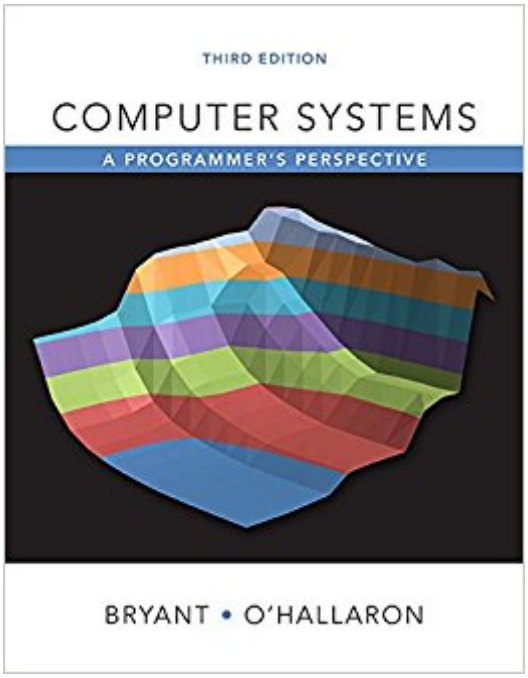
Computer Systems: A Programmer's Perspective (3rd Edition)



Synopsis

&>standaloneÂ product; MasteringEngineeringÂ®Â does not come packaged with this content. If you would like to purchase both the physical text and MasteringEngineering search for 0134123832 / 9780134123837 Â Â Computer Systems: A Programmer's Perspective plus MasteringEngineering with Pearson eTextÂ â ” Access Card Package, 3/e Â Package consists of: 013409266X/9780134092669 Computer Systems: A Programmer's Perspective, 3/e 0134071921/9780134071923 MasteringEngineering with Pearson eText -- Standalone Access Card -- for Computer Systems: A Programmer's Perspective, 3/e MasteringEngineering should only be purchased when required by an instructor. Â For courses in Computer Science and Programming Â Computer systems: A Programmerâ ™s Perspective explains the underlying elements common among all computer systems and how they affect general application performance. Written from the programmerâ ™s perspective, this book strives to teach readers how understanding basic elements of computer systems and executing real practice can lead them to create better programs. Â Spanning across computer science themes such as hardware architecture, the operating system, and systems software, the Third Edition serves as a comprehensive introduction to programming. This book strives to create programmers who understand all elements of computer systems and will be able to engage in any application of the field--from fixing faulty software, to writing more capable programs, to avoiding common flaws. It lays the groundwork for readers to delve into more intensive topics such as computer architecture, embedded systems, and cybersecurity. Â This book focuses on systems that execute an x86-64 machine code, and recommends that programmers have access to a Linux system for this course. Programmers should have basic familiarity with C or C++. Â Also availableÂ with MasteringEngineering MasteringEngineering is an online homework, tutorial, and assessment system, designed to improve results through personalized learning. This innovative online program emulates the instructorâ ™s office hour environment, engaging and guiding students through engineering concepts with self-paced individualized coaching Â With a wide range of activities available, students can actively learn, understand, and retain even the most difficult concepts. Students, if interested in purchasing this title with MasteringEngineering, ask your instructor for the correct package ISBN and Course ID. Instructors, contact your Pearson representative for more information.

Book Information

Hardcover: 1120 pages

Publisher: Pearson; 3 edition (March 12, 2015)

Language: English

ISBN-10: 013409266X

ISBN-13: 978-0134092669

Product Dimensions: 7.6 x 1.6 x 9.2 inches

Shipping Weight: 4.6 pounds (View shipping rates and policies)

Average Customer Review: 4.3 out of 5 stars 33 customer reviews

Best Sellers Rank: #2,391 in Books (See Top 100 in Books) #13 inÂ Books > Computers & Technology > Computer Science #17 inÂ Books > Textbooks > Computer Science

Customer Reviews

Randal E. Bryant received his bachelorâ ™s degree from the University of Michigan in 1973 and then attended graduate school at the Massachusetts Institute of Technology, receiving his PhD degree in computer science in 1981. He spent three years as an assistant professor at the California Institute of Technology, and has been on the faculty at Carnegie Mellon since 1984. For five of those years he served as head of the Computer Science Department, and for ten of them he served as Dean of the School of Computer Science. He is currently a university professor of computer science. He also holds a courtesy appointment with the Department of Electrical and Computer Engineering.
Â Professor Bryant has taught courses in computer systems at both the undergraduate and graduate level for around 40 years. Over many years of teaching computer architecture courses, he began shifting the focus from how computers are designed to how programmers can write more efficient and reliable programs if they understand the system better. Together with Professor Oâ ™Hallaron, he developed the course 15-213, Introduction to Computer Systems, at Carnegie Mellon that is the basis for this book. He has also taught courses in algorithms, programming, computer networking, distributed systems, and VLSI design.
Â Most of Professor Bryantâ ™s research concerns the design of software tools to help software and hardware designers verify the correctness of their systems. These include several types of simulators, as well as formal verification tools that prove the correctness of a design using mathematical methods. He has

