

Donald
Kossmann

A stylized white hand with black outlines is shown holding a black wand with a silver-colored tip. The wand is positioned diagonally across the cover. At the tip of the wand, there is a black starburst or spark effect. The background is a solid teal color with several overlapping, semi-transparent light blue rounded squares of various sizes scattered across it.

THE MAGIC OF COMPUTER SCIENCE

v/d/f

Weitere aktuelle vdf-Publikationen
finden Sie in unserem **Webshop**:

vdf.ch

- › Bauwesen
- › Naturwissenschaften,
Umwelt und Technik
- › Informatik, Wirtschafts-
informatik und Mathematik
- › Wirtschaft
- › Geistes- und Sozialwissen-
schaften, Interdisziplinäres,
Militärwissenschaft,
Politik, Recht

Gerne informieren wir Sie regelmässig per
E-Mail über unsere Neuerscheinungen.

Newsletter abonnieren

[Anmeldung auf vdf.ch](https://www.vdf.ch)

For Elli

**In loving memory of my mother,
Judith Kossmann (1935–2020)**

Donald Kossmann

THE MAGIC OF COMPUTER SCIENCE

Translated from German
by Judith Kossmann

v/d/f

Any opinions expressed in this book are the author's own and may not represent those of Microsoft.

Bibliographic information published by the Deutsche Nationalbibliothek

The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at <http://dnb.dnb.de>.

ISBN 978-3-7281-4055-5 (Printversion)

ISBN 978-3-7281-4056-2 (E-Book)

ISBN 978-3-7281-4057-9 (E-Pub)

DOI-Nr. 10.3218/4056-2

verlag@vdf.ethz.ch

www.vdf.ethz.ch

© 2021, vdf Hochschulverlag AG an der ETH Zürich

Cover design: Isabel Thalmann, buchundgrafik.ch

All rights reserved. Nothing from this publication may be reproduced, stored in computerised systems or published in any form or in any manner, including electronic, mechanical, reprographic or photographic, without prior written permission from the publisher.

Content

1 Foreword.....	7
1.1 About Me	7
1.2 About This Book.....	9
1.3 Overview	13
2 What Is Computer Science?.....	15
2.1 Summary	15
2.2 What Does Computer Science Want to Be?.....	16
2.2.1 The Natural Sciences.....	17
2.2.2 The Engineering Sciences	21
2.2.3 The Philosophical Sciences.....	24
2.2.4 Computer Science.....	26
2.3 What Does Computer Science Do?	29
2.3.1 Computer Science as an Engineering Science	29
2.3.2 How Computer Science Changes the World?.....	33
2.4 The Job Description of a Computer Scientist.....	36
2.4.1 Why Is There No Good Job Description Out There?	36
2.4.2 Vertical vs. Horizontal Computer Science	36
2.5 The History of Computer Science	39
2.5.1 The Five Eras of the IT Industry.....	39
2.5.2 Speculation About the Future	45
3 Classic Computer Science.....	47
3.1 Summary	47
3.2 Areas of Classic Computer Science.....	48
3.3 Elements of Classic Computer Science	49
3.4 Architecture.....	52
3.4.1 Models	52
3.4.2 Modeling.....	55
3.4.3 Why Do So Many Software Projects Fail?.....	56
3.4.4 KISS: Keep it Simple, Stupid	57
3.4.5 Teamwork	59
3.5 Behavior: What Is a Function?	61
3.5.1 Imperative Programming.....	61
3.5.2 Correctness and Completeness.....	66

Content

3.5.3 Complexity (Efficiency).....	69
3.5.4 Flexibility: High-Level Programming Languages.....	72
3.6 Customization and Configuration	75
4 Artificial Intelligence (AI).....	77
4.1 Summary	77
4.2 The Dream of Artificial General Intelligence	78
4.3 What Is Different?.....	80
4.3.1 Big Data, Data Science, Machine Learning.....	80
4.3.2 Data-Driven Artificial Intelligence.....	85
4.4 Logic vs. Experience	88
4.4.1 Experience Usually Wins.....	89
4.4.2 Limitations of Experience	92
4.4.3 Reinforcement Learning: Deep Blue, Jeopardy, AlphaGo, Ms. Pacman	95
4.5 Deep Neural Networks (DNN)	97
4.5.1 Principles of Neural Networks.....	97
4.5.2 Neural Networks as a Programming Language	101
4.6 Software 2.0	105
4.6.1 Limitations of Traditional Programming (Software 1.0)	105
4.6.2 Machine Teaching.....	111
4.6.3 Classic Computer Science and Software 2.0.....	113
4.7 Security, Privacy, and Integrity	115
4.7.1 Confidentiality and Privacy.....	116
4.7.2 Integrity	119
4.7.3 Availability	121
The End.....	123
Acknowledgments	125

Chapter 1

1 Foreword

1.1 About Me

I am a father, a husband, and a computer scientist. But before I was any of these, I was a confused teenager asking himself what he wanted to do with his life. Wise people gave me good advice: “Fulfill your passion.” “Think about what you can do well and what may be needed in the future.” “Be persistent and work hard.” And, of course, the best advice of all: “No matter what you do, always be conscious of why you are doing it.”

