically 168–9  
 and intuitive knowledge 142  
 and *knowing how* vs. *knowing that* 146–7, 162  
 and non-mechanistic knowledge 140, 143, 179–81, 192
- interactions, of language structures  
 and Chomskyan linguistics 266, 270–1  
 and deception 369  
 and the idea of a perfect language 303, 306–7, 627
- impaired in schizophrenia 177–8
- with knowledge structures  
 and Chomskyan linguistics 254–5, 256–7, 264  
 and deception 369  
 and the idea of a perfect language 303, 306–7

- interactions, of language structures (*cont.*)  
 with knowledge structures (*cont.*)  
 and Newspeak 122–3, 399, 401–3  
 and non-mechanistic knowledge 120, 164–5  
 and the rationalist fallacies 309–10  
 and similarity to software 290, 298–300, 324–5, 349, 364, 366–7  
 and software totalitarianism 856  
 and Wittgenstein's later philosophy 343–4  
 and Newspeak 399  
 and the rationalist fallacies 309–10  
 and similarity to software 324–5, 349  
 in stories 348–9, 355–7, 399  
 and structuralism 242  
 and totalitarianism 849–50  
 and Wittgenstein's later philosophy 340, 343–4  
 and word meanings 110, 348–9
- interactions, of software structures  
 and the delusion of mathematical programming 87–8, 583–5, 586, 667–8  
 and the delusion of the main structure 343–4, 348, 361–2  
 and file operations 688  
 and flow-control constructs 545–6, 579–82, 583–4, 652  
 ignored by programming aids 123–4, 191, 362–3  
 ignored by software devices 294, 818  
 ignored by software theories 12, 87–8, 361–3, 458–60, 500, 536–8  
 with knowledge structures  
 ignored by programming aids 123–4  
 ignored by software devices 294  
 and similarity to language 298–300, 303, 324–5, 365, 366–7  
 and software totalitarianism 833, 857  
 and the need for low levels 435–8, 614–16, 634–5, 669, 689–90, 691, 710, 723–4, 786, 789–90  
 and object-oriented programming 628–9, 634–6, 638, 639–40, 645, 648  
 and the relational model 226, 710, 720–1, 723–4, 747  
 requiring programming expertise 54, 119, 191, 292–3, 363, 413, 417, 554, 582–3, 688  
 and similarity to language 324–5, 537–8, 614–15  
 and software processes 344, 345–6, 348, 350, 358–61, 363, 434, 528, 545–6, 688  
 and software totalitarianism 853, 857  
 and structured programming 507, 523, 553–4, 577–8, 579–83
- interactions, of software structures (*cont.*)  
 and Wittgenstein's later philosophy 343–5, 358–61
- interactions, of structures  
 causing complexity 85, 103  
 in complex structures 88, 97, 114–17, 120, 162–3, 171, 628, 629–30, 634–5, 638  
 and creativity 173, 175–6, 181–4  
 and distinguishing chicks 167  
 and face recognition 109, 145–6  
 in human phenomena 126, 131, 159–60, 205, 275–6, 306  
 and humour 175–6, 184–5  
 impaired in autism 178–9, 183  
 impaired in schizophrenia 177–8, 183  
 in language, *see* interactions, of language structures  
 loss of 118–19, 121, 203  
 and mechanistic mind models 152, 154–5, 244–5  
 and non-mechanistic knowledge 146, 159–60, 163–5, 314, 321, 415–16, 638, 818–19  
 and non-mechanistic mind models 168–9, 179, 268–71  
 in software, *see* interactions, of software structures  
 and tacit knowledge 159–60, 162–3  
 in a three-body system 107–8  
 and voice recognition 145–6  
*see also* complex structures
- intuition, *see* intuitive knowledge
- intuitive knowledge  
 definition of 157  
 and intelligence 142  
 as manifestation of tacit knowledge 169  
 and non-mechanistic capabilities 162, 269–71, 481  
 required for expert performance 155–8, 166–7  
 and skill levels 155–8
- irrationality  
 blending with rationality 12–13, 46, 49–51, 53–6  
 due to software mechanism 31–2, 53–4, 600  
 in looking for confirmations 218–20  
 masked through humour 600, 602
- irresponsibility  
 causing software failures 472, 807–8  
 in corporations 473–5, 805–6  
 due to language mechanism 824–5  
 due to mechanism 40–1, 272–3, 804, 820, 822–4  
 due to software devices 472–5, 825

irresponsibility (*cont.*)  
 due to software mechanism 40–1, 804,  
 820–1, 858  
 due to the software myth 29, 408, 825  
 fostered by the software elites 606–7,  
 610–11, 805–6  
 justified by the idea of social forces 827, 829  
 justified by the idea of software forces 828–9  
 in programming 472–3, 484, 606–7, 610–11,  
 808–9  
 in software-related affairs 41, 820–1, 825, 828  
 spread of 40–1, 809  
*see also* responsibility  
 ISAM (Indexed Sequential Access Method) 672  
 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 561  
 and the GOTO debate 598  
 and non-standard constructs 575  
 and software engineering 482  
 jokes, *see* humour  
 jumps, in the flow of execution  
 definition of 503–4  
 explicit  
 vs. implicit 506, 515, 529–30, 550–1,  
 553–4, 577, 588–9, 602  
 necessary 554, 572  
 prohibited 524–7, 551  
 and systems of jumps and labels 608–11  
 using GOTO 504, 524–7, 529, 563, 588–9,  
 607  
 and flow-control attributes 529–30, 539,  
 550–1, 553, 579, 580–1  
 implicit, in flow-control constructs 506,  
 515, 529–30, 539, 544, 550–1  
 overriding the nesting scheme 533–4, 539,  
 544, 550–1, 576–7, 579, 588

## K

Kernighan, Brian W., and the GOTO debate 596  
 keys  
 in data files  
 ascending vs. descending 675  
 and the current key 677  
 definition of 674–5  
 duplicate 675

keys (*cont.*)  
 in the relational model 700–1  
*knowing how*  
 and complex structures 162  
 definition of term 146–7  
*knowing that*  
 definition of term 146–7  
 and simple structures 162  
 knowledge  
 and abstraction 153  
 approximating with simple structures 163,  
 416  
 and artificial languages 312–13  
 as complex structures 146, 152–3, 162,  
 163–5, 174, 268–71, 313–15, 321, 415–16,  
 627, 637–8  
 degrading, *see* impoverishment: of  
 knowledge  
 developed through experience, *see*  
 personal experience  
 and the fallacy of abstraction 153, 416  
 growth of, through trial and error 206–7,  
 216  
 impoverishment of, *see* impoverishment:  
 of knowledge  
 intuitive, *see* intuitive knowledge  
 mechanistic, *see* mechanistic knowledge  
 new, uncertainty of 204–5  
 non-mechanistic, *see* non-mechanistic  
 knowledge  
 perceived as simple structures 141, 313–15,  
 320–1, 627, 637–8  
 and reification 119, 152–3, 299, 313–14, 369,  
 398–9, 416  
 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 188, 860–1  
 dependence on, *see* dependence: on  
 knowledge substitutes  
 failure of 27, 141, 158, 198–9, 294  
 and the idea of a perfect society 17  
 as mechanistic delusions  
 causing irresponsibility 473–5  
 and the delusion of high levels 293–4,  
 427, 443–5, 817–18  
 justified by the software myth 26–7, 479  
 vs. non-mechanistic capabilities 141,  
 415–16, 448, 818–19  
 preventing non-mechanistic knowledge  
 417–18, 421–3, 425–6

- knowledge substitutes (*cont.*)  
 as mechanistic delusions (*cont.*)  
   promising to solve non-mechanistic  
   problems 65–7, 293–4  
 mind models as 141, 144–5, 186–7, 415–16  
 perceived as software power 62, 805  
 promoted by the elites  
   causing irresponsibility 473–6  
   and the delusion of high levels 428–9,  
   438–42  
   justified by the software myth 41  
   vs. non-mechanistic capabilities 188,  
   294–5, 518–19  
   preventing non-mechanistic knowledge  
   417–18, 425–6, 860–1  
   promising to solve non-mechanistic  
   problems 65–7, 193–5  
 and skill levels 158  
 spread of 41, 198–9, 293–4  
 Knuth, Donald E.  
   and the GOTO debate 595–6, 599, 601  
   and non-standard constructs 578 n. 17  
 Koestler, Arthur  
   and behaviourism 237, 238  
   and creativity 175, 185  
 Kraft, Philip, and the deskilling of  
   programmers 35 n. 2, 484 n. 13  
 Kurtz, Paul  
   and myths 7  
   and pseudoscience 201

## L

- labels, in the flow of execution  
 definition of 504, 608–9  
 and systems of jumps and labels 608–11  
 language  
 artificial, *see* artificial languages  
 aspects of 254  
 and contexts 120, 254–5, 257  
 controlled by the elite in imaginary society  
   364  
 deception through  
   by coining new terms 367  
   corrupting society 380  
   and doublespeak 380–2  
   due to language mechanism 367–8  
   examples of 371–2, 373, 374–7, 378–82  
   as a form of violence 375  
   preventing 383  
   principles of 370–3, 374–80, 382–3  
   by restricting alternatives 369–70, 382,  
   383, 392, 396  
   by turning words into abstract terms  
   372–3, 374–7, 381–2  
 language (*cont.*)  
 deception through (*cont.*)  
   by using abstract terms as disclaimers  
   374–80  
   by using acronyms 371–2, 393–4, 850  
   by using slogans 370–1, 383–4, 392,  
   393–4, 396, 849–50  
 deficiency in autism 178–9  
 deficiency in schizophrenia 178  
 degraded under totalitarianism 122–3,  
   403–5, 849, 855–6  
 emulating mathematics 307–10  
 formal, in computer science 259–60  
 ideal, *see* perfect language  
 in imaginary society 364  
 impoverishment of 122, 290, 364, 397–9,  
   400–3, 404–5, 849–50, 856  
 as innate faculty 76, 250–3, 267–8, 301,  
   824–5  
 as innate faculty, arguments against 168,  
   268–71  
 as interacting structures, *see* interactions,  
   of language structures  
 as machine in the mind 824–5  
 manipulated by the elites 122, 368, 396,  
   397–8, 404–5, 406–7, 849–50, 851, 856–7  
 as means to represent the world 297–8,  
   366–7, 406–7, 856  
 mechanistic, *see* mechanistic language  
 and the mechanistic fallacies, *see*  
   mechanistic fallacies: in language  
 mechanistic theories of 76, 110, 242, 249,  
   258–9, 266, 267–8, 306–7, 320–1,  
   328–31, 824–5  
 and non-mechanistic knowledge 120,  
   164–5, 254–5, 256–7, 268–71, 404,  
   409–10, 851  
 as non-mechanistic phenomenon  
   and Chomskyan linguistics 266, 268–71  
   and the idea of a perfect language 302–3,  
   306–7  
   and Newspeak 122–3  
   and the rationalist fallacies 309–10  
   and reification 118, 120, 242  
   and similarity to software 324–5, 614–15  
   and Wittgenstein's later philosophy  
   334–5, 341–2  
   and word meanings 12, 110  
 as organ, in Chomskyan linguistics 252,  
   263, 267–8  
 perceived as mapping system 298, 626–7  
 perfect, *see* perfect language  
 programming, *see* programming languages  
 and the rationalist fallacies 306–10, 311–16,  
   637–8

- language (*cont.*)  
 and responsibility 250, 824–5  
 similar to software, *see* software: similar to language  
 and the structuralist phoneme theory 242  
 and Swift's language machine  
   fallacies of 447–9  
   mechanistic basis of 316, 447–9  
   and similarity to programming substitutes 316, 447–50  
   software version of 449–50  
 in Wittgenstein's later philosophy 334–5, 339–40, 341–2  
*see also* linguistics
- language games, idea of, in Wittgenstein's later philosophy 340, 341
- language mechanism  
 and analytic philosophy 317–19  
 causing irresponsibility 824–5  
 circularity of 318  
 degrading minds through 122–3, 364, 397–9, 402–3, 404–5, 406–7, 849, 851, 855–6  
 impoverishing knowledge 122–3, 364, 398–9, 400–3, 856  
 and mind control 122–3, 364, 397–9, 402–3, 404–5, 406–7, 849, 851, 855–6  
 promoted by the elites 249, 824–5, 856  
 in the seventeenth century 306–10, 311–13, 315–16, 447–8, 637–8
- language structures  
 interacting, *see* interactions, of language structures  
 perceived as simple structures 258, 260, 298–300, 307, 626–7
- Laplacean determinism 72, 555, 810–11
- Leach, Edmund, and binary operations 245, 247
- learning  
 and context-dependent knowledge 156  
 and context-free knowledge 156  
 through generic mental processes 168, 269–70  
 and holistic knowledge 155–8  
 of interacting concepts 163–5  
 and intuitive knowledge 155–8  
 and personal involvement 156–7  
 process of 155–8  
 and progression from novice to expert 155–8, 414–15  
 and rule-following 155–6, 157, 270  
 and skill levels 155–8  
*see also* knowledge
- Leibniz, Gottfried  
 and the alphabet of thought 312–13, 314–15
- Leibniz, Gottfried (*cont.*)  
 and artificial languages 312–13  
 and hierarchical structures 304–5  
 and material world vs. mental world 304–5  
 and monadology 305  
 and the universal characteristic 313
- levels  
 in language structures  
 and acronyms 371–2, 393–4, 401–2  
 and deception 372–3, 382–3  
 and the idea of a perfect language 320–1, 328–31  
 and Newspeak 122–3, 400–2  
 and similarity to software 290, 298, 364, 367, 692  
 and slogans 370–1, 383–4  
 and software totalitarianism 856  
 and the term “technology” 384, 387, 388, 392  
 and totalitarianism 849–50  
 and Wittgenstein's later philosophy 339–40  
 and word meanings 372–3, 382  
*see also* levels: in structures
- in software structures  
 and the delusion of high levels 427, 436–42, 455–7, 612, 614–16, 691–2, 786–7  
 emulating manufacturing concepts 287, 439  
 and file operations 670, 673, 689–90  
 and high-level database operations 664–5, 688–9, 786–7  
 and high-level devices 293–4, 428–9, 445–6  
 and high-level programming aids 113–14, 123–4, 191, 291, 451, 612, 634–5, 661–2  
 low in high-level systems 438–45, 452–3, 455–6, 614–16, 634–5, 671, 690, 786–7  
 and the need for low levels 435–8, 614–16, 634–5, 669, 688–90, 691, 710, 723–4, 786, 789–90  
 and ready-made applications 124  
 and the relational model 226, 662–3, 664, 668–9, 688–9, 722–4  
 and software mechanism 288, 290, 417, 428–9, 434–5, 817–18  
 and software totalitarianism 852–3  
*see also* levels: in structures
- in structures  
 definition of 90–1, 94, 540, 622–3  
 embodying reductionism and atomism 83–4  
 and the fallacy of abstraction, *see* abstraction, fallacy of  
 loss of 95, 112–14

levels (*cont.*)

- in structures (*cont.*)
  - in mathematical systems 714–15, 721–3
  - in mind models 149, 151–2
  - 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 155–8, 414–15
- Lévi-Strauss, Claude
  - and analysis of myths 242–3
  - and binary operations 245, 247
  - and classification of social customs 241
  - denying creativity and free will 278
- liberal democracy
  - definition of 835, 859
  - and liberal software concepts 859–61
- life cycles, of software applications, delusion of 464–5
- Linger, Richard C.
  - and Böhm and Jacopini's work 562
  - and error-free programming 518
  - and mathematical programming 487–8, 517
- linguistics
  - Chomskyan, *see* Chomskyan linguistics
  - and scientism 76, 110, 231, 241, 249, 824–5
  - theories in, as myths 824
- little man, *see* homunculus fallacy
- logic, formal, *see* logic systems
- logic systems
  - benefits of 694
  - Boolean logic as 697
  - description of 693, 694–6
  - as hierarchical structures 694, 718
  - informal aspects of 696
  - limitations of 694, 696, 699, 718–19
  - logical operations in 695–6, 698, 716
  - as mathematical systems 693, 714–15
  - as models of phenomena 718–19
  - predicate calculus as 697–8, 718
  - premises in 696
  - propositional calculus as 697
  - syllogistics as 693–4
  - used to explain non-mechanistic phenomena
    - mechanistically 694, 698–9
  - see also* mathematical systems
- logical atomism, Russell's
  - description of 319, 320–1
  - and the idea of a perfect language 319, 320–1, 403
  - as mechanistic theory 320–1
- logical atomism, Wittgenstein's, *see* Wittgenstein's early philosophy