published over 150 technical papers. His research results are used by major computer manufacturers, including Intel, IBM, Fujitsu, and Microsoft. He has won several major awards for his research. These include two inventor recognition awards and a technical achievement award from the Semiconductor Research Corporation, the Kanellakis Theory and Practice Award from the Association for Computer Machinery (ACM), and the W. R. G. Baker Award, the Emmanuel Piore Award, the Phil Kaufman Award, and the A. Richard Newton Award from the Institute of Electrical and Electronics Engineers (IEEE). He is a fellow of both the ACM and the IEEE and a member of both the US National Academy of Engineering and the American Academy of Arts and Sciences. Â David R. Oâ ™Hallaron is a professor of computer science and electrical and computer engineering at Carnegie Mellon University. He received his PhD from the University of Virginia. He served as the director of Intel Labs, Pittsburgh, from 2007 to 2010. Â He has taught computer systems courses at the undergraduate and graduate levels for 20 years on such topics as computer architecture, introductory Â computer systems, parallel processor design, and Internet services. Together with Professor Bryant, he developed the course at Carnegie Mellon that led to this book. In 2004, he was awarded the Herbert Simon Award for Teaching Excellence by the CMU School of Computer Science, an award for which the winner is chosen based on a poll of the students. Â Professor Oâ ™Hallaron works in the area of computer systems, with specific interests in software systems for scientific computing, data-intensive computing, and virtualization. The best-known example of his work is the Quake project, an endeavor involving a group of computer scientists, civil engineers, and seismologists who have developed the ability to predict the motion of the ground during strong earthquakes. In 2003, Professor Oâ ™Hallaron and the other members of the Quake team won the Gordon Bell Prize, the top international prize in high-performance computing. His current work focuses on the notion of autograding, that is, programs that evaluate the quality of other programs.

I've probably read a hundred books on computer programming and skimmed hundreds more. This is one of the best books on computer programming ever written. I can't think of any book I have ever read that would allow a motivated reader to advance every aspect of their systems programming

knowledge so rapidly. It teaches basic computer architecture, practical x86-64 assembly, and C programming all side by side and with few apparent compromises. If there is any flaw in this book it’s that a reader without much previous experience might occasionally need to consult other sources for more practice and background. Still, anyone who mastered the material in this book, along with the API of the operating system they work on, would be world class systems programmer.

This is going to be a textbook that follows me till it goes obsolete. Well written and with good examples.

This book is very readable. It gives you a pretty comprehensive tour of modern computer system while leaving many unnecessary details out.That being said, I think there are times when the authors can be more concise (e.g. chap 2 is too "mathematical"). As I know this book is usually used for a second/third undergraduate CS course, so putting too many details in the book is kind of demanding.Doing the labs is an indispensable part of reading this book. They can be found on the book site and they are amazingly fun.Some people have pointed out that this book is too "academic." Even though that is true to some extent, I think this book is still a must-read since it is meant to give you a good foundation. Once you are done with this book it is fairly easy to pick up "industry" knowledge.

Needless to say, the content is very useful for all serious programmers.Furthermore it’s extremely clear, informative and well-arranged.Concise without leaving out important knowledge, meanwhile you don’t need to worry about understanding the material.While reading the book, you feel like traveling in the computer world with a smart compass that keeps telling you everything you just need, no more, no less.Nice book!

I really like this book. I’m new to computer systems, and this text was enlightening and easy to understand. Would recommend to other students new to systems!

Apparently the exercises in this book is different from the US version (according to the author himself) so if you are looking to work on the problems included perhaps you’ll want to get the US version.

New, Nice, Expesive

This book arrived on time and was exactly what was needed for class.

Computer Systems: A Programmer's Perspective (3rd Edition) Draw in Perspective: Step by Step, Learn Easily How to Draw in Perspective (Drawing in Perspective, Perspective Drawing, How to Draw 3D, Drawing 3D, Learn to Draw 3D, Learn to Draw in Perspective) Ada Lovelace, Poet of Science: The First Computer Programmer Ada's Ideas: The Story of Ada Lovelace, the World's First Computer Programmer Computer Programmer (Careers with Character (Mason Crest)) 1st Grade Computer Basics : The Computer and Its Parts: Computers for Kids First Grade (Children's Computer Hardware Books) Computer Forensics: Investigating File and Operating Systems, Wireless Networks, and Storage (CHFI), 2nd Edition (Computer Hacking Forensic Investigator) Mathematics and Computer Science in Medical Imaging (Nato a S I Series Series III, Computer and Systems Sciences) Automation and Systems Issues in Air Traffic Control (Nato a S I Series Series III, Computer and Systems Sciences) CMOS VLSI Design: A Circuits and Systems Perspective (3rd Edition) Computer Organization and Design MIPS Edition, Fifth Edition: The Hardware/Software Interface (The Morgan Kaufmann Series in Computer Architecture and Design) The Automated Lighting Programmer's Handbook How Not to Make a Short Film: Secrets from a Sundance Programmer Fractals: A Programmer's Approach Getting to Know the Raspberry Pi (Code Power: A Teen Programmer's Guide) Getting to Know Scratch (Code Power: A Teen Programmer's Guide) Getting to Know Alice (Code Power: a Teen Programmer's Guide) Getting to Know Ruby (Code Power: A Teen Programmer's Guide) Getting to Know Hackety Hack (Code Power: a Teen Programmer’s Guide) Getting to Know Python (Code Power: a Teen Programmer’s Guide)