All that was good advice, but it didn’t help me much. I had no particular passion. I was pretty good at math but not good enough to stand out. Furthermore, I had no idea of what use math could be. I was certainly not very persistent. I was unsure and easily influenced, and I had no idea what “hard work” meant. I didn’t even know what I should do, much less be aware of *why* I was doing it. More or less by chance, I ended up studying computer science at the University of Karlsruhe in Germany. I must thank my brother, Daniel, for this choice. Daniel took a course in computer science at the Harvard Summer School (Cambridge, MA) when he was 18 which infected him with the computer-science virus. I was his obedient pupil, helping him with diverse, small programming projects after he returned from Harvard and while I was still in high school.

The virus had not yet infected me, but I wanted to please my brother.

Four years later, I took the same computer-science course at Harvard and enjoyed it. It made sense to consider computer science as my field of study. In the end, however, it was the lack of alternatives that led me to computer science. I could not have been luckier.

In the 30 years that followed, I had a remarkable career as a computer scientist. After receiving my diploma from the University of Karlsruhe, I earned my doctorate at the Technical University of Aachen. I completed my postdoctorate studies at the University of Maryland. After a few additional hops up the academic ladder, I ended up as a Professor of Computer Science at ETH Zurich, one of the most prestigious universities in the world. This journey was linear: I never had an alternative.

1.1 About Me

I never made a conscious decision. Had I had to make a decision, it would have been by chance. I never earnestly thought much about why I chose this path.

During my first few years at ETH Zurich, I was a passionate teacher. My approach was to explain the concepts of computer science using everyday examples. I used a wedding, for example, to explain the “all or nothing principle” (called *atomicity* in computer science). The groom must marry the bride *and* the bride must marry the groom – both or none. In fact, computer science is common sense: It contains little of importance which humans have not practiced for thousands of years – without computers or computer science. ETH Zurich is a tolerant university that puts great confidence in its professors. That is why my teaching methods were never questioned.

My biggest success factor at ETH Zurich was my age. I was the youngest full professor of the department when I joined ETH Zurich at the age of 34. At that time, I still looked like a student myself. As a result, it was much easier for me to recruit students for my research than for all the other professors because students were not shy to approach me. I also did well in teaching evaluations for the same reason. The early ETH Zurich days were great fun.

Unfortunately, I did not stay 34 forever. Ten years later, the magic was gone. I was on the side of the professors who maintained that students learned for life, but that the students just didn’t know it yet. The realization that I had to do something different came to me a bit later when one of my Ph.D. students burst into my office, eyes shining, and told me proudly that he had a terrific idea. Instead of being excited, I thought: “Oh, dear, not again.” At that moment, I knew I was finished as a teacher and as a professor. I decided to join Microsoft in Redmond (USA) and focus my research on making the world a better place. I am still a computer scientist, and I look forward to the second half of my career as a computer scientist. It is a blessing that in this profession it is possible to hit refresh and start from scratch.

So much about my career as a computer scientist. My career as a husband (not relevant for this book) and as a father began much later than my professional career, but it was just as straightforward – and has happily not led to any type of burnout. I am the proud father of four teenagers – one boy and three girls – all healthy and intelligent and developing splendidly. They are simply fun. But my children ask difficult ques-

tions: “How come you became a computer scientist?” “What career should I choose?” In all those years, I had apparently not answered the most important question of all: Why? I had weaseled myself through a whole career as a professor without an answer. I have no intention of failing here as a father.

1.2 About This Book

The first goal of this book is to find the answer – in hindsight – of why I was so lucky to have studied computer science. However, I am not writing this book for myself. Rather, I am writing it for my children, who need to decide whether they want to go to college and, if so, what to study. They shouldn’t study computer science by chance, and they also should not *not study* it by chance either.