## logical positivism

- description of 321–2
  - and the idea of a perfect language 322–4
  - and the idea of unity of science 323
  - and meaningfulness 322
  - as mechanistic project 322
  - and reductionism 323–4
  - and verifiability 321–2
- logical thinking, difficulty of 204–6
- loop, *see* file scanning loops; software constructs, flow-control: iterative
- Lutz, William, and language abuse 371–2, 375–6, 380–2

## 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
  - in modern society 45
  - as prescientific thinking 44
  - as pseudoscience 44
  - similar to science 46
  - term used in software-related names 42–3
  - see also* software magic
- magic beliefs, due to false assumption 49–50, 52
- main structure
  - delusion of 343–4, 346–7, 348, 351–2, 361–2, 536–8
  - the flow-control structure as 361–2, 515, 530–1, 536–8
  - and flowcharts 343, 362
  - perceived as the application's logic 344, 349, 351–2, 361–2, 515
  - and structured programming 361–2, 515, 530–1, 536–8
- maintenance, of software applications
  - as continuous development 464
  - and the delusion of life cycles 464–5
  - distorted attitude toward 462–4
  - and preference of new applications 413, 462–3, 464–5, 506
  - and programming incompetence 413, 462–3, 505–6
- Malinowski, Bronislaw
  - and magic 45, 46, 47–9, 53, 57
  - and myths 2–3
- mana
  - compared with magic 57

- mana (*cont.*)  
 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, 484, 665–6  
 deskilling of 198, 293–4  
 similar to primitives 805  
 as software bureaucrats 34–5
- manufacturing  
 based on mechanism 18  
 emulated in all activities 18  
 emulated in object-oriented programming  
   618, 619, 621, 647  
 emulated in software theories 27, 32, 34,  
   52, 72–3, 77, 287, 362, 439, 618, 619, 621,  
   852–3  
 emulated in structured programming 517,  
   531–2, 547–8  
 and hierarchical structures 84–5, 357, 439  
 progress in 18
- mapping  
 through language 298, 626–7  
 as one-to-one correspondence to the  
   actual things 298, 301–2, 308, 316,  
   320–1, 322–4, 326, 328–31, 333, 342–3,  
   448, 626–7  
 through a perfect language 301–3, 317,  
   320–1, 322–5, 326, 342–3  
 through software 298, 302–3, 324–5, 327,  
   333, 343–4, 628, 638
- Marcuse, Herbert, and language abuse 371
- Martin, James  
 and CASE 522  
 and denormalization 782  
 and the GOTO debate 592, 606  
 and the idea of software reuse 626  
 and mathematical programming 585 n. 18  
 and non-standard constructs 571  
 and object-oriented programming 618,  
   631, 633, 634  
 and the relational model 727  
 and software engineering 481
- Marxism  
 as myth 4–5  
 as pseudoscience 208, 224
- Maslow, Abraham H., and mechanism 273
- mathematical functions, as high-level  
 operations 670, 689
- mathematical logic, definition of 693
- mathematical systems  
 definition of 714–15  
 emulated with language 307–10  
 as hierarchical structures 9–10, 85, 307–8,  
   714–15, 721–2  
 levels of abstraction in 714–15, 721–3  
 limitations of 721–2  
 logic systems as 693, 714–15  
 manipulation in  
   as formal aspect 716–17, 721–3, 724  
   as main aspect 724  
   vs. translation 724  
 as models of phenomena 9–10, 87, 103,  
   107–8, 154, 586, 714–15  
 simple vs. complex 722  
 translation in  
   as informal aspect 716–17, 724  
   vs. manipulation 724  
   as minor aspect 724  
*see also* logic systems
- mathematics  
 based on tacit knowledge 161–2  
 as basis of science 85–6  
 delusion of, in software theories 87–8,  
   487–8, 509–10, 517–18, 583–5, 585 n. 18,  
   586, 618–19, 666–9, 711, 717–23, 724–5,  
   727, 751–3, 777–8  
 and determinism 87  
 and empirical science 207  
 equivalent to mechanism 9–10, 86–7  
 ideas in, theoretical vs. practical 555, 584–5  
 the world identified with 86–7, 299–300,  
   307–10
- meaning  
 of sentences 254, 257, 309–10, 317–18  
 of words 12, 110, 254, 257, 309–10, 314,  
   348–9, 399, 627
- meaningfulness  
 and logical analysis 317–18  
 and logical positivism 322  
 and Wittgenstein's early philosophy 331–2
- mechanical metaphors  
 in human sciences 132  
 as method of explanation 71, 79–80, 83–4  
 in programming 28, 52–3, 84, 132  
 as source of confidence 79–80
- mechanical models  
 in biology 134–5  
 in economics 133–4  
 as method of understanding 71  
 in politics 135

mechanical models (*cont.*)

in psychology 133, 137–8

in sociology 133, 137–8

mechanical terms, used illegitimately 132,  
136–7

## mechanics

as basis of mechanism 9, 70

perceived as ultimate science 69, 74–5, 76, 78

reducing software to 28, 52–3, 72, 619 n. 12

## mechanism

attempting to reconcile with indeterminism  
811–12, 813–14, 819–20

as bad myth 23–4

based on mechanics 9, 70

as basis of a perfect society 17, 135–6, 822,  
823, 836

as basis of behaviourism 234, 236

as basis of Chomskyan linguistics 259–61

as basis of information technology 73

as basis of manufacturing 18

as basis of science 69

as basis of software engineering 27–8,  
486–9, 493

as basis of structuralism 241

causing charlatanism 20–2, 90, 122, 699, 843

causing irresponsibility 40–1, 272–3, 804,  
820, 822–4

causing scientism 20, 125–6, 127–8, 132–6,  
139, 159–60, 822–3, 836

causing the new pseudosciences 13, 189,

202–3, 231–3, 240, 248, 266, 272, 498, 713

causing totalitarianism, *see* totalitarianism:  
due to mechanism

causing utopianism 131, 831, 836, 858

as convention 71

corrupting societies 431–2, 804, 845–7

definition of 8, 9, 68, 70

denying creativity and free will 276–8, 432,  
813–14

## and determinism

equivalence of 80–1, 125, 810, 811–12,  
813–14

equivalence of, denying 811–12, 813–14

and Laplace's claim 72, 555, 810–11

in models of phenomena 102–3, 104, 105

in software 290–2

as dogma 7, 26, 89, 107, 231–3, 272–3, 489

equivalent to hierarchical structures 8–9,  
69, 84, 125

equivalent to mathematics 9–10, 86–7

failure of mechanists to adhere to 77–8,  
282–4, 285–6

and the homunculus fallacy 149–50, 151

as ideology, conforming to 839–40, 841,  
843–4, 848, 854–5, 859

mechanism (*cont.*)

in language, *see* language mechanism

as method, vs. myth 25, 830

as method of discovery 80–1

as method of explanation 70–1

as method of science 9, 69, 272–3, 499

as myth, *see* mechanistic myth

## naivety of

in behaviourism 236

in Chomskyan linguistics 260

in formal reductionism 811

as old myth 25–6

perceiving all phenomena as simple  
structures 81

reflected in the new pseudosciences

189, 206, 231–2

in software theories 73, 502, 533, 666–7,  
785

in structuralism 244

as philosophy 9–10, 68–9, 141, 811–12

promoted by corporations 830, 831, 842–3

promoted by universities

causing corruption 271–3

causing totalitarianism 16, 823–4, 840–1

degrading the concept of research 231–3,  
728

denying creativity and free will 275–8

through indoctrination 831

through mechanistic software concepts  
39, 728

as myth 7

through the new pseudosciences 200,  
240, 470–1

through propaganda 830

promoted through indoctrination, *see*

mechanistic indoctrination

promoted through propaganda, *see*

mechanistic propaganda

as psychological support 82, 83

as religion 12, 80, 315

similar to political ideologies 831, 851–2

in software, *see* software mechanism

spread of 431–2

success of 69

as trap 128, 417

*see also* mechanistic theories

## mechanistic approximation

adequate for mechanistic phenomena 89,  
357, 812

causing loss of alternatives 815–17

classification systems as 11, 101–2, 104, 357,  
632

and determinism 812, 814–17

inadequate for non-mechanistic phenomena  
89, 107, 128, 305–6, 357, 812–13

- mechanistic approximation (*cont.*)  
 of knowledge 163, 416  
 and mind models 152, 154, 170  
 through reification  
   and application definitions 459–60, 463  
   in artificial intelligence 143  
   in behaviourism 236–7  
   characteristic of the new pseudosciences  
     105, 125, 202–3, 231–2, 498  
   and face recognition 108–9  
   and human phenomena 126, 283, 305–6  
   and language 110, 306–7  
   and mind models 152  
   in object-oriented programming 632  
   in the relational model 710–11  
   in structured programming 536–8, 567  
 simple structures as 11, 104, 143, 146, 231–2,  
   236–7, 460, 536–8, 567, 812, 814–17  
 through software 103, 116–17  
 theories as 104–5, 814–17
- mechanistic capabilities  
 and non-mechanistic problems 23, 25, 66–7  
 and simple structures 162
- mechanistic delusions  
 application definitions as 458–60, 463,  
   465–8  
 causing software delusions  
   and the fallacy of software engineering  
     32–3, 72–3, 481–2, 486–9, 491–2  
   and inadequacy of software devices  
     189–90, 192–3, 415–16, 417–18, 431–2  
   and object-oriented programming  
     618–19, 626–8  
   and the relational model 666–7, 765–7  
   and similarity to language 297–8, 300,  
     302–3, 324–5, 626–8  
   and structured programming 517–19,  
     530–3, 536–8, 547–8, 554–5, 583–5  
   and Wittgenstein's early philosophy 333  
 consequences of 21, 32–4, 81, 204, 272–3, 366  
 exploited by software companies 188,  
   417–18, 419, 425–6  
 knowledge substitutes as, *see* knowledge  
 substitutes: as mechanistic delusions  
 mechanism as cause of 69, 81, 89, 105–6, 499  
 method to guard against 205  
 methodologies as 458–9, 461–2, 465–7  
 mind models as, *see* mind models: as  
   mechanistic delusions  
 object-oriented programming as 628–9,  
   631–2, 638–40, 653  
 perfect languages as 299–300, 302–3,  
   317–18, 321, 324–5, 627–8  
 in programming 32–3, 36, 192–3, 288–90,  
   443–5, 469–71, 487–90
- mechanistic delusions (*cont.*)  
 programming substitutes as 189–90, 300,  
   409–10, 413, 417, 425–6, 448–9, 450  
 ready-made products as 842–3  
 relational model as 666–7, 699, 713, 734–5,  
   765  
 and scientific thinking 75  
 software devices as 27, 188, 434–5, 614–16  
 software ease of use as 426–8, 438–40  
 software power as 426–8, 438–40  
 software theories as, *see* software theories:  
   as mechanistic delusions  
 structured programming as 515–16, 530–3,  
   536–8, 547–8, 553–4, 566–7, 576–7,  
   585–6, 603, 653  
 totalitarianism as 21, 24, 274, 838, 845–6
- mechanistic explanation  
 based on mechanical metaphors 71, 79–80,  
   83–4  
 based on reductionism and atomism 80–1  
 definition of 70  
 identified with mathematics 85–7  
 by separating structures 119
- mechanistic fallacies  
 abstraction, *see* abstraction, fallacy of  
 causing loss of alternatives, *see* alternatives,  
   loss of  
 committed together 120–1, 283, 294, 416,  
   817, 847  
 in language  
   and acronyms 371–2, 394, 401–2, 850  
   and deception 369–70, 382–3  
   and the idea of a perfect language 306–7,  
     313–15, 321  
   and linguistics 118, 255, 256–7  
   and Newspeak 122–3, 398–9, 400–4  
   and similarity to software 290, 297–300,  
     364, 367–8, 406–7, 447–9, 824–5, 855–6  
   and slogans 370–1, 849–50  
   and Swift's language machine 447–9  
   and the term "technology" 392  
   and totalitarianism 849, 851  
 and mechanistic models 121–2, 124–5  
 and mind models 152–3  
 and non-mechanistic models 121–2, 124–5  
 reification, *see* reification, fallacy of  
 in software  
   and application definitions 460  
   causing loss of alternatives 123–4, 363,  
     365, 416  
   and the delusion of high levels 288–90,  
     434–5, 436–7, 612, 614–16, 636, 661–3,  
     668–9, 786–7  
   due to emulating manufacturing  
     concepts 287, 408, 409, 547–8

- mechanistic fallacies (*cont.*)  
 in software (*cont.*)  
 due to programming incompetence 119, 124  
 and high-level devices 434, 818–19  
 and high-level programming aids 113–14, 190–1, 417, 668–9  
 and knowledge substitutes 188, 415–16  
 and mechanistic theories 500–1, 612, 661–2, 699  
 and similarity to language 297–300, 364–5, 406–7, 419, 628  
 and software totalitarianism 852–3
- mechanistic indoctrination  
 through advertising 831, 843  
 through education 16, 831, 841
- mechanistic knowledge  
 as novice level 155–6, 162–3, 167, 421–3, 474, 490, 820  
 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 364, 849–50, 851, 856  
 prevalence of 404–5, 850  
 preventing non-mechanistic knowledge 367–8, 404–5, 851, 856  
 and progression toward totalitarianism 397, 403–5, 850, 855–6  
*see also* language mechanism
- mechanistic models  
 adequate for mechanistic phenomena 10, 69, 81, 87, 88–9, 102–3, 106, 357, 486–7, 566–7  
 as approximation, *see* mechanistic approximation  
 in artificial intelligence 143–4, 260  
 hierarchical structures as 83–4, 88–9, 91, 106, 125, 275–6, 281, 814–17  
 and large number of alternatives 281–4, 814–17  
 and the mechanistic fallacies 121–2, 124–5  
 and non-mechanistic phenomena 89, 102–3, 105–7, 202–3, 555  
 and the three-body system 107–8
- mechanistic myth  
 alternatives to 25  
 as belief system 1, 7, 14, 82  
 causing programming incompetence 32–3  
 dependence on 25, 26  
 due to reductionism and atomism 82  
 and non-mechanistic phenomena 10  
 origins of 70
- mechanistic myth (*cont.*)  
 software as 26, 28, 31, 33  
*see also* software myth
- mechanistic performance  
 as novice level 155–6, 162–3, 167, 414  
 restriction to 421–3, 819, 820–1, 822, 841, 843–4
- mechanistic phenomena 10, 69, 80–1, 87, 88, 102–3, 106, 357, 486–7, 566–7
- mechanistic problems 66, 193–5, 417–18
- mechanistic propaganda  
 conducted by corporations 830, 842–3  
 conducted by universities 830, 840
- mechanistic software  
 restriction to 29, 66–7, 273, 289, 409–11, 432, 852–4, 856–7, 858, 860–1  
*see also* software mechanism
- mechanistic solutions  
 products as 842  
 and the promise to solve non-mechanistic problems 21, 65–7, 420–3, 840, 842–3, 851, 854, 858  
 software devices as, *see* software devices
- mechanistic theories  
 and analogies 77, 244  
 as approximation, *see* mechanistic approximation  
 in artificial intelligence 143–4  
 based on the idea of innate faculties 282–3, 814  
 circularity of 132, 150, 152, 205, 253, 268, 318, 837–8  
 as delusions, *see* mechanistic delusions  
 failure of  
 due to misinterpreting creativity 281–4, 286–8  
 due to perceiving minds as machines 275–6, 840  
 masked through informal analogies 77  
 in the new pseudosciences 189, 232–3, 555, 713–14  
 as frauds 81, 499  
 for human phenomena 126, 127–8, 132–8, 275–6, 814–17  
 and language 76, 110, 242, 249, 258–9, 266, 267–8, 306–7, 320–1, 328–31, 824–5  
 and mechanistic phenomena vs. non-mechanistic phenomena 10, 69, 205, 231–2  
 of mind, *see* mind models  
 and non-mechanistic capabilities 141, 192–3, 324–5, 415–16, 582–3, 648, 820–1  
 for non-mechanistic phenomena  
 based on reification, *see* mechanistic approximation: through reification

