Preface

The *Concise Guide to Computing Foundations* is designed to meet two main goals. The first goal is to help science students understand how the computers work so they can better use the computers and software when doing their work. The second goal is to provide an introduction into computational science. This book introduces topics from computer science that are important for scientists to understand. This includes (but not limited to) understanding how simulations work (including their strengths and limitations), understanding the computer's precision, and how much work the software needs to do to complete the task. It also takes these and other concepts in computer science and applies them to different topics in a number of scientific disciplines to help the science students understand how the two areas (science and computers) interact. The book is designed to be taught to second semester freshmen or sophomore science majors who have already had one major science lab course in their major and are ready to take calculus.

Why the Book Was Written

The book is the result of an NSF CCLI grant to study a new approach to teaching noncomputer science majors what they need to know about computing. Other introductory courses for nonmajors typically include one or more of the following: (a) how to program, (b) how to use office suite (and other software), (c) an overview of how computers work, and (d) ethics of computing and society. The need for the first two (programming and using office suite software) has become less essential for the following reasons. With the development of extremely powerful applications and simulations, fewer non-computer scientists need to program. The software needed to do advanced work often already exists and can be customized to the needs of the user without strong programming skills. In addition, many users are already able to use a computer for everyday tasks (via word processing, spreadsheets, email, browsing the web, etc.).

However, as society becomes more and more dependent on computers, users are trusting them more without understanding how they work. There are aspects of the working computing that are important for users to understand. These aspects vary from area to area. For scientists, it is important to understand the precision of computers and sensors so that they understand where errors can creep into their results. Musicians need to understand what is lost when audio files are compressed. Researchers that use computers to study language and literature need to understand how computers understand and interpret language. Every discipline depends on computing in an increasing amount. But it is essential for the user to understand what is happening within the computer to know how to use the results intelligently.

This textbook is an attempt to develop this new approach to computing for non-computer science majors. We decided to work in the area of the sciences because we were most familiar with that area and had a strong interest of some science faculty to help us. We hope this book can help the science students become better scientists by understanding how the computer helps to solve their problems.

Why You Should Be Interested in This Book

There are three primary reasons someone would be interested in this book. The first is that computing is everywhere and it is essential to understand many aspects of how a computer works. Today's society is extremely dependent on computers. These devices impact just about every aspect of life. The impact is not a surface impact any more. Computers are fundamental to almost every area. We rely on them to make our society work at the level it does. And as time progresses, this reliance will only become greater. It is very short sighted to develop a society where every aspect of it is very dependent on a technology that very few have any understanding of. Most of modern life is dependent on electricity. While most people may not understand the details of how electricity works, they know enough to know how to use it wisely, when it is dangerous, and what to do if it fails. But many people in today's society lack a similar level of understanding about computers. We don't understand what they are capable of, why they may not work, where the limitations are, and where errors can occur. This book seeks to address some of these areas for non-computing professionals. It aims to help the average person better understand important aspects of computing in the areas in which they work.

This leads to the second reason why someone would be interested in this book. This book offers a new approach on educating the general student about computing to address the needs of today's society. As mentioned previously, while knowing how to use office suite software is a valuable skill, it does not help the user understand how a computer works. While knowing how to program a computer does help users understand the workings of a computer, most users will not need to write programs. The knowledge that would be more useful is understanding how computers do their work at a level that impacts the results. How computers convert electrical signals to words on the screen is not important to most people (just as the details of how a battery works are not to important to quality use of electricity). However, scientists need to understand the limits of precision in a computer just as most people need to understand that it is dangerous to overload an electrical circuit.

This level of knowledge is what many of today's students need as they prepare for a future where computers are very pervasive.

The third reason for someone to be interested in this book is because they want to be better in their profession. As people use computers in careers, they can view the computer as an able assistant that they manage or they can view computers as a black box that they need to accommodate. Those that understand how the computer works in their profession can optimize the use of the computer to meet their needs. Those that don't have to adjust their work to the computer. The first type of person is much more productive in their job and able to do more difficult work. This book seeks to help develop such a person. While the focus on this book is for STEM professionals, the lessons can be applied (and expanded) to all other disciplines.

