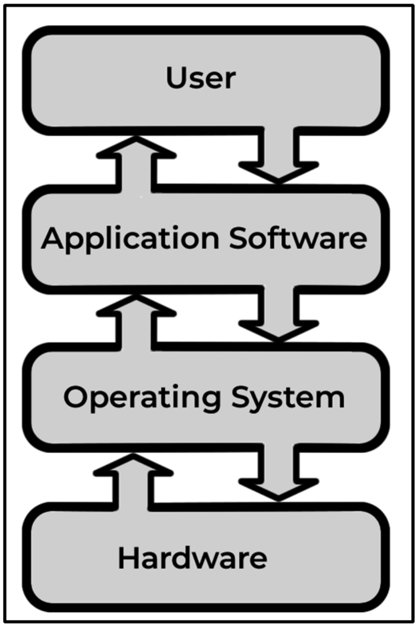
Foreword

Why study this book? To understand the importance of this type of book, one needs to look at the topics in the book and their relevance to students studying a computing discipline in the 21st century. The relevance and importance of the topics are crucial in understanding the foundations for information technology, computer science, and the other computing disciplines. Why are the topics included in this book important to you?

If you were building a house, you would not start by putting walls up without first preparing a foundation. A foundation serves to keep the house stable. When a foundation is unstable, the whole house becomes unstable. How does this relate to computing? The concepts in this book are the foundation for computing disciplines.

The diagram shows the hierarchy of the interaction between a computing user, application software, operating systems, and hardware.

Notice that hardware is at the bottom of the hierarchy. It is our computing foundation. Hardware interacts with software, both operating systems and application software and all of these interact with users. Each one of these levels has many components. To get to each higher level we need to understand the levels below it. To build and use an operating system correctly, we need to understand all components that make up computing hardware and how they interface with the operating systems. To understand the use and need for application software, we should know the basis of operating systems so we can write software application code to work efficiently and accurately with the operating system and hardware. The application software must support the needs of the users.



Hierarchy of computing interactions
McGeddon
English Wikipedia

Most computing disciplines concentrate on the "soft" side of computing – primarily operating systems and computer applications. Computer engineers are more concerned with computer hardware – in other words, their discipline teaches how to design and build the hardware components. However, this does not mean that computing majors don't need to know anything about hardware and how it interfaces with the operating system and application software. Practitioners of information technology computer science, and other computing disciplines need to know what components make up the hardware of a computer system, but not how to design and manufacture the actual hardware.

Many computer programmers believe they don't really need to understand operating systems or hardware because they are just working with computer programs. A computer programmer who uses, writes, and updates application software needs the knowledge of how operating systems work, and how the hardware components interface with the operating system. This knowledge is crucial in writing complete, efficient, and usable software applications. However, most programmers feel very removed from the operating systems and hardware. As shown in the hierarchy diagram, the interfacing of hardware, operating systems, application software and users is both "up" and "down". Computing professionals need to be able to move both up and down in the hierarchy. Our primary study in information technology and computer science is the operating system and application software, but many information technology and computer science students go into networking where they work with the physical network and the software that controls it. They also go into cloud computing or web programming where they interface with physical and virtual networks. Computer security is a huge area of study in computing the practitioners of which need to have a good understanding of hardware and and its interaction with software in order to protect both.

This book is dedicated to helping students in the computing disciplines master the basics of computing hardware and how it interfaces with the operating system, application software, and the users. It provides some fundamental information on how information is represented in computer systems and surveys some areas you will study more deeply in later classes.

Dr. Bob Brown has been teaching hardware and software concepts for many years to undergraduate students majoring in computing disciplines. He also worked for 30 years in the information technology industry dealing with hardware, software, and operating systems. His professional work experience has given him a unique knowledge of these areas and he brings this knowledge to this textbook. Learning these important foundational concepts will give you a solid foundation for your computing career.