- mechanistic theories (*cont.*)  
 for non-mechanistic phenomena (*cont.*)  
 in behaviourism 234, 236–7, 276–7  
 in Chomskyan linguistics 254–6, 261–3,  
 278–80  
 as delusions 87, 106–7, 131, 205, 499,  
 814–17  
 in economics 847  
 inadequacy of 10–12, 102–3, 105–6,  
 159–60, 231–3  
 in the new pseudosciences 189, 231–3,  
 275–6, 281–5, 430–2, 713–14  
 in software 192–3, 363, 487–9, 612,  
 628–9, 699, 711  
 in structuralism 242–5, 246–8, 278  
 as poor approximations  
 of human phenomena 126, 128, 137–8,  
 305–6, 814  
 of non-mechanistic phenomena 811,  
 812–13, 815–17  
 of software phenomena 460, 536–8, 567,  
 667  
 popularity of 81  
 similarity of 83  
 in software, *see* software theories  
 ways to identify 125  
*see also* mechanism
- mechanistic understanding, *see* mechanistic  
 explanation
- media  
 employing confirmations rather than  
 falsifications 219–20, 613, 660, 808  
 and software propaganda 220, 522–3, 613,  
 660, 808
- medical technology, analysis of term 395–6
- messages, in object-oriented programming,  
*see* object-oriented programming:  
 messages in
- metaphors  
 and creativity 174  
 hidden 84, 85  
 hierarchical 84, 86  
 immediate 84  
 mechanical, *see* mechanical metaphors
- method, geometrical, and hierarchical  
 structures 86, 87, 206
- method of science, mechanism as 9, 69,  
 272–3, 499
- methodologies  
 and application definitions 458–60  
 and CASE 467–9  
 as frauds 461–2  
 as low levels in high-level systems 462  
 as mechanistic delusions 458–9, 461–2,  
 465–7
- methodologies (*cont.*)  
 requiring non-mechanistic knowledge  
 466–7  
 as substitutes for programming expertise  
 144–5, 195–7, 461–2  
*see also* programming substitutes
- methods, for logical thinking 205–6
- methods of inquiry, vs. magic practices 44
- Microsoft Corporation  
 fostering ignorance and dependence 475  
 fostering programming incompetence  
 472–3
- Mills, Harlan D.  
 and Böhm and Jacopini's work 562, 591  
 and error-free programming 518, 520–1  
 and the GOTO debate 591  
 and mathematical programming 487–8, 517  
 and structured programming 515–16, 589 n. 1
- mind  
 and artificial intelligence 143–4, 181–4  
 and body, perceived as separate entities  
 147, 304–6, 308  
 and complex structures 162, 192, 245,  
 268–71, 414–16  
 controlling, *see* mind: degrading  
 degrading  
 through language mechanism 122–3,  
 364, 397–9, 402–3, 404–5, 406–7, 849,  
 851, 855–6  
 through software devices 188, 409–11,  
 421–3, 427–9, 818–19, 820–1  
 through software domination 411,  
 425–6, 854  
 through software mechanism 365,  
 409–11, 431–2, 856–7  
 disorders, and creativity 176–9, 183  
 and the Ghost in the Machine 147  
 mechanistic capabilities of, *see* mechanistic  
 capabilities  
 mechanistic theories of, *see* mind models  
 as natural phenomenon 154–5  
 non-mechanistic capabilities of, *see* non-  
 mechanistic capabilities  
 and non-mechanistic problems 23, 25,  
 66–7, 140, 417–18, 843–4  
 potential of, as myth 25  
 represented with hierarchical structures  
 141–2  
 and simple structures 162, 191–2, 414–16
- mind control, *see* mind: degrading
- mind mechanism, *see* mind models
- mind models  
 based on analogy to machines 179–80  
 based on hierarchical structures  
 in artificial intelligence 143–4, 260

- mind models (*cont.*)
- based on hierarchical structures (*cont.*)
    - due to the mechanistic philosophy 141–2
    - and knowledge substitutes 293–5, 818
    - as mechanistic delusions 147–8, 151–2
    - and programming substitutes 291–2
    - in structuralism 243–5
  - based on mechanistic theories
    - in artificial intelligence 143–4
    - in behaviourism 234–5
    - in Chomskyan linguistics 259–60, 266, 268, 279–80
    - circularity of 151–2
    - failure of 143–4, 170, 820
    - fallacies of 147–8, 152–3
    - in structuralism 243–5
  - as basis of software devices 188, 291–2, 818
  - as basis of software theories 144–5, 190, 291–2, 820
  - limitations of 181–5, 186–7
  - as mechanistic approximation 152, 154
  - as mechanistic delusions
    - in artificial intelligence 151–2, 260
    - in Chomskyan linguistics 266, 269–71
    - and contexts 186–7
    - and creativity 170, 179–81
  - and the mechanistic fallacies 152–3
  - non-mechanistic 168, 179, 268–71
  - and political ideologies 278–80
  - programming substitutes as 190, 291–2
  - and rationalism vs. empiricism 278–80
  - as substitutes for knowledge 141, 144–5, 186–7, 415–16
- Minsky, Marvin L., and artificial intelligence 143
- mirroring, *see* mapping
- MMX technology, analysis of term 386–8
- 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 212
- mythology, *see* myths
- myths
- ancient 2–4
  - bad 15–16, 23–4
  - based on false assumption 14
- myths (*cont.*)
- 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 242–3
  - transition between 15, 24–6

## N

- Nagel, Ernest, and mechanism 69, 72
- naivety
- of artificial intelligence 143, 259–60, 429
  - of determinism 72
  - of the early scientists 80–1
  - of mechanism, *see* mechanism: naivety of
  - of modern scientists 81, 143, 189, 231–2, 236, 244, 259–60
  - of the software theorists 502, 533, 563, 578, 603–4, 650, 667–8, 713, 734–5, 785
- natural order, idea of
- as scientism 836
  - used to justify totalitarianism 837
- Nazism
- degrading the concept of responsibility 822
  - as myth 6
  - as totalitarian democracy 835
  - and utopianism 130
- nesting
- of file scanning loops 683–6, 687, 774, 779
  - of files 683–6, 688
  - of files, in the relational model, *see* relational model: file nesting in
  - of flow-control constructs 511–14, 568, 683
  - of messages, in object-oriented programming 650, 652, 653
- 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 202, 498, 713
  - due to mechanism 13, 189, 202–3, 231–3, 240, 248, 266, 272, 498, 713

- Newspeak  
 and abstraction 400–3  
 and acronyms 401–2  
 eliminating words and meanings 398–9  
 and the GOTO debate 601  
 and the idea of a perfect language 403–4  
 inventing new words 400–1  
 and mind control 122–3, 397–8  
 and reification 398–9  
 and similarity of language and software 409–10  
 simplifying the grammar 402–3
- Newton's theory of gravitation  
 contradicting mechanism 78  
 as deterministic theory 71–2  
 as inspiration in human sciences 132  
 interpreted as mechanistic 80  
 as mathematical model 714–15  
 as mechanistic approximation 104
- non-mechanistic approximation, complex structures as 146, 168–9
- non-mechanistic capabilities  
 and complex structures 108–9, 110, 145–6, 162, 192–3, 415–16, 487, 490, 579–80  
 and creativity 170, 172–4, 175–6, 275–6, 428–9  
 and evolution 140, 152, 269  
 and free will 275–6  
 and intuitive knowledge 162, 269–71, 481  
 and mechanistic theories 141, 192–3, 324–5, 415–16, 582–3, 648, 820–1  
 and multiple attributes 102, 314, 335–9, 354–7  
 neglecting 25, 140, 188, 421–3, 431–2, 433, 445–6, 818–19, 844, 851–2  
 and non-mechanistic mind models 168–9, 179, 268–71  
 and non-mechanistic problems 23, 25, 66–7, 140, 417–18, 843–4  
 and non-mechanistic theories 126, 128  
 and recognizing contexts 171, 182–3  
 vs. software devices 66–7, 186–7, 188, 194–5, 288–90, 415–16, 421–3, 431–2, 433, 434, 445–6, 819
- non-mechanistic knowledge  
 appreciation of 433  
 craftsmanship as 483, 490, 491  
 development of 146, 155–8, 159–61, 162–3, 164–5, 166–7, 169, 268–71, 414–15, 421–3, 433  
 and intelligence 140, 143, 179–81, 192  
 as interacting structures 146, 159–60, 163–5, 314, 321, 415–16, 638, 818–19  
 and language 120, 164–5, 254–5, 256–7, 268–71, 404, 409–10, 851
- non-mechanistic knowledge (*cont.*)  
 prevention of  
 due to denying creativity and free will 818–19, 841  
 due to mechanism 23, 25  
 due to mechanistic language 367–8, 404–5, 851, 856  
 due to software devices 65–7, 288–90, 409–11, 417–18, 420–3, 428–9, 430, 431–2, 433, 445–6  
 due to software mechanism 66–7, 368, 431–2, 474, 475, 490, 518–19, 580, 585–6  
 due to totalitarianism 843–4, 851–2, 854, 856–7, 860–1
- in programming  
 and the delusion of high levels 444–5, 661–2, 817–18  
 and the fallacy of software engineering 481, 487, 490, 493, 501  
 and interacting structures 553–4, 579–83, 585–6, 648, 667–8  
 and methodologies 466–7  
 prevention of 192–3, 292–3, 409–11, 491–2  
 and reification 119, 152–3, 299, 313–14, 369, 398–9, 416  
 in software-related affairs 66–7, 191, 425–6, 466–8, 474, 475  
 and stories 355–7  
*see also* tacit knowledge
- non-mechanistic models  
 interacting structures as 88  
 and the mechanistic fallacies 121–2, 124–5  
 of mind 168, 179, 268–71  
 and non-mechanistic phenomena 89
- non-mechanistic phenomena  
 and charlatanism 65–7, 843  
 and determinism 89, 232, 281–4, 812–13  
 examples of 107, 108, 110, 145–6, 815–16  
 language as, *see* language: as non-mechanistic phenomenon  
 and mechanistic models 89, 102–3, 105–7, 202–3, 555  
 and the mechanistic myth 10  
 and mechanistic naivety 81, 189, 231–2, 236, 244  
 mechanistic theories for, *see* mechanistic theories: for non-mechanistic phenomena  
 non-mechanistic theories for 204  
 in programming 192–3, 196, 487  
 and reification 114–16, 231–2, 236, 242, 255, 847  
 software as, *see* software: as non-mechanistic phenomenon

- non-mechanistic phenomena (*cont.*)  
 and software theories 123-4, 192-3, 204, 853  
*see also* complex structures
- non-mechanistic problems  
 and inadequacy of software devices 65-6,  
 193-5, 415-16, 417-18, 420-1  
 and mechanistic capabilities 23, 25, 66-7  
 and the mind 23, 25, 66-7, 140, 417-18,  
 843-4  
 and non-mechanistic capabilities 23, 25,  
 66-7, 140, 417-18, 843-4  
 and the promise of mechanistic solutions  
 21, 65-7, 420-3, 840, 842-3, 851, 854,  
 858
- non-mechanistic theories  
 and human phenomena 126, 127-8  
 required for non-mechanistic phenomena  
 204
- normal forms, *see* normalization, in the  
 relational model
- normalization, in the relational model  
 1NF (first normal form) in  
 annulment of 772-3, 775-6  
 definition of 704, 706, 772  
 examples of 704-5, 772-3  
 as formal principle 707, 729, 733-4, 770,  
 777  
 importance of 775-6  
 inefficiency of 771, 773, 773 n. 2, 778-9
- 2NF, etc. (higher normal forms) in  
 annulment of 780-1  
 definition of 704, 706, 731  
 examples of 758-65  
 inefficiency of 732, 755, 771, 780  
 as informal principle 707, 734, 770  
 and optimal normal forms 755  
 and updating operations 707, 729, 733-4,  
 755, 756, 787
- based on fallacious arguments 766-7  
 definition of 704, 729, 770
- denormalization in  
 definition of 781  
 as falsification of the relational model  
 771, 780-2, 783-5  
 as informal procedure 782-4  
 misrepresented by the theorists 781-3  
 perceived as exact method 782-4  
 procedure of 782-3  
 as reversal of normalization 729, 771,  
 780-2  
 vs. traditional design principles 781,  
 783-4
- and field dependencies 705, 747-9, 751-3,  
 754-7, 758, 759-60, 761-3, 764-5, 788  
 as fraud 712, 729, 748, 757, 769
- normalization, in the relational model (*cont.*)  
 as game 753  
 as independent theory 732, 765  
 inefficiency of 732, 771  
 as informal procedure 707-8, 712, 724, 734,  
 747, 748-9, 750-1, 754-5, 758-65,  
 783-4, 787-8  
 manipulation in, vs. translation 752  
 mathematical claims of 729, 734, 750,  
 752-3  
 as mechanistic delusion 728-9, 734-5,  
 750-1, 757, 765, 769, 787-8
- non-1NF (non-first-normal-form) in  
 as annulment of 1NF 772-3, 775-6  
 definition of 773-4  
 as falsification of the relational model  
 774, 775-6, 777, 780  
 as informal principle 777  
 promotion of 774-5, 778-80  
 as replacement of JOIN operations 774-5
- vs. normalization in traditional databases  
 706, 731-2, 743, 744-5
- and ordinary field dependencies 756-7, 788
- overnormalization as formal procedure of  
 755  
 perceived as formal design principles 712,  
 728-9, 734-7, 747, 748-9, 750-1,  
 758-65, 766-7, 768-9, 783-4, 787-8  
 perceived as substitute for expertise 746,  
 747-8, 750  
 procedure of 706, 731, 747-8, 755, 771  
 and similarity to the traditional file  
 relationships 744-5, 748-9  
 as specious mathematics 751-3  
 as system of representation 752-3
- vs. traditional design principles 748-9,  
 750-1, 758-65, 767-9, 784  
 translation in, vs. manipulation 752  
 as unnecessary theory 712, 735-8, 743, 745,  
 748-9, 750-1, 754, 758
- novice performance, characteristics of 155-8,  
 162-3, 414

## O

- object-oriented programming  
 and abstraction 622-3, 637, 643, 655-6  
 attributes in  
 and complex structures 628-9  
 definition of 622  
 and the modified concept of inheritance  
 644-5, 646
- classes in  
 as basis of software reuse 624-5  
 definition of 623

object-oriented programming (*cont.*)

- classes in (*cont.*)
  - as hierarchical structures 623, 631-2
  - as templates 624
  - used as subroutines 642, 643
- and classification systems 622-3, 624-5, 628-9, 631-2, 638-40, 644-5, 646
- delusions of, summarized 636-7
- difficulties of 633-4, 648
- downgraded to informal theory 642, 646, 648
- emulating manufacturing concepts 618, 619, 621, 647
- encapsulation in 656
- expanded in order to suppress falsifications
  - by allowing any attributes 647
  - by allowing informal principles 655-8, 659
  - by allowing interacting hierarchies 635-6, 637, 641-2
  - by allowing multiple flow-control structures 653
- failure of 635, 647, 653, 657-8, 661-2
- fallacies of
  - allowing interacting hierarchies 641-2, 644, 647, 648
  - attempting to represent applications mathematically 618-19
  - claiming both simplicity and difficulty 632-4
  - ignoring the flow of execution 648-50
  - ignoring the significance of hybrid languages 642
  - ignoring the structures generated by messages 652, 653, 654
  - misinterpreting the problem of multiple hierarchies 641-2
  - misinterpreting the problem of one hierarchy 638-9
  - perceiving inheritance as modifiable principle 643-4, 645-6
  - perceiving software mechanism as substitute for expertise 661-2
  - promising automatic application development 619
  - reinstating the traditional programming concepts 634-5, 641-2, 646, 647-8, 654, 656
  - replacing the original promises with trivial concepts 655-8, 659
  - restricting relations to one hierarchy 628-9, 631-2, 634-5, 637-40
  - reversing the principles of programming 615-16, 634-5
  - using classes as subroutines 642, 643

object-oriented programming (*cont.*)