Who Is This Book for?

This book is aimed for STEM majors who have taken at least one lab science course in their major. These students should be excited about science, technology, engineering, and mathematics and feel comfortable in at least one of these areas. It assumes a basic understanding of the scientific method. There should be a strong motivation to study these areas. While the student is not required to do any calculus, there is some calculus shown in a few modules. Students cannot be fearful of mathematics and at least ready to study calculus. This book is also aimed at the sophomore level. This is to ensure that students are ready to do discovery learning (the primary method of the book) but also leaves them time to pursue additional work in this area later in their degree.

Additional Considerations

Those using this book should expect some additional benefits. Because this book takes examples from a number of different STEM areas, students will start to make connections between the different STEM branches. A student may experience a connection between chemistry and biology. Another might see a connection between geology and engineering. A third might see how mathematics and computing relate. There are many opportunities to build these bridges between all the different disciplines. Students will also get a good interdisciplinary introduction to computational science. This will spark enough interest in some of the students to pursue further studies in a very dynamic and promising field.

Organization of the Textbook

This book has two main categories of modules. This first category is computer science chapters. These chapters explain specific computer science concepts using different aspects of science as examples. These include modules on computational science, data representation, algorithms, self-defining data, and performance complexity. The second category is science modules where a specific science topic is explained and the connection to the related computer science is drawn. The areas of science include bioinformatics, chemical kinetics, engineering analysis, GIS, flow analysis, solving equations, curve fitting, optimization, and data acquisition. The book also includes a glossary and some short tutorials for some of the software used.

Guide to the Students

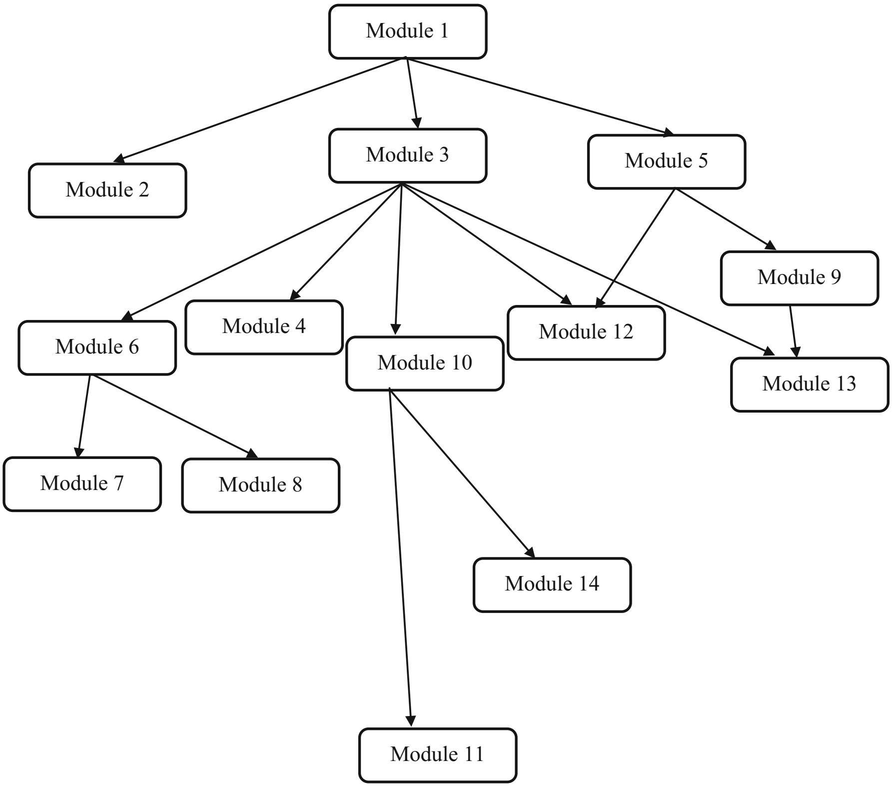
As is true with most of the things you study, you will get out of this what you put in. Try to engage the material as much as possible. There are "Discuss It" questions. Don't skip these. Discuss them with classmates. They are there to help you make important connections and to apply what you learn. Don't just try to complete the modules. Instead, always asked yourself what is this module trying to teach me at this moment about computing and the sciences. When working on modules that are not your discipline, try to make connections back to things you are familiar with. Try to apply the knowledge in other areas. To aid in doing this, try to work with students from other disciplines. This will broaden your knowledge and help you see things in a different way.