Rebecca H. Rutherfoord
Professor Emerita of Information Technology
former Information Technology Department Chair
June, 2023

Preface

About this book

In more than twenty years of university teaching, I spent countless hours translating textbook information into language comfortable to students who were engaged with the material. Prior to my career in academia, I spent nearly thirty years managing information technology in the healthcare industry. The person who recommended me for my first university teaching position told the department chair that the basics of computing hardware and software "made sense" to me. After fifty years of the material making sense to me and helping others make sense of it, and after I had retired from teaching, I convinced myself that I could explain it in writing in a way that would be clear to university students.

This book is suitable for those with no prior study of computer systems, although it may be helpful to have had prior experience with a high-level programming language such as Java or Python.

The careful reader will notice that the level of detail in this book is highly variable. That is the result of a conscious decision to delve into those parts that are likely to be mysterious to students and treat more lightly the parts with which most students are comfortable and those they are likely to study in more detail in later classes. This book principally addresses students of information technology. They will have full semester classes in subjects like operating systems and information security, but almost none will have a course in computer architecture. The chapters on digital logic and how the CPU and memory work have enough detail to stand alone. Subjects like information security are written to provide a starting point for further study.

The careful reader will also notice numerous references to people, their contributions to computing, and the dates of those contributions. Students who may read this book have never known a world without iPhones. I believe it is important that they know the inventions that make 21st century computing possible did not spring fully formed from some Silicon Valley corporate campus.

This is an open access, that is, free[1] book, I've tried to achieve the same level of quality as books costing hundreds of dollars. Every chapter has been reviewed by a peer and at least one subject matter expert, then proofread by a colleague who is an absolute stickler for detail. I hope you won't judge the value of this book by its cost.

[1] Unless you want a "real book," printed on dead trees or a formatted ePub book. Those necessarily cost money, but in this case, not very much.

To the student

This book tells a story. It's a story with characters and events, and a plot line that leads from the earliest attempts at automatic computation to the ubiquitous, seamless, always-connected world of computing in the 21st century. I urge you to read it like a story, beginning at Chapter 0 and continuing to the end. If you do that and put effort into seeing how the pieces fit together, you will come away with a conceptual model of computing hardware and software that, one hopes, has just the right amount of detail in every part to prepare you for your future studies.

Conventions

Boldface indicates defined words and important concepts.
Italics are for emphasis, for foreign words and abbreviations, publication titles, and to highlight proper names in the index.
SMALL CAPS are function names, operators, logic signals, and instruction mnemonics.
`Monospaced text` is used for code, of which this book has very little.

Acknowledgments

This book would not have been possible without the help and encouragement of Bob Harbort, Emeritus Professor of Computer Science, Kennesaw State University. He read every word of this book, many of them more than once, and provided advice and discussion that vastly improved on the original draft.

The subject matter expert reviewers checked the details, spotted my mistakes, and kept me on track. Their help was essential to success. Thank you to James Cannady, Georgia Tech Research Institute, Sumit Chakravarty, Lance Crimm, Bill Forsyth, Dan Lo, Douglas Malcolm, and Liang Zhao of Kennesaw State University.

Thanks to Becky Rutherfoord for help, encouragement, and for the Foreword to this book.

Thanks also to Betty Abbott, Katie Abbott, Cindy Neck, and Anthony Trauring for their painstaking and careful reading and for their helpful suggestions.

The cover design and the Martian in Figure 3-3 are by Andrea Vom Ende Thoma.

Thanks to Ken Saladin, author of *Anatomy & Physiology–The Unity of Form and Function*, for advice on the structure of the chapter summaries of learning objectives.

Errata

I had an enormous amount of help with this book, but any errors that remain are mine alone. As errors are discovered and reported, I will maintain a list of errata here: https://computingconcepts.net/errata.html

The latest edition of the book is available here: https://computingconcepts.net/

Remarks and corrections

If you have remarks on this material, and especially if you find errors, please write to me: Bob.Brown@Kennesaw.edu.

Emory Village, Georgia
July, 2023