- flow of execution in
  - between objects 651-2, 653
  - as complex structure 652, 653
  - ignored by the theory 648-50
  - within objects 652
  - same as in structured programming 651-2, 652 n. 3
  - and state transition diagrams 651 n. 1
- as fraud 636, 637, 654
- and hierarchical structures 622-3, 628-9, 631-2, 638-40, 644-5, 652
- information hiding in 656
- inheritance in
  - allowed to change and omit attributes 644-5
  - allowed to take attributes from other classes 645-6
  - benefits of 623-4
  - definition of 623, 643
  - modifying the concept of, as falsification of the theory 643-4, 645-6
  - multiple 646
  - obviated by using classes as subroutines 642, 646
  - as reverse of abstraction 623, 643
  - and interacting software structures 628-9, 634-6, 638, 639-40, 645, 648
- as low levels in high-level systems 634-5, 636, 661-2
- as mass madness 617-18
- mathematical claims of 618-19
- as means to control programming practices 661-2
- as mechanistic delusion 628-9, 631-2, 638-40, 653
- messages in
  - definition of 622
  - determining the flow of execution 649-50, 651-2, 653
  - generating structures 654
- methods in 649-50, 652
- and misuse of the terms "object" and "object-oriented" 619-20, 655, 658-9
- nesting scheme in
  - generated by messages 650, 652, 653
  - vs. structured programming 649-50, 651-2
- objects in
  - definition of 617, 622
  - degrading the concept of 659
  - as hierarchical structures 622-3, 649-50
  - as instances of classes 624, 649-50
  - similar to subroutines 622, 652, 653-4
  - as software principle 653-4

object-oriented programming (*cont.*)  
 objects in (*cont.*)  
   as software processes 653–4  
 origins of 617  
 overloading in 657  
 perceived as a new paradigm 617, 621  
 perceived as a revolution 616–17, 626, 632  
 polymorphism in 656–7  
 preventing interactions between structures 639–40  
 and programming incompetence 621–2, 661–2  
 promises of 616–18, 620–1, 625–6, 630–1, 647, 649  
 promotion of 616–18, 620–1, 632–3, 659–60  
 as pseudoscience 635, 637, 641, 647, 653, 658  
 and reification 628, 631–2, 638  
 requiring non-mechanistic knowledge 648, 662  
 requiring traditional languages 641–2, 643  
 and software reuse  
   through one class hierarchy 624  
   principles of 617, 624–5  
   through several class hierarchies 625  
 and software wrapping 659  
 and structures, static vs. dynamic 650  
 objects, in object-oriented programming, *see*  
   object-oriented programming:  
   objects in  
 one-to-one correspondence, *see* mapping  
 operationalism, and scientism 137  
 operations, between elements of structure 90–1, 534, 540, 623  
 operations, logical  
   in logic systems 695–6, 698, 716  
   in predicate calculus 698, 701–3, 718  
 operations, software  
   on files, *see* database operations; file  
   operations  
   low-level vs. high-level, *see* levels: in  
   software structures  
   in programming languages, definition of 503 n. 1  
   in the relational model, *see* relational  
   operations  
   repeated, as software processes 353  
   in Turing machines 555–6  
 Orwell, George  
   and central economic planning 846, 848  
   and corruption of the intellectuals 845  
   and mechanistic language 398–9, 400–5, 849, 850, 856  
   and mind control through language 122–3, 396–9, 400–3, 848–9

Orwell, George (*cont.*)  
   and origins of *Nineteen Eighty-Four* 396–7, 844–5, 848–9  
   and progression toward totalitarianism 397, 403–5, 844–5, 846, 850, 851, 857  
   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 79  
 Pascal, Blaise, and intuitive knowledge 142  
 perfect language  
   idea of 300, 317–19, 342–3, 403–4, 626–7  
   idea of, in software 300, 324–5, 343–4, 628  
   and logical positivism 322–4  
   as mapping system 301–3, 317, 320–1, 322–5, 326, 342–3  
   as mathematical language 318, 320–1  
   as means to represent the world 299–300, 301–2, 317–19, 323–4, 328–31, 626–7  
   as mechanistic delusion 299–300, 302–3, 317–18, 321, 324–5, 627–8  
   and Russell's theory of logical atomism 319, 320–1, 403  
   and the story of Babel 301–2  
   *see also* artificial languages  
 perfect society  
   as aim of totalitarianism 16–17, 18–19, 131–2, 822  
   based on knowledge substitutes 17  
   based on mechanism 17, 135–6, 822, 823, 836  
   based on science and conformism 835, 852  
   through behaviourism 277  
   idea of 17–18, 129–30, 131–2  
   and liberal democracy vs. totalitarian  
   democracy 835  
 personal experience  
   based on whole knowledge structures 428–9  
   as basis of non-mechanistic knowledge 146, 147, 162–3, 414–15  
   impossible for mind models 154, 180–1, 186  
   and language acquisition 270–1  
   prevented by the software elites 421–3, 425–6  
   in programming  
     prevented by software devices 445–6  
     required for expertise 191, 196  
     and skill acquisition 155–8, 159–60  
 Pettit, Philip, and structuralism 246

- 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, 80–1, 87, 88, 102–3, 106, 357, 486–7, 566–7  
 non-mechanistic, *see* non-mechanistic phenomena  
 perceived as hierarchical structures 9–10, 80–1  
 represented mathematically 9–10, 87, 103, 107–8, 154, 586, 714–15  
 software, *see* software phenomena  
 supernatural, software as, *see* software magic; software power
- philosophy, mechanism as 9–10, 68–9, 141, 811–12
- Piaget, Jean  
 and binary operations 243–5  
 denying creativity and free will 278
- Plato, and totalitarianism 129
- Poincaré, Henri, and creativity 172–3
- pointer, *see* file operations: pointer in
- Polanyi, Michael  
 and the learning process 165  
 and tacit knowledge 158–62
- politics  
 and scientism 135–6, 241, 822, 836  
 turned into morbid preoccupation 860
- Popper, Karl R.  
 and the criterion of demarcation 211–12, 214–16, 222–4, 230  
 and determinism 89–90, 812  
 and empiricism 214  
 and Freudianism 222  
 and the growth of knowledge 206–7, 213  
 and the idea of falsifiability 214–16  
 and the idea of verifiability 322, 332  
 and Marxism 224  
 and the Popper legend 230  
 and the problem of demarcation 206, 207, 208–9  
 and the problem of induction 209, 211–12  
 and reductionism 82  
 and totalitarianism 129
- power  
 as abstract term 58  
 definition of term 58  
 term used in software-related names 59, 60  
 term used in the sense of mana 58–60, 62–5  
 term used in various grammatical roles 63–5  
*see also* mana; software power
- practices  
 business, definition of 346  
 software, definition of 346  
 as software principle 346–8, 528–9  
 as software processes 346–8, 359  
 as structures 346–7, 359–60
- predicate calculus  
 as basis of the relational model 699–700, 701–2, 707, 711, 715–16, 718, 732–3, 751, 777  
 description of 697–8  
 examples of 697–8  
 limitations of 719–20  
 as logic system 697–8, 718  
 operations in 698, 701–3, 718  
 relations in 697, 718  
 tuples in 697, 718, 732–3
- primitive beliefs, due to ignorance 42, 43–4, 59–60, 805
- 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
- principles of demarcation, *see* demarcation: principles of
- problems  
 mechanistic 66, 193–5, 417–18  
 non-mechanistic, *see* non-mechanistic problems  
 software-related, *see* software-related problems
- processes, software, *see* software processes
- products  
 dependence on 842, 844  
 as mechanistic solutions 842  
 software perceived as 42 n. 1, 852–4
- programmers  
 application vs. system 36, 197  
 avoiding programming 196, 417, 445–6, 472, 805–6  
 compared to craftsmen 480, 483–5, 491, 621–2, 661  
 dependent on programming substitutes 32–3, 54–5, 196–7, 292–3, 409–11, 413, 445–7, 454–6, 472–3, 612, 661–2, 778, 805–6  
 deskilling of 34–5, 35 n. 2, 472–3, 479, 483–6, 484 n. 13, 604–7, 611–12, 661–2  
 education of 32, 195, 412–13, 483, 491–2, 605, 607  
 incompetence of, *see* programming incompetence  
 perceived as magicians 56

- programmers (*cont.*)  
   similar to primitives 54-5, 65-7, 805  
   as software bureaucrats 32, 35-6, 412-13,  
     472, 479, 484-6, 611  
   spurious activities of 54-5, 197, 469-71  
   as unskilled workers 34-5, 196-7, 483-4,  
     604-5, 606-7, 661, 664-5
- programming  
   avoided, as principle 196, 417, 445-6, 472,  
     805-6  
   expertise in, *see* programming expertise  
   incompetence in, *see* programming  
     incompetence  
   inefficiency in 32-3, 124, 197, 469-70, 472  
   irresponsibility in 472-3, 484, 606-7,  
     610-11, 808-9  
   mechanistic delusions in 32-3, 36, 192-3,  
     288-90, 443-5, 469-71, 487-90  
   non-mechanistic knowledge in, *see* non-  
     mechanistic knowledge: in  
     programming  
   non-mechanistic phenomena in 192-3,  
     196, 487  
   reversing the principles of 451, 455-6,  
     615-16, 634-5, 671  
   science of, software theories perceived as  
     502, 509, 603-4  
   *see also* software engineering; software  
     theories
- programming aids, *see* programming  
   substitutes
- programming expertise  
   as capacity for complex structures 119, 191,  
     292-3, 363, 413, 417, 419, 491-2, 505,  
     553-4, 582-3  
   as capacity for low levels 365, 417, 419, 688  
   as craftsmanship 491-2, 501  
   definition of 412  
   degraded, vs. traditional expertise 412-13,  
     433, 470, 490, 491-2, 607, 612  
   degrading the concept of, *see* program-  
     ming substitutes: replacing expertise  
   prevention of 34-5, 54-5, 195-7, 292-3,  
     409-11, 412-13, 473, 479, 483-6, 606-7  
   replacing with substitutes, *see* program-  
     ming substitutes: replacing expertise  
   standards for 34, 195, 607, 808, 808 n. 3
- programming incompetence  
   accepted by society 34, 607  
   causing dependence on the software elites  
     445-7, 461-2, 612, 661-2, 778, 802-3  
   causing poor application maintenance 413,  
     462-3, 505-6  
   causing software failures 32-3, 469-71,  
     807-8
- programming incompetence (*cont.*)  
   causing the software bureaucracy 34-5,  
     37-8, 196-7, 409-10, 665-6  
   and the concept of expertise 484, 491-2  
   cost of 32-3, 198, 414, 472, 476, 506, 806 n. 2  
   and database systems 664-5, 691, 745-6,  
     778, 803  
   due to software mechanism 32-3, 124, 195,  
     289, 292-3, 445-6, 469-71, 490, 611-12,  
     661-2  
   fostered by software companies 35-6,  
     446-7, 454, 805-6, 806 n. 2  
   fostered by the software elites 411, 414,  
     472-3, 604-5, 606-8, 610-12, 661-2  
   fostered by universities 39, 65, 273, 662, 780  
   GOTO prohibition perceived as answer to  
     604-6  
   and object-oriented programming 621-2,  
     661-2  
   and the relational model 671, 745-6, 750,  
     778, 803  
   and software magic 43-4, 54-5, 805, 807  
   software mechanism perceived as answer  
     to 501, 506-7, 604-5, 611-12, 661-2,  
     746, 750  
   and structured programming 502, 604-8,  
     611-12, 660-2  
   turned into management challenge 34-5,  
     484, 664-6
- programming languages  
   and formal grammars 259-61  
   fourth-generation, *see* fourth-generation  
     languages (4GL)  
   generations of 452-3, 489  
   hybrid, in object-oriented programming  
     642  
   levels of 452-3, 489  
   non-procedural, as low levels in high-level  
     systems 442, 795-6  
   in relational database systems  
     as means to override the relational  
     principles 792-4  
     *see also* SQL (database language)  
   traditional features of  
     as correct level of abstraction 451-3,  
     614-16, 641-2, 689-90, 691, 786  
     needed in database languages 439-41,  
     442-3, 662-3, 671, 689-90, 778-9,  
     786-7, 792  
     needed in development environments  
     435-7, 439-41, 451, 454-6, 615-16  
     needed in fourth-generation languages  
     (4GL) 439-41, 452-3  
     needed in object-oriented systems  
     615-16, 634-5, 641-2, 652

- programming languages (*cont.*)  
*see also* C (programming language);  
 COBOL (programming language)
- programming profession  
 vs. craftsmanship 483–5, 491  
 destruction of 27, 32–4, 37, 195, 198, 292–3,  
 472–3, 479, 662
- programming substitutes  
 as consumer products 38–9  
 database systems as 664–6, 745–6, 803  
 and the delusion of high levels  
 dishonest nature of 434–5, 438–42,  
 445–7, 450, 454–6  
 in object-oriented programming 614–16  
 promoted by the elites 288–90, 291, 853  
 in ready-made applications 456–7  
 in the relational model 662–3, 688–9,  
 786–7
- dependence on, *see* dependence: on  
 programming substitutes
- failure of 27, 144–5, 158, 196, 292–3, 807–8
- as mechanistic delusions 189–90, 300,  
 409–10, 413, 417, 425–6, 448–9, 450
- as mind models 190, 291–2
- perceived as magic systems 54–5, 805, 807  
 promoted by the elites 38–9, 145, 190,  
 438–42, 472–3, 517–19, 604–5, 636,  
 745–6
- replacing expertise  
 as mechanistic delusions 32–4, 144–5,  
 189–90, 192–3, 292–3, 585–6  
 and methodologies 461–2  
 and object-oriented programming  
 614–16, 626, 661–2  
 and primitive beliefs 54–5, 56, 65–6  
 and the relational model 663–4, 745–6,  
 747–8  
 and the software bureaucracy 195–7,  
 409–10, 413, 414  
 and structured programming 502,  
 506–7, 517–19, 553–4, 611–12
- programming theories, *see* software theories
- programs, structured 514, 523, 563, 573
- promotion, *see* advertising
- propaganda  
 mechanistic, *see* mechanistic propaganda  
 software, *see* software propaganda
- propositional calculus, as logic system 697
- pseudosciences  
 astrology as 13, 229  
 based on false assumption 203, 498  
 based on induction 209–10  
 behaviourism as 233, 236–7, 238–9, 277  
 causing totalitarianism 277, 832, 845–6  
 characteristics of 189, 201–3, 231–3, 497
- pseudosciences (*cont.*)  
 Chomskyan linguistics as 249–50, 257–8,  
 261–4, 271, 278  
 definition of 201, 497  
 examples of 201, 222, 224, 225–7, 233, 240,  
 249, 495–6  
 exposed through the principles of  
 demarcation 206, 228, 498  
 Freudianism as 208, 222  
 ignoring falsifications 219, 229, 237, 248,  
 497, 500, 613  
 as logical activities within a delusion 203,  
 495–6, 498–9, 500  
 magic as 44  
 making vague statements 215, 222, 239,  
 245–6  
 Marxism as 208, 224  
 the new, *see* new pseudosciences
- object-oriented programming as 635, 637,  
 641, 647, 653, 658
- perceiving human beings as deterministic  
 systems 179–81, 275–6, 431–2, 813–14,  
 819–20, 840–1
- perceiving societies as deterministic  
 systems 819–20, 840–1
- popularity of 227–8, 277
- relational model as 225–7, 662–3, 692, 711,  
 713, 770, 776–7, 781, 787
- resembling science 202, 203, 498–9
- software engineering as 478, 495, 613
- software theories as 29, 204, 207–8, 430–2,  
 441, 496–7, 500–1, 535, 612–13, 637, 660
- as spreading belief systems 52, 497–8
- structuralism as 240, 245, 278
- structured programming as 535, 538, 565,  
 576, 587, 612–13, 653, 658
- totalitarianism as 832, 845
- unfalsifiable nature of  
 and behaviourism 238–40  
 and Chomskyan linguistics 257–8,  
 263–4, 265, 267  
 as concept, explanation of 208, 215–16,  
 222–4, 498, 710–11  
 and the new pseudosciences 232–3, 342,  
 499, 713–14, 813  
 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 441, 612, 787  
 and software theories 500, 612  
 and structuralism 246–8

pseudosciences (*cont.*)