Guide to Instructors

You don't need to understand the details of all the STEM areas covered to be qualified to teach this course. This course has been taught by a computer science professor (with an interest in the sciences but not a trained scientist). This course has also been taught by a geology professor (with some background in computer science). The modules are designed to be self-teaching. So you can learn with your students. Teach them to how to be self-learners by demonstrating your own learning skills. Also, if the developers of this work (authors and contributors of given chapters) can be of any help, please contact us (contact information can be found at the book's website). All of us would like to be of help in any way possible.

The questions at the end of many of the modules (computing questions) are optional questions designed to help the students explicitly think about computing questions when working on modules in the sciences. Use them as you think appropriate.

If you are using this as the primary textbook for a course, it does not need to be covered sequentially (via the module numbers). We recommend that you use "just-in-time" methodology when picking the order to cover the modules. Cover enough of the computer science necessary for the other modules. Next is a dependency chart for each modules. Feel free to take any path through the modules. Also feel free to skip modules that you don't want to use.



To aid in teaching this course, we have a number of resources. The first is an answer key. We also maintain on the website the software needed (and any extra files that would be helpful). The primary webpage for this book is http://www.springer.com/9783319299525. If you are interested in the background of this work and possible contributing additional modules to future additions, additional information can be found at: http://twiki.olivet.edu/twiki/bin/view/ComputationalScience/WebHome.

This work can also be used as supplements to other STEM courses. You can assign the students to work with the necessary computer science modules to do the modules associated with a specific course. You can also use material to supplement other courses by asking them specific computer science questions.

Uses of This as a Class

This work has been used as a class at the home institution for over 4 years. Each module assigned is allocated an average of a week. Students work in pairs going through the modules on their own with the instructor stopping the class every so often to ask important questions. There have been about two or three exams in the semester. The last 2 or 3 weeks of the class are devoted to semester projects. Students (working individually or in pairs) develop a project of their own choosing applying what they have learned. They then present the projects in class. It has been very insightful to see them apply things that they have learned to new areas. The students are also very excited to be given a chance to apply this work to something they are interested in.

Future of This Work

There are many ways this work can be continued. If you are interested in any of the following, please contact the authors. One area of expansion would be more STEM modules. A computer science module on data mining is needed. Other computer science modules might be very appropriate. In addition, other STEM areas might benefit from additional modules.

However, this new approach to teaching computer science to the general student can easily be expanded to non-STEM areas. Chapters on the foundations of computer science as used in music would be very appropriate. The same is true for all disciplines. If you are interested in expanding this work to new disciplines, please contact us.

Thanks

Many people were involved in this work. We would like to thank them all.

- Dr. Larry Vail contributed a great deal to this work (both in the writing of a number of the modules and in pioneering the teaching of this course).
- Dr. Willa Harper contributed the chemistry knowledge and was able to make sure that the computer science was not too technical.
- Dr. Greg Long contributed the biology knowledge and his excitement for the project keeps us going.
- Dr. Joe Schroeder helped with the engineering knowledge.
- The NSF was foundational in this work because they believe in the overall concept and funded the work.

Bourbonnais, IL, USA

Kevin Brewer
Cathy Bareiss

http://www.springer.com/978-3-319-29952-5

Concise Guide to Computing Foundations
Core Concepts and Select Scientific Applications
Brewer, K.; Bareiss, C.
2016, XV, 191 p. 76 illus., 41 illus. in color., Hardcover
ISBN: 978-3-319-29952-5