This book is therefore directed at teenagers who have had no experience with programming or computer science – just like my children and myself when I was their age. After receiving feedback from high-school students, however, it became clear that this book could also be of value for teenagers who have already been infected by the computer-science virus because it gives a different, somewhat unusual perspective on the field. I have also received positive feedback from adults who appreciate an intuitive introduction to computer science and artificial intelligence.

I chose my field of study by chance, a gut feeling, and a few other factors added in. Typical factors for choosing any subject of study are role models, talent (feel blessed if you are talented in exactly one field), cultural environment, economic factors, etc. My brother was my role model, and I had a bit of mathematical talent (albeit clearly nothing else). I don’t come from an engineering family, though my parents had come to terms with such an exotic field as computer science, seeing that my brother had already studied it. The economic factor also played a role: Computer science was generally considered to be a profession with a future! As I know today, that was an elegant excuse for adults to evade other questions about computer science because it was (and still is) considered too complicated. I misinterpreted that statement and believed that one could make a lot of money with computer science. I was right, but I had no idea of how long it would take: three decades. The social status of computer scientists has also improved over the last 30 years, although many people still don’t understand this profession and look upon it with suspicion.

In hindsight, I can say two key factors about my approach to choosing a field of study:

- This approach is not bad. It worked for me and countless others. After all, gut feeling is a great adviser.
- No matter what factors you take into account, they are irrelevant. I know of no existing relevant factors. I don't believe in passion (success generates passion, not the other way around!), in talent (everybody can learn most anything and everything), or in financial considerations (the world changes too quickly). What I do believe in is role models. We all should have role models and learn from them, but we should not follow them blindly. What worked for me may not work for my children.

The alternative approach works as follows: Gather a great amount of information on several fields of study and wait until it “clicks” for you. This approach requires work. I don't know whether it actually delivers better results; but it does have one advantage: You learn a lot in the process.

There have been and still are many misconceptions about computer science which can lead totally unsuited people to study it. Worse still, many students who are literally made for computer science fail to study it. In particular, teenage girls are often distracted from computer science for all the wrong reasons. Misconceptions arise because computer science is still a young science that is continuously evolving. In most secondary schools, it is taught only sporadically or incorrectly. To make matters worse, computer science sometimes creates a bubble, leading to additional misunderstandings and confusion. I remember the dot-com boom around the turn of the millennium. The big thing at the time were companies from the so-called “new economy,” which caused a rush of students to flock to computer science. These students were dreaming of big money, creative jobs, working environments without a dressing code. These jobs would consist of working in crazy teams to invent wild new things for the internet. As a whole, this picture was not wrong; computer scientists do work in teams and could care less about their appearance. In the IT industry, only innovation matters, and I still believe that the IT industry offers the coolest jobs with the best perks. But the reality of *studying* computer science was and is different!

Believe me, studying computer science is not cool. Students are largely left alone, and in the end, it is no more or less creative than any other field of study. I remember my first steps in the year 2000 as an Associate Professor at the Technical University of Munich in the middle of the dot-com boom. Suddenly, we had more than 1000 freshman students to deal with. The lecture halls were bursting at the seams, and the regulations required oral exams for *every* student. This regulation led to meaningless 10-minute oral exams, which were a waste of time for both students and faculty. The curricula were not adapted to the needs of the students nor to the ever-changing needs of industry. Since it was a bubble, it was wise not to adapt the material, though that decision did nothing to better the public perception of studying computer science. The result was that the dot-com bubble burst, the students failed to pass their exams and eventually dropped out. Harry Potter's friend, Hermione Granger, did not study computer science either; that was left to the nerds. So, we were back to Square One.

The ETH Zurich and many other universities continuously evolve their curriculum to promote teamwork and creativity. Unfortunately, this will be a long process.

This book serves to prepare computer-science students for their academic studies. I would like to provide students with a better overall picture of computer science so that they have a clearer understanding of why they should (or should not) study it. This clarity should also help them overcome the storms and challenges all students face no matter what they study. Having a clear goal makes it easier to tolerate the noise and detours fundamental to any course of study. I would be pleased if this book helps to prevent misunderstandings such as those that arose during the dot-com boom. Today, we are experiencing a similar hype triggered by the Artificial Intelligence (AI) revolution. That's why I believe this book is timely.

Another important goal of this book is to generate interest in computer science among other groups. My daughter inspired me to write this book. She mentioned wanting to study medicine, so I gave her several books to introduce her to the wonderful world of medicine. Among these books were *The Brain that Changes* by A. Dodge and *The Emperor of Maladies* by S. Mukharjee. Recently, she asked me whether there were similar books on computer science. I was tempted to reply that the