- unfalsifiable nature of (*cont.*)
  - and structured programming, *see* structured programming: expanded in order to suppress falsifications
  - using stratagems to avoid falsifications 222-4, 236-7, 246-8, 261-4, 342, 658, 840

psychology, and scientism 133, 137-8, 231, 233-4, 241

## Q

query systems, as low levels in high-level systems 444-5

## R

Ralston, Anthony

- and Böhm and Jacopini's work 562, 591
- and the first normal form 776
- and the GOTO debate 590, 591
- and non-standard constructs 567-8
- and the relational model 727
- and software engineering 494
- and structured programming 564-5

rationalism

- and hierarchical structures 304
- and the language fallacies 306-10, 311-16, 637-8
- and material world vs. mental world 304-6
- and mind models 278-80

rationality, blending with irrationality 12-13, 46, 49-51, 53-6

recognition, as non-mechanistic phenomenon 108-9, 117-18, 145-6, 159, 167, 181-2

record area, *see* file operations: record area in record keys, *see* keys

records

- in data files
  - definition of 674
  - duplicate 675
  - and the record area 677
- in the relational model, *see* relational model: records (rows, tuples) in

reduction

- of complex to simple structures
  - and application definitions 459-60
  - causing loss of alternatives 121-2, 123-4, 283-4, 287-8, 436-8, 815-17, 818-19, 853
  - and economics 847
  - and face recognition 108-9
  - and language 110, 266, 306-7, 319
  - as mechanistic delusion 102-3, 106-7, 205, 319, 415-16

reduction (*cont.*)

- of complex to simple structures (*cont.*)
  - and object-oriented programming 631-2
  - and reification 115-16, 117-18, 119, 120, 125, 498
  - and the relational model 710-11, 713
  - and structured programming 530-1, 536-8, 547-8, 567
  - and the three-body system 108
  - of *knowing how* to *knowing that* 147
  - to mechanics 74-5, 76, 77, 78, 132-8
  - of one discipline to another 77
  - of software elements to simpler ones 517, 519-20
  - of software to mechanics 28, 52-3, 72, 619 n. 12

reductionism

- as basis of understanding 80-1
- causing the mechanistic myth 82
- in corpuscular theories 76, 79
- definition of 9, 68
- failure of 77-8, 82
- formal 9, 76, 77-8
- in hierarchical structures 83-4
- informal 76
- and logical positivism 323-4
- modern 77-8
- partial 9, 77-8, 282-4, 321, 817

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

regularities, as basis of induction and causality 209-10

reification, fallacy of

- and abstraction 120-1
- in behaviourism 236
- causing loss of alternatives 118-19
- in Chomskyan linguistics 255, 256-7, 266
- in classification systems 313-14, 627, 631-2
- definition of 114
- in economics 847
- examples of 117-18, 255
- in human phenomena 126, 131, 283, 305-6, 817, 847
- and indeterminism 119
- in knowledge structures 119, 152-3, 299, 313-14, 369, 398-9, 416
- and mind models 152-3
- and the new pseudosciences 105, 125, 202-3, 205, 231-2, 498
- in object-oriented programming 628, 631-2, 638
- in the relational model 668, 710-11, 765-7, 786
- in structuralism 242

- reification, fallacy of (*cont.*)  
 in structured programming 530-1, 536-7, 567  
*see also* alternatives, loss of; mechanistic fallacies
- Reilly, Edwin D., *see* Ralston, Anthony
- relational model  
 as abstract concept 226, 709, 785  
 and abstraction 668, 669, 689, 692, 704, 709, 774, 786  
 and automatic application development 669  
 data redundancy and inconsistencies in  
 definition of 705, 729  
 due to incorrect relationships 738, 742-3, 748-9  
 examples of 705, 730  
 identical to traditional concepts 735, 737-8  
 impossible to formalize 735-6, 743, 747, 748-9, 750-1  
 misrepresented as difficult problem 735-6, 745, 750  
 and data validity  
 delusion of 707-8, 720-2  
 guarantee of 703, 719  
 and the delusion of high levels 669, 688-9, 691-2, 709-10, 722-4, 785-7  
 delusions of, summarized 712-13  
 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 802  
 expanded in order to suppress falsifications  
 by allowing low-level operations 711-12, 787, 790  
 by allowing non-1NF 772-3, 776-7, 779-80  
 by allowing programming 226-7, 671, 792-4  
 by allowing traditional database concepts 672, 692, 713, 769-70  
 by allowing unnormalized files 226-7, 711-12, 780-1  
 and extended relational databases 772, 777, 778-80  
 failure of 692, 711, 769, 781, 785, 790, 802  
 fallacies of  
 annulling the exact principles 711-12, 713, 734, 770, 779-80  
 annulling the mathematical benefits 726, 776, 777, 800-1, 802
- relational model (*cont.*)  
 fallacies of (*cont.*)  
 annulling the restriction to high-level operations 710, 711-12, 726, 769-70, 785-6, 790, 792-4, 799-800  
 annulling the restriction to normalized files 710, 711-12, 729, 771, 772-3, 780-2, 784-5  
 annulling the restriction to whole files 723-4, 726, 786, 799, 800  
 confusing cause and effect in database design 767  
 defining data as valid by default 700, 719, 720  
 destroying the integration of the database and the application 691-2, 786, 797, 801  
 expanding SQL into a programming language 794-6, 798, 799-800  
 ignoring the significance of the informal aspects 708, 720  
 misinterpreting the aspects of a database 766-7, 768-9  
 misrepresenting the concept of denormalization 781-3  
 misrepresenting the non-1NF feature 774-6, 777  
 perceiving denormalization as exact method 782-4  
 perceiving normalization as formal design principles 712, 728-9, 734-7, 747, 748-9, 750-1, 758-65, 766-7, 768-9, 783-4, 787-8  
 perceiving normalization as important theory 712, 734, 735-7, 750-3, 757, 758  
 perceiving software mechanism as substitute for expertise 668-9, 745-6, 747-8, 750, 778, 803  
 perceiving the integrity functions as high-level operations 790, 791  
 perceiving the mathematical aspects as important quality 666-7, 707-8, 712, 714, 716, 721-3, 724-5, 727, 745-6, 777-8, 779-80  
 recognizing only one type of field dependencies 755-7, 788  
 recognizing only the database structures 666-8, 719, 720-1, 750, 754  
 reinstating programming capabilities 721, 786, 790, 792-4, 796, 799-800, 801-3  
 reinstating the traditional database concepts 662-3, 672, 692, 711-12, 713, 770, 771, 774, 781, 800, 801-3

- relational model (*cont.*)
- fallacies of (*cont.*)
    - replacing design principles with correction principles 728–9, 732, 735–7, 745, 746–9, 758–65, 784
    - replacing design principles with dependency principles 705–6, 731, 734–5, 745, 746–9, 750–1, 766–9
  - restricting database operations to whole files 668, 703, 723–4, 726
  - reversing the principles of programming 615–16, 671
  - separating the integrity functions from the application 788, 790, 791
  - separating the operations from their implementation 225–6, 708–9, 784
- field dependencies in
- database design based on 747–9, 754–5, 758, 759–60, 761–3, 764–5, 768–9, 788
  - definition of 705, 730–1
  - examples of 705, 754, 788
  - identical to traditional concepts 706, 744–5
  - impossible to formalize 735–6, 748–9, 754–5, 758–65, 768–9
  - irrelevance of 733, 745
  - and normalization 705, 747–9, 751–3, 754–7, 758, 759–60, 761–3, 764–5, 788
  - as ordinary dependencies 756–7, 788
- fields (columns, attributes) in 700, 701, 701 n. 4, 704–5, 716, 719, 777
- file nesting in
- definition of 774
  - as falsification of the relational model 774, 775–6, 777, 780
  - as high-level operations 774
  - as informal aspect 777
  - promotion of 774–5, 778–80
  - as replacement of JOIN operations 774–6, 779
  - similar to traditional file nesting 774, 775–6, 779
- files (tables, relations) in 668, 700, 701, 701 n. 4, 703–5, 716, 719, 721–3, 777
- formal aspects of
- data format as 707, 721–2
  - first normal form as 729, 733–4, 770
  - vs. informal aspects 707–8, 710–11, 717, 724–5, 734–5, 751
  - manipulation as 717, 721–2, 724, 746, 751
  - query operations as 707, 720, 724, 726, 785
  - restricting the model's usefulness 725, 726, 746
  - founded on a convention 700, 719
- relational model (*cont.*)
- as fraud 441, 671, 672, 690–1, 714, 728, 777, 786, 800, 803
  - as high-level operations 664, 669, 688–9, 704, 709, 722–4, 776, 785, 786–7, 798–9
  - as illogical idea 671, 707, 708, 724–5, 726, 746
  - as inappropriate use of mathematics 707, 708, 719–20, 724–5
  - inefficiency of 709–10, 775, 778–9, 801–2
  - informal aspects of
    - data integrity as 708, 720, 724
    - database design as 746, 748–9, 750–1, 766, 767–9, 782–4
    - database language as 708, 724
    - field dependencies as 734–5, 748–9, 754–5
    - file nesting as 777
    - and the need for programming 725, 746
    - normalization as 707–8, 712, 724, 734, 747, 748–9, 750–1, 754–5, 758–65, 770, 787–8
    - replacing the formal aspects 711–12, 734, 770, 771, 779–80
    - translation as 717, 724, 746, 751
    - updating operations as 707, 724, 733–4, 785
  - integrity functions in
    - as built-in operations 791
    - and database languages 792–3
    - definition of 789
    - examples of 788, 789, 790–1
    - as extension to normalization 788
    - as fraud 790, 792–3
    - identical to traditional concepts 789, 791
    - as low-level operations 789–90, 791–2, 793
    - as part of the application's logic 789
    - and triggers 791, 793
    - as unnecessary feature 788, 790
    - and updating operations 787, 788
  - and interacting software structures 226, 667–8, 710, 720–1, 723–4, 747, 765–6
  - levels of abstraction in 722–4
  - as low levels in high-level systems 441, 662–3, 671, 690–1, 711–12, 784, 785–7, 790, 799–800
  - manipulation in
    - definition of 716–17
    - as formal aspect 717, 724
    - as minor aspect 718, 724, 746, 751
    - vs. translation 717, 724–5
  - mathematical claims of 225, 666–7, 668, 703–4, 707, 711, 714, 727, 747, 777, 802
  - as mechanistic delusion 666–7, 699, 713, 734–5, 765
  - motivation for 663–4, 672

- relational model (*cont.*)
- and nested relational databases 774, 777, 778–80
  - 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 699–700, 701–2, 707, 711, 715–16, 718, 732–3, 751, 777
  - and the problem of performance 709, 771, 775, 784
  - and programming incompetence 671, 745–6, 750, 778, 803
  - as programming substitute 664, 745–6, 747–8, 778, 803
  - promises of 225, 666, 668–9, 703–4, 745–6, 785, 802
  - promotion of 709, 727, 802–3
  - as pseudoscience 225–7, 662–3, 692, 711, 713, 770, 776–7, 781, 787
  - record keys in 700–1
  - records (rows, tuples) in 700, 701, 701 n. 4, 704–5, 716, 719, 721–3, 733, 777
  - and reification 668, 710–11, 765–7, 786
  - and similarity to the traditional data files
    - in file concepts 701, 706, 737–8
    - in file relationships 737–8, 743, 744–5, 748–9
  - as simple mathematical system 716, 717, 721–3, 724–5
  - SQL in, *see* SQL (database language)
  - stored procedures in
    - and database languages 794
    - definition of 794
    - as low-level operations 794
    - as means to bypass the relational principles 794
  - translation in
    - definition of 716–17
    - as informal aspect 717, 724
    - as main aspect 718, 724, 746, 751
    - vs. manipulation 717, 724–5
  - treating files as tables in 668, 703–4, 776, 785, 798–9
- relational operations
- benefits of 703
  - definition of 701–3
  - DIFFERENCE, definition of 702
  - and file nesting 774
  - and formal definitions 703
  - as high-level operations 664, 669, 688–9, 704, 709, 722–4, 776, 785, 786–7, 798–9
- relational operations (*cont.*)
- JOIN
    - definition of 702
    - file nesting as replacement of 774–6
    - inefficiency of 732, 771, 775, 778–9
    - uses of 702–3, 732, 771
  - as logical operations 701–2, 716
  - PRODUCT
    - definition of 702
    - uses of 702–3
  - PROJECTION
    - definition of 702
    - uses of 702–3
  - SELECTION
    - definition of 702
    - uses of 702–3
    - simplicity of 704
  - as subset of the basic file operations 777, 778 n. 9
  - treating files as tables 668, 703–4, 776, 785, 798–9
  - UNION, definition of 702
- relations
- in predicate calculus 697, 718
  - in the relational model, *see* relational model: files (tables, relations) in
- relations, between elements
- in software structures
    - reflecting jumps in execution 524, 529–30, 550–1, 553–4, 588–9
    - reflecting practices 346–7
    - reflecting shared data 350, 526–7, 552, 580–1
    - reflecting shared operations 344, 525–6, 552, 580–1, 653–4
    - reflecting the flow of execution 529–30, 532, 536–7, 544, 550–1, 552, 580–1, 583–4, 588
    - reflecting various processes 354, 536–7, 549, 641
    - see also* software processes
  - in structures 90–1, 96–7, 104, 110, 162, 623
  - see also* attributes
- relative files, definition of 674–5
- requirements, for software applications
- and automatic application development 465–6, 467–8, 522, 586, 669
  - perceived as exact specifications 291, 458–60, 463
  - reflected in the file relationships 742–4, 745, 747–9, 750, 754–5, 758–65, 766–7, 783–4
  - specified by selecting options 443–5
- research, charlatanism in, *see* corruption: in universities

- responsibility  
 degrading the concept of 820–2, 823–4, 831  
 denied by deterministic social theories 826–7  
 denied by software theories 828  
 and free will vs. determinism 809–10, 813–14, 819–20, 822–3, 826–9  
 and language 250, 824–5  
*see also* irresponsibility
- Ritchie, Dennis M., and the GOTO debate 596
- Rousseau, Jean-Jacques, and the idea of a general will 837
- rows, in the relational model, *see* relational model: records (rows, tuples) in
- rule-following, in novice performance 155–6, 157, 270
- Russell, Bertrand  
 and empiricism 211  
 and the humanist cause 319–20  
 and the idea of a perfect language 299, 325, 403  
 and mechanism 70  
 and the philosophy of language 319  
 and the problem of induction 210  
 and the rationalist philosophy 304  
 and the theory of logical atomism 319, 320–1
- Ryle, Gilbert, and mind models 146–8, 154
- ## S
- S-R units, *see* stimulus-response units
- sailing, expertise in, vs. magic 48–9, 50
- Salomon, Albert, and scientism 130
- Sampson, Geoffrey, and Chomsky's views on freedom and creativity 280–1
- scanning loop, *see* file scanning loops
- scanning sequence, in data files  
 ascending vs. descending 675  
 definition of 675, 676
- schizophrenia  
 as failure to combine knowledge structures 177–8, 183  
 manifestations of 177–8
- scholasticism, as scientific philosophy 12, 68, 70, 149
- science  
 based on conventions 89–90  
 based on induction 209–10  
 based on mechanism 69  
 based on tacit knowledge 161–2  
 of behaviour, behaviourism perceived as 233, 234  
 empirical  
 and mathematics 207  
 and the problem of induction 211
- science (*cont.*)  
 empirical (*cont.*)  
 vs. pseudoscience 207  
 vs. software theories 207–8  
 identified with mathematics 86–7  
 identified with mechanism 7, 10, 14, 86–7, 231–3  
 mechanism as method of 9, 69, 272–3, 499  
 of programming, software theories  
 perceived as 502, 509, 603–4  
 as pursuit of sound theories 204  
 similar to magic 46
- Scientific Revolution, and the mechanistic philosophy 15, 25, 68, 74–5, 82, 86, 128
- scientific theories, falsifiable nature of 208, 214–16
- scientism  
 in biology 134–5  
 causing the new pseudosciences 202  
 causing utopianism 20, 128, 836  
 definition of 127  
 due to mechanism 20, 125–6, 127–8, 132–6, 139, 159–60, 822–3, 836  
 in economics 131, 133–4, 847  
 and the idea of natural order 836  
 in linguistics 76, 110, 231, 241, 249, 824–5  
 and operationalism 137  
 in politics 135–6, 241, 822, 836  
 in psychology 133, 137–8, 231, 233–4, 241  
 in sociology 133, 137–8, 231, 241, 826–7, 836  
 views on 127
- scientists, as charlatans, *see* corruption: in universities
- selling, deception in 22, 831
- sentences  
 logical analysis of 317–18  
 meaning of 254, 257, 309–10, 317–18
- sequence  
 of execution, *see* flow of execution  
 of operations, static vs. dynamic 504, 529
- sequential constructs, *see* software constructs, flow-control: sequential
- sequential files, definition of 674–5
- service files  
 data files as, used to relate files 687, 739–40, 759, 763–4  
 index files as, *see* index files
- shared data, as software principle 351, 528–9
- shared operations, as software principle 353, 528–9, 653–4
- Shlaer, Sally  
 and mathematical programming 618–19  
 and object-oriented programming 619
- Simon, Herbert A., and artificial intelligence 143

- simple structures  
 alternatives in 91, 111–13, 281–4  
 attributes in 11–12, 93–4, 96–7, 98, 104, 360  
 definition of 88, 90  
 determinism of 97, 275–6  
 as mechanistic approximation 11, 104, 143,  
 146, 231–2, 236–7, 460, 536–8, 567, 812,  
 814–17  
 and the mind 162, 191–2, 414–16  
 as models, *see* mechanistic models  
 used for both analysis and synthesis 92–3  
*see also* hierarchical structures
- skills  
 blending with magic 47–9, 50–1  
 blending with software magic 51–2, 54–5  
 diverse, capacity for 168, 269–70, 414–15, 426  
 holistic knowledge in 155–8, 270  
 levels of 155–8, 414–15  
 natural 157–8, 164, 269  
 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 277  
 denying creativity and free will 277  
 and high-level stimulus-response units  
 239–40  
 and the Skinner box 235, 239
- slogans, used to impoverish knowledge 370–1,  
 383–4, 392, 393–4, 396, 849–50
- social forces  
 as deterministic explanation of historical  
 evolution 826–7  
 as idea, circularity of 826
- society  
 corrupted  
 by mechanism 431–2, 804, 845–7  
 by software mechanism 433, 493, 833–4  
 by software totalitarianism 409–11,  
 476–7, 833–4, 857–8  
 by totalitarianism 832, 851–2, 855, 857  
 inefficiency in 30, 38–9, 469, 476–7, 490  
 perceived as deterministic system  
 in the new pseudosciences 819–20, 840–1  
 under totalitarianism 838  
 perfect, *see* perfect society  
 primitive, *see* primitive society
- sociology  
 and determinism 826–7  
 and the idea of social forces 826–7  
 and scientism 133, 137–8, 231, 241, 826–7, 836
- software  
 controlled by the elites 407–8, 409–11, 419  
 deception through, *see* software charlatanism
- software (*cont.*)  
 dependence on 296–7, 407–8, 409–11, 431–2  
 easy to use  
 due to high levels 427, 438–40  
 as mechanistic delusion 426–8, 438–40  
 as environment, *see* development  
 environments; software devices  
 impoverishment of 124, 288–90, 365, 818,  
 853–4, 856–7  
 as interacting structures, *see* interactions,  
 of software structures  
 manipulated by the elites 368, 406–7, 856–7  
 as means to represent the world 254,  
 298–300, 343–4, 366–7, 406–7, 409–11,  
 528–9, 628–9, 638, 833, 854, 856  
 mechanistic, *see* mechanistic software  
 and the mechanistic fallacies, *see*  
 mechanistic fallacies: in software  
 as mechanistic myth 26, 28, 31, 33  
 as model of phenomena 103  
 as myth, *see* software myth  
 as non-mechanistic phenomenon  
 and the fallacy of software engineering  
 487–9, 501  
 and inadequacy of software devices 818,  
 853  
 and interacting structures 103, 123–4,  
 358–61, 362  
 and similarity to language 302–3, 324–5  
 and software theories 285, 583–5, 628–9  
 and Wittgenstein's later philosophy 343–5  
 perceived as mapping system 298, 302–3,  
 324–5, 327, 333, 343–4, 628, 638  
 perceived as mathematical concept 87,  
 487–8, 583–5, 586, 618–19, 711  
 perceived as products 42 n. 1, 852–4  
 perceived as supernatural phenomenon,  
*see* software magic; software power  
 potency of 31  
 promoted through confirmations rather  
 than falsifications 220, 396, 500, 613,  
 660  
 reducing to mechanics 28, 52–3, 72, 619 n. 12  
 similar to language  
 and Chomskyan linguistics 254, 261 n. 17  
 as means to represent the world 296–7,  
 298–300, 343–4, 345, 366–7, 626–7,  
 628, 638  
 and the mechanistic fallacies 290, 297–8,  
 324–5, 364–5, 368, 406–7, 419, 447–9,  
 692  
 and non-mechanistic knowledge 67,  
 368, 409–11, 419  
 perceived as mapping system 298–300,  
 302–3, 316, 324–5, 327, 333, 343–4, 448

software (*cont.*)similar to language (*cont.*)

- perceived as simple structures 298, 316, 342-3, 362, 628, 637-8
- and responsibility 825
- and the search for a perfect system 300, 324-5, 342-4
- and software totalitarianism 833, 854, 856, 858
- and Swift's satire 316, 447-50
- used to control minds 123, 366, 368, 406-8, 409-11
- and Wittgenstein's shift in philosophies 327, 342-4

## tested using the falsification principle 221

## turned into morbid preoccupation 860

## software applications

- aspects of 344, 346, 417, 434, 435-6, 438, 458, 507, 528, 614, 666-8
- created by their users 37-9, 473
- and databases, integration of 691, 786, 797
- and definitions, *see* definitions, of software applications
- and the delusion of life cycles 464-5
- and the delusion of perfect applications 463-4
- development of 456, 458-60
- failure of, *see* software failures
- as interacting structures
  - and application definitions 458-60
  - and the delusion of mathematical programming 87-8, 583-5, 586, 667-8
  - and the delusion of the main structure 343-4, 348, 361-2, 530-1
  - and file operations 688
  - and the need for low levels 435-8, 614-16, 634-5, 669, 688-90, 691, 710, 723-4, 786, 789-90
  - and object-oriented programming 628-9, 634-6, 638, 639-40, 645, 648
  - and the relational model 226, 710, 720-1, 723-4, 747
  - and similarity to language 362, 537-8, 614-15
  - and software processes 344, 345-6, 348, 350, 358-61, 363, 434, 528, 545-6, 688
  - and structured programming 507, 523, 553-4, 577-8, 579-83
  - and Wittgenstein's later philosophy 343-5, 358-61
- loss of alternatives in, *see* alternatives, loss of: in software applications
- and maintenance, *see* maintenance, of software applications

software applications (*cont.*)

- and methodologies, *see* methodologies
- perceived as simple structures, *see* software structures: perceived as simple structures

## ready-made

- and the delusion of high levels 456-7
- inadequacy of 456-8
- and the mechanistic fallacies 124
- and software totalitarianism 854
- and requirements, *see* requirements, for software applications
- and software innovations 458, 462

## software blocks

## definition of 503 n. 1

- as implicit jumps 506, 529-30
- as software elements 528, 540-1

## software bureaucracy

- aspects of 40-1, 412-13
- due to ignorance 29, 31-2, 419, 804-5
- due to programming incompetence 34-5, 37-8, 196-7, 409-10, 665-6
- due to software mechanism 29, 40-1, 198, 522, 662
- due to the elite 36-7, 662, 805-6
- spread of 34-5, 37-9, 476-7, 479, 833
- and structured programming 502, 565, 587-8, 611

## software bureaucrats

- managers as 34-5
- programmers as 32, 35-6, 412-13, 472, 479, 484-6, 611
- similar to primitives 805-6
- software users as 37-9, 473-5
- workers as 37-9, 473-5

## software charlatanism

- and the delusion of high levels 438-43, 454-6, 614-16, 636
- due to programming incompetence 37, 113-14, 434-5, 446-7, 803
- due to software mechanism 65-7, 72, 188, 272, 367-8, 406-7, 434, 450, 727-8

## software claims, assessing 46, 206

## software companies

- contributing to the software bureaucracy 35-6
- exploiting the mechanistic delusions 188, 417-18, 419, 425-6
- fostering programming incompetence 35-6, 446-7, 454, 805-6, 806 n. 2
- justified by the software myth 36, 853
- see also* software elites

## software constructs, flow-control

- and application maintenance 505-6, 514, 610

- software constructs, flow-control (*cont.*)
- built-in
    - vs. constructs using GOTO 550-1, 567-9, 577, 588-9, 597-9, 602-3, 604, 605
    - as language enhancements 597-9
    - perceived as substitute for expertise 605-7
  - CASE
    - definition of 567
    - as non-standard conditional construct 567, 575
  - conditional
    - definition of 503, 511
    - and the elimination of ELSE 601
    - equivalent to sequential constructs 542-3
    - non-standard 506-7, 524-5, 568-9
    - reducing to iterative constructs 558-60, 563
  - definition of 503, 511-12
  - elementary
    - and Böhm and Jacopini's work 558-61, 562-3
    - and transformations 558-61, 562-3
  - enhancing the nesting scheme 548, 551, 553
  - and file operations 676, 680-3, 684-5, 690, 691, 724, 799
  - and flow-control attributes 580-1
  - implicit jumps in 506, 515, 529-30, 539, 544, 550-1
  - and interacting structures 545-6, 579-82, 583-4, 652
  - iterative
    - definition of 503, 512
    - equivalent to sequential constructs 542-3
    - non-standard 506-7, 526, 567-9
  - language-specific, *see* software constructs, flow-control: non-standard, permission of
  - misinterpreted as operations between elements 534, 539-42, 546-7, 578-9, 599, 605
  - and multiple flow-control structures 534, 539-40, 542-5, 546-7, 576-7, 579, 583-4, 652
  - nesting of 511-14, 568, 683
  - and the nesting scheme, *see* structured programming: nesting scheme in non-standard, permission of
    - to avoid GOTO statements 567-9, 572, 575, 586, 588-9, 592, 597-9
    - to avoid transformations 566, 567-9, 575
    - justifying 567-8, 570-2, 575-6, 597-8
    - perceived as higher levels 578, 605
    - and similarity to standard constructs 576-7, 579, 588
  - software constructs, flow-control (*cont.*)
    - and object-oriented programming 649, 652
    - perceived as a minimal set of operations 546-7, 552
  - sequential
    - definition of 511
    - as operations in flow-control structures 540-1, 579
  - as source of programming errors 504-6
  - standard, restriction to
    - and absence of GOTO statements 515, 550-1
    - as basic principle 506-7, 511
    - impracticality of 552-3, 554-5, 557
    - inconvenience of 566-9, 572-4, 578
    - mathematical claims of 517-18
    - self-contradiction of 560-1
    - and transformations 523-7, 548-9, 588-9
  - software consumers, turning workers into 38-9
  - software crisis
    - aspects of 480, 618, 620
    - due to software mechanism 490, 621
    - due to the software theorists 490
    - and the idea of software reuse 626
    - origins of 490
    - and the prevention of expertise 501, 507, 604-5
    - spread of 490
  - software delusions, due to mechanistic delusions, *see* mechanistic delusions: causing software delusions
  - software devices
    - adequate for mechanistic problems 66, 193-5, 415-16, 417-18
    - based on mind models 188, 291-2, 818
    - degrading minds through 188, 409-11, 421-3, 427-9, 818-19, 820-1
    - dependence on, *see* dependence: on software devices
    - as frauds 65-7, 615-16
    - human beings perceived as operators of 472-3, 475, 476, 479, 486
    - inadequate for non-mechanistic problems 65-6, 193-5, 415-16, 417-18, 420-1
    - and irresponsibility 472-5, 825
    - as low levels in high-level systems
      - examples of 441
      - and fourth-generation languages (4GL) 441
      - inducing dependence 438-42, 451, 614-16, 691
      - and non-procedural languages 442, 795-6
      - and object-oriented programming 634-5, 636, 661-2

- software devices (*cont.*)
- as low levels in high-level systems (*cont.*)
    - and query systems 444-5
    - and the relational model 441, 662-3, 671, 690-1, 711-12, 784, 785-7, 790, 799-800
    - by selecting options 443-5
  - as mechanistic delusions 27, 188, 434-5, 614-16
  - vs. non-mechanistic capabilities 66-7, 186-7, 188, 194-5, 288-90, 415-16, 421-3, 431-2, 433, 434, 445-6, 819
  - replacing knowledge with selection of options 291, 294, 444-5, 449-50, 474, 475, 818
  - as substitutes for creativity 170, 280, 291-2, 294-5, 818-19
  - as substitutes for knowledge, *see* knowledge substitutes
  - as substitutes for programming expertise, *see* programming substitutes
  - vs. traditional devices 416
- software diagrams
- classification-style vs. flowcharts 362
  - see also* flowcharts
- software domination
- consuming people's time 419-20
  - degrading minds through 411, 425-6, 854
  - due to mechanistic delusions 31, 417, 446-7
  - fostering ignorance and dependence 419, 425-6, 438-42, 454-6, 461-2, 476-7, 479
  - through high-level systems 438-42
  - as new phenomenon 408
- software elites
- causing the software bureaucracy 36-7, 662, 805-6
  - as charlatans 65-7, 113-14, 289, 446-7, 510, 636, 660, 727-8, 803, 829
  - consuming people's time 419-20
  - controlling software 407-8, 409-11, 419
  - enforcing mechanistic knowledge, *see* non-mechanistic knowledge: prevention of
  - exploiting society 65-7, 113-14, 409-11, 446-7, 476-7
  - fostering dependence, *see* dependence: fostered by the software elites
  - fostering ignorance 65-7, 289, 420, 421-3, 425-6, 475-6, 479, 486, 805-6
  - fostering irresponsibility 606-7, 610-11, 805-6
  - fostering programming incompetence 411, 414, 472-3, 604-5, 606-8, 610-12, 661-2
  - justified by the software myth 26, 27, 31, 36-7, 833, 853
- software elites (*cont.*)
- manipulating software 368, 406-7, 856-7
  - 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, 145, 190, 438-42, 472-3, 517-19, 604-5, 636, 745-6
  - promoting software magic 43-4, 807
  - promoting software mechanism 29, 32-4, 36, 288-92, 293-4, 486-9, 517-19, 616-18, 728, 856-7
  - promoting totalitarian concepts 284, 824, 829, 833, 854-5, 857
- software engineering
- based on mechanism 28, 486-9, 493
  - definition of 480, 494
  - as idea, circularity of 481-3, 485, 491, 494-5
  - as pseudoscience 478, 495, 613
  - resembling astrology 496
  - resembling superstitions 495-6
  - and software magic 52-3, 54-5, 495 n. 4
  - transition to 480-1, 485, 491
  - as wish 482, 495
  - see also* software theories
- software evolution
- conforming to 828-9
  - explained as deterministic process 828-9
  - as justification for software devices 828-9
- software experts, *see* software theorists
- software exploitation, *see* software domination
- software failures
- due to irresponsibility 472, 807-8
  - due to programming incompetence 32-3, 469-71, 807-8
  - due to software mechanism 33, 286-8, 470-1, 490, 613
  - manifestations of 807, 808-9
  - rationalized through software mechanism 469-71, 472
  - suppressed through propaganda 808
- software forces
- as deterministic explanation of software evolution 828-9
  - as idea, circularity of 828-9
- software freedom
- and the delusion of high levels 438-42, 451, 454-6, 615-16, 691
  - and the fallacy of abstraction 95-6
  - and similarity to language 409-11
  - and the software myth 408

- software magic  
 blending with skills 51-2, 54-5  
 and the definition of software engineering  
 495 n. 4  
 emulating manufacturing concepts 52-3  
 as extension to skills 51-2, 54-5  
 and programming incompetence 43-4,  
 54-5, 805, 807  
 programming substitutes perceived as  
 54-5, 805, 807  
 promoted by the elites 43-4, 807  
*see also* magic
- software mechanism  
 causing charlatanism, *see* software  
 charlatanism  
 causing irrationality 31-2, 53-4, 600  
 causing irresponsibility 40-1, 804, 820-1  
 causing programming incompetence 32-3,  
 124, 195, 289, 292-3, 445-6, 469-71,  
 490, 611-12, 661-2  
 causing software failures 33, 286-8, 470-1,  
 490, 613  
 causing software-related problems 193-5,  
 418, 424, 469-71, 473-4  
 causing spurious preoccupations 193-5,  
 197, 418, 419, 424-5, 469-71  
 causing the software bureaucracy 29, 40-1,  
 198, 522, 662  
 causing the software crisis 490  
 causing totalitarianism, *see* totalitarianism:  
 due to software mechanism  
 causing utopianism 131, 860  
 as commitment 433  
 corrupting societies 433, 493, 833-4  
 cost of 33, 124, 446-7  
 degrading minds through 365, 409-11,  
 431-2, 856-7  
 degrading the concept of responsibility  
 821-2  
 as delusion 192-3, 467-71, 613  
 denying creativity and free will 290-2,  
 294-5, 432, 817-18, 828  
 as dogma 26, 489, 490  
 early successes of 486-7, 489  
 first stage in spread of 472-3, 476, 478  
 as ideology, conforming to 197, 409-11,  
 431-2, 476-7, 828-9, 854-5, 860-1  
 impoverishing knowledge 365, 833-4, 857  
 and mind control 365, 409-11, 431-2, 856-7  
 perceived as answer to programming  
 incompetence 501, 506-7, 604-5,  
 611-12, 661-2, 746, 750  
 preventing non-mechanistic knowledge  
 66-7, 368, 431-2, 474, 475, 490, 518-19,  
 580, 585-6
- software mechanism (*cont.*)  
 promoted by the elites 29, 32-4, 36, 288-92,  
 293-4, 486-9, 517-19, 616-18, 728,  
 856-7  
 proven invalid by the failure of CASE  
 467-9  
 second stage in spread of 473-5, 476, 478  
 similar to traditional mechanism 272, 500,  
 501  
 spread of 469-71, 472-7, 478  
 third stage in spread of 475-6, 478  
 as trap 66, 188, 417, 422-3, 446, 819, 861  
 used to rationalize failures 469-71, 472  
*see also* software theories
- software metrics, as mechanistic delusion  
 586 n. 19
- software modules  
 perceived as independent 353-4  
 as software elements 351-2, 528, 540-1  
 used as subroutines 353-4
- software myth  
 as basis of totalitarianism 31  
 definition of 26  
 due to ignorance 28, 31-2  
 fallacies of 27-8, 408  
 and irresponsibility 29, 408, 825  
 as justification for elites 26, 27, 31, 36-7,  
 833, 853  
 as justification for software companies 36,  
 853  
 and programming 27  
 promoted through propaganda 31, 39-40,  
 194, 413  
*see also* mechanistic myth
- software phenomena  
 alternatives in 286, 817-18, 853  
 as human phenomena 10, 87, 125, 287-8,  
 487  
 indeterministic nature of 87-8, 125, 285,  
 287-8, 817-18  
 perceived as mechanistic 817-18, 852-3  
 perceived as supernatural, *see* software  
 magic; software power
- software power  
 high levels perceived as 427, 438-40  
 knowledge substitutes perceived as 62, 805  
 as mechanistic delusion 426-8, 438-40  
 similar to mana 58-60, 61-2  
 as universal potency 62  
*see also* mana; power
- software practitioners, *see* programmers
- software principles  
 databases as 349-51  
 definition of 346, 528-9  
 flow-control as 529

- software principles (*cont.*)
- memory variables as 351
  - objects as, in object-oriented programming 653–4
  - practices as 346–8, 528–9
  - shared data as 351, 528–9
  - shared operations as 353, 528–9, 653–4
  - subroutines as 351–2, 353, 653
- software processes
- and attributes 346, 358–61, 528, 549–50
  - databases as 350, 359, 688
  - definition of 346, 528
  - and the delusion of high levels 191, 417, 436–8
  - external subroutines as 353
  - flow-control features as 529
  - and flow-control structures 545–6
  - and interacting structures 190–1, 344, 345–6, 348, 350, 358–61, 363, 434, 528, 545–6, 688
  - objects as, in object-oriented programming 653–4
  - practices as 346–8, 359
  - repeated operations as 353
  - and structures 346, 358–61, 528
  - subroutines as 351–2, 353, 359
- software propaganda
- and the media 220, 522–3, 613, 660, 808
  - promoting ignorance 66–7
  - promoting the software myth 31, 39–40, 194, 413
  - redefining the meaning of expertise 56, 413, 854
  - suppressing failures 808
- software-related affairs
- impoverishment of 113–14, 124, 289–90, 294
  - incompetence in 31–2, 41, 425–6
  - irresponsibility in 41, 820–1, 825, 828
  - non-mechanistic knowledge in 66–7, 191, 425–6, 466–8, 474, 475
- software-related names
- evoking electricity 60
  - evoking magic 42–3
  - evoking mana 59
  - and misuse of the term “technology” 390–1
- software-related problems
- due to software mechanism 193–5, 418, 424, 469–71, 473–4
  - spurious nature of 193–5, 197, 418, 419, 424, 469–71
- software structures
- and classification systems 347–8, 350, 352, 358–61, 528
  - databases as 350, 359, 688
  - flow-control, *see* flow-control structures
- software structures (*cont.*)
- interacting, *see* interactions, of software structures
  - perceived as simple structures
    - and the delusion of high levels 288
    - and the delusion of mathematical programming 87–8, 487–8, 517–18, 666–9
    - denying creativity and free will 290–2, 817–18
    - due to software mechanism 500–1, 517–18, 536–8, 566–7
    - emulating manufacturing concepts 27, 285, 287, 362, 547–8
    - in object-oriented programming 622–3, 628–9, 631–2, 638–40
    - in the relational model 666–8, 710–11, 713, 719, 723, 754, 765–6
    - and similarity to language 298, 300, 537–8, 628, 637–8
    - and software totalitarianism 852–3
    - in structured programming 511–14, 517–18, 519–21, 522, 533–4
    - and Wittgenstein’s early philosophy 333
    - and Wittgenstein’s later philosophy 343–5
  - practices as 346–7, 359–60
  - and software processes 346, 358–61, 528
  - subroutines as 351, 359–60
- software systems, honest vs. dishonest 450–2, 454–6
- 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 144–5, 190, 291–2, 820
  - and the delusion of the main structure 343–4, 351–2, 361–2, 536–8
  - and determinism 286, 290–2, 817–18, 827–9
  - emulating engineering concepts 84, 480–2, 489–90, 492–3, 494–5, 619
  - emulating manufacturing and construction concepts 27, 32, 34, 52, 72–3, 77, 287, 362, 618, 619, 621, 852–3
- failure of
- due to mind mechanism 144–5
  - due to software mechanism 12, 285–8, 467–70, 500–1, 516, 585–6, 612, 613, 621, 661–2, 667–8
  - as logical systems 207–8
  - reinstating the traditional concepts 501, 612, 662
- as frauds 29, 535, 637

- software theories (*cont.*)
- as informal concepts 492
  - as mechanistic delusions
    - and application definitions 458–60
    - and the delusion of high levels 434–5, 489, 612, 614–16, 661–2
    - and the delusion of mathematical programming 87–8, 487–8, 509–10, 517–18, 583–5, 585 n. 18, 586, 618–19, 666–9, 711, 717–23, 724–5, 727, 751–3, 777–8
    - perceiving software as simple structures 333, 430–2
    - pseudoscientific nature of 29, 500–1, 613, 635
    - see also* object-oriented programming: as mechanistic delusion; relational model: as mechanistic delusion; structured programming: as mechanistic delusion
  - and non-mechanistic phenomena 123–4, 192–3, 204, 853
  - as pseudosciences 29, 204, 207–8, 430–2, 441, 496–7, 500–1, 535, 612–13, 637, 660
  - reflected in software devices 291, 293–4
  - similar to deterministic social theories 827–9
  - similarity of 28, 72, 501
  - as substitutes for programming expertise, *see* programming substitutes
  - and Wittgenstein's early philosophy 327, 333, 342–3
  - see also* object-oriented programming; relational model; software mechanism; structured programming
- software theorists
- causing the software crisis 490
  - as charlatans 37, 145, 493, 535–6, 603, 637, 658, 727–8, 803
  - and circularity of software engineering concepts 481–3, 485, 491, 494–5
  - comparing programmers to craftsmen 483–5, 491, 621–2, 661
  - and the delusion of software mechanism 487–90, 515–16, 520–1, 565, 583–5, 636, 666–8
  - early contributions of 486–7, 493
  - lacking engineering knowledge 84, 492
  - lacking programming experience 486, 565
  - and methodologies 461
  - naivety of 502, 533, 563, 578, 603–4, 650, 667–8, 713, 734–5, 785
  - perceiving software as simple structures 84, 492–3, 500–1, 517–18, 536–8, 566–7, 666–8
  - see also* software elites
- software totalitarianism
- as conformism to the mechanistic software ideology 476–7, 854–5, 860–1
  - corrupting societies 409–11, 476–7, 833–4, 857–8
  - extending the scope of software 860
  - and Orwell's model 410–11, 834, 855–7, 858–61
  - similar to political totalitarianism 411, 476–7, 833, 834, 855, 857, 858–61
  - and Talmon's model 834, 852–5, 858–61
  - see also* totalitarianism: due to software mechanism
- software users
- creating their own applications 37–9, 473
  - dependent on software devices 56, 473–6
  - similar to primitives 65–7
  - as software bureaucrats 37–9, 473–5
  - spurious activities of 469, 471
- Sorokin, Pitirim A., and scientism 136–9
- sorting sequence, in data files, definition of 675
- space technology, analysis of term 395
- Spinoza, Baruch
- and hierarchical structures 304–5
  - and language structures 308
  - and material world vs. mental world 304–5
- SQL (database language)
- annulling the mathematical benefits 800–1
  - and the delusion of high levels 442–3, 795–6
  - embedded, definition of 800
  - expansion of
    - through embedded SQL 800
    - through explicit file scanning loops 798, 799, 800, 801–2
  - as falsification of the relational model 226, 786, 794–6, 799–800, 801–2
  - through general programming features 799–800, 801–2
  - through SELECT features 795, 798, 799, 800
  - through updating operations 796
  - and implicit file scanning loops 796–7, 798–9
  - as low levels in high-level systems 442–3, 795–6, 799
  - as means to suppress falsifications 226, 786, 794
  - as non-procedural language 442–3, 795–6
  - operations in
    - and the cursor 798, 799
    - DELETE 796–7, 798
    - FETCH 798, 800
    - INSERT 796–7
    - SELECT 794–5, 796–7, 796 n. 27, 798–9

- SQL (database language) (*cont.*)  
 operations in (*cont.*)  
 similar to the traditional file operations  
 797, 798, 799, 800, 801–2  
 UPDATE 796–7, 798  
 as query language 794, 799  
 standard constructs, *see* software constructs,  
 flow-control: standard, restriction to  
 standards, for programming expertise 34,  
 195, 607, 808, 808 n. 3  
 statements, in programming languages,  
 definition of 503 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 302  
 and modern myths 4–5
- stimulus-response units  
 definition of 234  
 as elements of structure 236–7  
 failure to reduce behaviour to 238–40  
 modified to explain complex acts 239–40
- stories  
 aspects of 349, 355, 399  
 as interacting structures 348–9, 355–7, 399  
 and non-mechanistic knowledge 355–7
- structuralism  
 analysis of art and literature in 243  
 analysis of myths in 242–3  
 based on mechanism 241  
 binary operations in 242–5, 246  
 computer analogies in 244  
 definition of 240–1  
 denying creativity 278  
 failure of 248  
 mathematical pretences of 244, 247, 248  
 as myth 4–5  
 as pseudoscience 240, 245, 278  
 and reification 242  
 and simple structures 275  
 stemming from language mechanism 242  
 transformations in 243–4, 246, 247–8
- structured design, *see* structured programming:  
 top-down design in
- structured programming  
 and abstraction 511, 517, 578, 599  
 black-box principle in  
 definition of 573 n. 14  
 as degradation of the original principles  
 570, 573–4, 575–6, 659  
 inappropriate use of 573 n. 14  
 as justification of non-standard  
 constructs 570, 573–4, 577
- structured programming (*cont.*)  
 black-box principle in (*cont.*)  
 and the nesting scheme 512–14, 573–4,  
 577  
 and Böhm and Jacopini's work, *see* Böhm  
 and Jacopini, work of  
 and CASE 521–2, 586  
 causing the software bureaucracy 502, 565,  
 587–8, 611  
 and the delusion of the main structure  
 361–2, 515, 530–1, 536–8  
 delusions of, summarized 534–5  
 dominating programming practices 502,  
 565  
 emulating manufacturing concepts 517,  
 531–2, 547–8  
 expanded in order to suppress falsifications  
 by allowing GOTO statements 569, 589,  
 594–6  
 by allowing informal principles 564–5,  
 658  
 by allowing multiple flow-control  
 structures 538–9, 653  
 by allowing non-standard constructs  
 567–9, 570–2, 576, 587  
 by allowing transformations 516
- failure of  
 denied by the software theorists 509–10,  
 516, 538, 565, 566–7, 570–2, 585, 587,  
 589 n. 1, 602, 612–13  
 and the failure of object-oriented  
 programming 653, 658, 661–2
- fallacies of  
 attempting to represent applications  
 mathematically 583–5, 585 n. 18, 586  
 blaming GOTO for bad programming  
 506–7, 508–9, 606  
 ignoring all systems of jumps and labels  
 607, 610–11  
 increasing the complexity of software  
 549, 554, 580–3  
 misinterpreting the inconvenience of  
 the transformations 566–7, 570–3,  
 578–9  
 misinterpreting the nesting scheme 532,  
 533–4, 536, 538, 577–8  
 misinterpreting the role of flow-control  
 constructs 534, 539–42, 546–7, 578–9,  
 599, 605  
 misrepresenting Böhm and Jacopini's  
 work 509, 552–3, 554–5, 557–63, 570–1,  
 591  
 perceiving built-in constructs as  
 substitute for expertise 506–7, 553–4,  
 605–7

- structured programming (*cont.*)
- fallacies of (*cont.*)
    - perceiving software mechanism as substitute for expertise 502, 517–19, 582–3, 585–6, 612, 661–2
    - promising error-free programming 520–1
    - recognizing only one flow-control structure 520–1, 545, 547, 553–4, 567, 577, 578, 649
    - recognizing only the flow of execution 507, 536–8, 567, 649
    - replacing the original principles with the black-box principle 570, 573–4, 575–6, 659
    - replacing the original principles with the GOTO preoccupation 589, 591, 592, 593, 594–6, 597–9, 602–3, 604–5
    - replacing the original promises with trivial concepts 564–5, 590–1, 592–3, 658
  - flow-control constructs in, *see* software constructs, flow-control
  - flow of execution in
    - as complex structure 542–4, 553–4, 576
    - and flow-control attributes 529–30, 550–1, 579–80
    - and the GOTO debate 588–9, 598–9, 602, 605
    - and the nesting scheme 514–15, 531–2, 536, 577–8
    - perceived as simple structure 523–4, 539–42, 649
    - see also* flow-control attributes; flow-control structures; software constructs, flow-control
  - formal theory of 511–14, 563, 564, 591, 602
  - as fraud 535, 563, 578, 587
  - GOTO statement in
    - as aspect of programming style 589, 602, 604, 607
    - and the attempt to replace it with other jumps 592, 593–4, 597–9
    - blamed for bad programming 506–7, 508–9, 606
    - vs. built-in constructs 550–1, 567–9, 577, 588–9, 597–9, 602–3, 604, 605
    - and the contradictory claims 593, 594–6
    - definition of 504
    - and flow-control attributes 529
    - formal arguments against 515, 605
    - informal arguments against 508–9, 589–91, 594–6, 597–9, 605–6
    - and the informal theory vs. formal theory 602–3
  - structured programming (*cont.*)
    - GOTO statement in (*cont.*)
      - as means to control programming practices 607–8, 611
      - needed for explicit jumps 504, 524–7, 529, 588–9
      - as ongoing debate 572, 583, 586–7, 592
      - permission of 569, 589, 594–6, 600
      - preoccupation with, as substitute for the original principles 589, 591, 592, 593, 602–3, 604–5
      - prohibition of 524–7, 563, 588–91, 597–9, 603, 604–6
      - as superstition 493, 502, 600, 602
      - and systems of jumps and labels 608–11
      - and the use of humour 600–1
    - and hierarchical structures 511–14, 533–4, 536–8, 547–8, 566–7, 604
    - informal theory of, vs. formal theory 508–10, 563, 564, 602–3
    - and interacting software structures 507, 523, 553–4, 577–8, 579–83
    - as mass madness 587, 598–9
    - mathematical claims of 509–10, 514, 517–18, 520, 551, 553, 573, 579–80, 583–5, 586
    - as means to control programming practices 34–5, 35 n. 2, 607–8, 611–12, 661–2
    - as mechanistic delusion 515–16, 530–3, 536–8, 547–8, 553–4, 566–7, 576–7, 585–6, 603, 653
    - and misuse of the term “structured” 593
    - and modularity 514, 519–20
    - nesting scheme in
      - and the black-box principle 512–14, 573–4, 577
      - concept of, as convention 531, 544
      - and flow-control attributes 531, 544, 546–7, 550–1, 580
      - and nesting of constructs 503, 511–14
      - and nesting of modules vs. nesting of constructs 514
      - overridden by jumps 533–4, 539, 544, 550–1, 576–7, 579, 588
      - reflected in the flow diagram 512–14, 531, 544, 577, 649
      - static vs. dynamic 544–5, 649
      - and top-down design 512–14
    - non-standard constructs in, *see* software constructs, flow-control: non-standard, permission of
    - perceived as a new science 509, 563, 603–4
    - perceived as a revolution 501, 522–3
    - as programming discipline 517–18, 523–4
    - and programming incompetence 502, 604–8, 611–12, 660–2

- structured programming (*cont.*)  
 promises of 507, 515, 517–18, 519–22  
 as pseudoscience 535, 538, 565, 576, 587,  
 612–13, 653, 658  
 and reification 530–1, 536–7, 567  
 requiring non-mechanistic knowledge  
 579–83, 662  
 and restriction to mechanistic knowledge  
 518–19, 579–80, 585–6  
 standard constructs in, *see* software  
 constructs, flow-control: standard,  
 restriction to  
 and the structure theorem 509  
 and structured programs 514, 523, 563, 573  
 and structures, static vs. dynamic 531–3,  
 544–5, 577, 583–4, 649–50  
 top-down design in  
 benefits of 351, 514, 519–20, 523–4  
 definition of 512–13  
 practice of 512–14, 517, 519–20  
 transformations in  
 to avoid GOTO statements 525–7, 551,  
 552, 586  
 and elementary constructs 558–61,  
 562–3  
 as falsification of the theory 516, 525–7,  
 549  
 and flow-control structures 548–9,  
 551–3, 580–1, 588–9  
 futility of 553–4  
 impracticality of 552–3, 554–5, 557,  
 580–2, 583  
 inconvenience of 566–9, 572–4, 578,  
 580–3, 585  
 as mechanistic fantasy 557  
 misrepresenting Böhm and Jacopini's  
 work 552–3, 554–5, 557–63  
 motivation for 549–51  
 replacing flow-control attributes 547,  
 551–2, 580–1  
 by sharing data 526–7, 551–2, 581  
 by sharing operations 525–6, 551
- structures  
 complex, *see* complex structures  
 hierarchical, *see* hierarchical structures;  
 simple structures  
 imaginary 115–16, 117–18, 119, 120, 355, 360  
 interacting, *see* interactions, of language  
 structures; interactions, of software  
 structures; interactions, of structures  
 in language, *see* language structures  
 real 115–16, 117–18, 119, 120  
 simple, *see* hierarchical structures; simple  
 structures  
 in software, *see* software structures
- structures (*cont.*)  
 static vs. dynamic 531–3, 544–5, 545 n. 1,  
 577, 583–4, 649–50  
 systems of, *see* complex structures;  
 interactions, of structures
- subroutines  
 external  
 as high-level operations 670  
 as software processes 353  
 as high-level elements 452  
 as software principle 351–2, 353, 653  
 as software processes 351–2, 353, 359  
 as structures 351, 359–60  
 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 209–10, 218–19  
 and the fallacy of software engineering  
 495–6  
 and the GOTO debate 493, 502, 600, 602  
 and the mechanistic myth 1, 493  
 and the principles of demarcation 219  
 resembling rational pursuits 13, 495–6  
 as systems of belief 1, 43, 493, 495–6, 600,  
 805
- Swanson, Guy E., and mana 57, 59, 62
- Swift, Jonathan, and language mechanism  
 315–16, 447–9
- syllogistics, as logic system  
 description of 693–4  
 examples of 693
- symbolic logic, definition of 693
- synthesis, in simple structures 92–3
- 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 161–2  
 as basis of non-mechanistic mind models  
 168–9  
 as basis of science 161–2  
 and complex structures 158–62  
 definition of 158–9  
 examples of 159–62, 163–4, 165–7, 269–71

- tacit knowledge (*cont.*)  
 and intuitive knowledge 169  
 as normal human capability 167–8  
*see also* non-mechanistic knowledge
- Talmon, J. L.  
 and liberal democracy vs. totalitarian  
 democracy 834–5, 859  
 and the paradox of freedom 838–9  
 and scientism 136, 836  
 totalitarian model of, *see* totalitarianism:  
 Talmon's model of
- Taylor, David A.  
 and the industrialization of software 620–1  
 and object-oriented programming 616,  
 617, 620–1, 660
- technology  
 as abstract term 384  
 definition of term 384  
 misuse of term  
 analysis of 386–7, 388–9, 392  
 in company names 390–1  
 deception through 388–9, 392  
 and dictionaries 392  
 examples of 384–8, 389, 390–1, 659 n. 2,  
 778 n. 11  
 in general discourse 391  
 grammatical aspects of 388  
 inflation in 384, 386–7, 619–20  
 logical aspects of 388, 392  
 as plural 390  
*see also* information technology: misuse  
 of term  
 term used as slogan 370, 383–4, 387, 392,  
 393–4, 395, 619–20
- testability, as quality of scientific theories  
 214–16
- tests  
 through falsifications rather than  
 confirmations 214–16, 217, 220–1  
 genuine, definition of 217
- theories  
 based on false assumption 203  
 based on particles 74–5  
 corpuscular 74–5, 78–80  
 degraded by adding features 223–5, 226–7,  
 229–30, 576, 647, 658, 671, 769–70  
 doubting vs. defending 206–7, 224–5, 267,  
 326 n. 1, 497, 500  
 expanding in order to avoid refutation, *see*  
 falsifications: suppressed by  
 expanding the theory  
 falsifiable, *see* falsifiability  
 identifying as mechanistic 82–3  
 mechanistic, *see* mechanistic theories  
 as models 89–90, 104–5, 207
- theories (*cont.*)  
 old, and the falsification principle 220  
 pseudoscientific, unfalsifiable nature of,  
*see* pseudosciences: unfalsifiable  
 nature of  
 scientific, falsifiable nature of 208, 214–16  
 in software, *see* software theories  
 as systems based on conventions 89–90  
 as systems that permit or forbid events  
 215–16, 215 n. 17, 224  
 unfalsifiable, *see* falsifiability
- theorists, *see* software theorists
- three-body system, as complex structure  
 107–8, 117
- top-down design, *see* structured programming:  
 top-down design in
- totalitarian democracy  
 definition of 835, 859  
 similar to academic ideologies 835–6  
 similar to business ideologies 835–6  
 and totalitarian software concepts 859–61  
*see also* totalitarianism
- totalitarianism  
 based on democratic ideas 834–5  
 based on the idea of natural order 837  
 and central economic planning 846–8  
 characteristics of 16–17, 128–30, 131–2, 840,  
 851, 855  
 as conformism to a mechanistic ideology  
 839–41, 843–4, 848, 854–5, 859  
 corrupting societies 832, 851–2, 855, 857  
 degrading language 122–3, 403–5, 849, 855–6  
 due to mechanism  
 based on mechanistic theories 16, 20,  
 200, 838  
 based on the idea of a perfect society  
 16–17, 131–2, 822, 823–4  
 promoted by corporations 829–30, 831  
 promoted by universities 274, 277,  
 829–30, 831  
 due to pseudosciences 277, 832, 845–6  
 due to software mechanism  
 based on mechanistic theories 131, 274,  
 852–3  
 based on the idea of a perfect society  
 131, 824  
 as conformism to a mechanistic ideology  
 852–5  
 by controlling minds through software  
 409–11, 476–7, 857  
 promoted by the elites 284, 824, 829,  
 833, 854–5, 857  
*see also* software totalitarianism  
 due to utopianism 20, 128–30, 131–2, 822,  
 831, 836–7, 845, 858–9

- totalitarianism (*cont.*)  
 embraced by intellectuals 845  
 as enactment of a myth 30  
 extending the scope of politics 860  
 fragmentary, through many elites 833  
 as idea, circularity of 837–8, 839  
 inefficiency of 30  
 justification of 19–20, 131–2, 837, 838,  
 839–40, 847–8  
 as means to create a perfect society 16–17,  
 18–19, 131–2, 822  
 as mechanistic delusion 21, 24, 274, 838,  
 845–6  
 non-political, similar to political 411, 832–3,  
 834, 851–2, 855, 857, 858–61  
 as non-political ideology 832–3, 851–2, 855,  
 857–61  
 objections to 19–20  
 Orwell's model of  
 and the academic elites 845–6  
 and central economic planning 846–8  
 and mechanistic language 396, 403–5,  
 848–50  
 and software totalitarianism 410–11,  
 834, 855–7, 858–61  
 perceiving human beings as deterministic  
 systems 822, 838, 847–8  
 perceiving societies as deterministic  
 systems 838  
 progression toward 30, 40, 397, 476–7,  
 829–30, 832, 844–5, 849, 851–2  
 promoted by corporations 829–30, 832,  
 839, 843–4  
 promoted by universities 274, 277, 823–4,  
 829–30, 832, 839, 840–1, 845  
 as pseudoscience 832, 845  
 replacing individuality with conformism  
 839, 843–4, 848  
 role of the bureaucracy under 30, 476–7,  
 830, 833  
 role of the elite under 30, 476–7  
 through software, *see* software  
 totalitarianism; totalitarianism:  
 due to software mechanism  
 and the software myth 31  
 Talmon's model of  
 and the academic elites 840–1  
 and the corporate elites 842–4  
 and the origins of totalitarianism 834–7  
 and software totalitarianism 834, 852–5,  
 858–61  
 transformational grammar  
 definition of 249  
 as extension of simpler grammars 260–1  
 transformations in 258–9, 260–1  
 transformational grammar (*cont.*)  
*see also* Chomskyan linguistics  
 transformations, in structuralism 243–4, 246,  
 247–8  
 transformations, in structured programming,  
*see* structured programming:  
 transformations in  
 transformations, in transformational  
 grammar 258–9, 260–1  
 transition, between myths 15, 24–6  
 trap, mechanistic 66, 128, 188, 417, 422–3, 446,  
 819, 861  
 truth functions  
 definition of 330, 696  
 in Wittgenstein's early philosophy 330  
 truth tables, description of 330, 696  
 truth values, definition of 330, 693, 695  
 tuples  
 in predicate calculus 697, 718, 732–3  
 in the relational model, *see* relational model:  
 records (rows, tuples) in  
 Turing machines  
 and Böhm and Jacopini's work 557–8, 559  
 description of 555–6  
 as mechanistic models 555–6  
 as theoretical devices 555–6, 557–8  
 two-body system, as simple structure 107–8, 117
- ## U
- understanding, mechanistic, *see* mechanistic  
 explanation  
 unity of science, and logical positivism 323  
 universal characteristic, as mathematical  
 language 313  
 universal grammar  
 definition of 252  
*see also* Chomskyan linguistics  
 universality, of scientific theories 89–90  
 universities  
 as bureaucracy 14, 16, 273, 841  
 corruption in, *see* corruption: in universities  
 as elite 830, 832, 835–6, 840–1, 845–6  
 fostering programming incompetence 39,  
 65, 273, 662, 780  
 promoting mechanistic concepts, *see*  
 mechanism: promoted by universities  
 promoting totalitarian concepts 274, 277,  
 823–4, 829–30, 832, 839, 840–1, 845  
 unspecifiable knowledge, *see* non-mechanistic  
 knowledge; tacit knowledge  
 utopianism  
 causing totalitarianism 20, 128–30, 131–2,  
 822, 831, 836–7, 845, 858–9  
 due to mechanism 131, 831, 836, 858

- utopianism (*cont.*)  
 due to scientism 20, 128, 836  
 due to software mechanism 131, 860

## V

- variables, as software principle 351  
 verifiability  
 vs. falsifiability 212, 214–15, 322  
 and logical positivism 322  
 and Wittgenstein's early philosophy 332  
 visual perception  
 as acquired skill 158  
 as non-mechanistic phenomenon 148–9  
 von Halle, Barbara, *see* Fleming, Candace C.

## W

- Westfall, Richard S.  
 and mechanism 80  
 and scientism 134–5  
 wholes  
 as justification of irresponsibility 827, 828  
 as unexplained social forces 826–7  
 as unexplained software forces 827–8  
 Whyte, Lancelot L., and atomism 79, 82  
 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 319, 326–7,  
 334–5  
 and the shift in philosophies 326–7, 334–5,  
 341–2  
 Wittgenstein's early philosophy  
 as abstract system 328–9, 332  
 description of 328–31  
 limitations of 332  
 as mapping system 326, 328–31  
 and meaningfulness 331–2  
 popularity of 326–7, 332, 334  
 and software theories 327, 333, 342–3  
 structures and elements in 328–31  
 and truth functions 330  
 and verifiability 332  
 Wittgenstein's later philosophy  
 and classification systems 336–9, 340–1,  
 354–7, 358–61

- Wittgenstein's later philosophy (*cont.*)  
 and complex structures 334–41, 343–5  
 and the idea of family resemblance 335–6,  
 340–1  
 and the idea of games 335–9, 340–1, 345  
 and the idea of language games 340, 341  
 misinterpreted by mechanists 327  
 as non-mechanistic system 326, 334, 341–2  
 as repudiation of mechanism 326, 342  
 and software applications 343–5, 358–61  
 and word meanings 334–5, 341  
 words  
 and deception through language 372–3,  
 374–80  
 meaning of 12, 110, 254, 257, 309–10, 314,  
 348–9, 399, 627  
 in Wittgenstein's later philosophy 334–5, 341  
 workers  
 deskilling of 198, 293–4  
 as software bureaucrats 37–9, 473–5  
 turned into software consumers 38–9  
 world  
 explained through empiricism 207–8, 214  
 identified with mathematics 86–7, 299–300,  
 307–10  
 material vs. mental, in rationalism 304–6  
 perceived as logically perfect 299–300,  
 317–18, 342–3  
 perceived as simple 79–80, 81, 89–90  
 represented with a perfect language  
 299–300, 301–2, 317–19, 323–4, 328–31,  
 626–7  
 represented with language 297–8, 366–7,  
 406–7, 856  
 represented with software 254, 300, 343–4,  
 366–7, 406–7, 409–11, 528–9, 628–9,  
 638, 833, 854, 856

## X

- X-ray pictures, learning to interpret 165

## Y

- Y2K (year 2000), software problems of 198  
 Yourdon, Edward  
 and Böhm and Jacopini's work 562  
 and the GOTO debate 589–90, 592, 600  
 and non-standard constructs 570

