



openStax™

Introduction to **Computer Science**



Study where you want, what you want, when you want.

When you access your book in our web view, you can use our new online **highlighting and note-taking** features to create your own study guides.

Our books are free and flexible, forever.

Get started at openstax.org/details/books/introduction-computer-science

College Success **2.2 The Motivated Learner**

Table of contents

- Preface
- 1 Exploring College
- 2 Knowing Yourself as a Learner
 - Introduction
 - 2.1 The Power to Learn
 - 2.2 The Motivated Learner**
 - 2.3 It's All in the Mindset
 - 2.4 Learning Styles
 - 2.5 Personality Types and Learning
 - 2.6 Applying What You Know about Learning
 - 2.7 The Hidden Curriculum
- Summary
- Career Connection
- Reflecting
- Where do you go from here?
- 3 Managing Your Time and Priorities
- 4 Planning Your Academic Pathways
- 5 Reading and Note-taking
- 6 Studying, Memory, and Test-taking
- 7 Thinking
- 8 Communicating
- 9 Understanding Civility and Cultural Competence
- 10 Understanding Financial Literacy
- 11 Engaging in a Healthy Lifestyle
- 12 Planning for Your Future
- A Conducting and Presenting Research
- B Recommended Readings
- C Activities and Artifacts From the Book
- Index

Search this book: My highlights Print

Resilience and Grit

While much of this chapter will cover very specific aspects about the act of learning, in this section, we will present different information that may at first seem unrelated. Some people would consider it more of a personal outlook than a learning practice, and yet it has a significant influence on the ability to learn.

What we are talking about here is called grit or resilience. Grit can be defined as personal perseverance toward a task or goal. In learning, it can be thought of as a trait that drives a person to keep trying until they succeed. It is not tied simply to a tendency not to give up until something is finished or accomplished.



Figure 2.2 U.S. Army veteran and captain of the U.S. Paralympic team, Will Reynolds, races to the finish line. (Credit: DoD Photo / Flickr) Attribution: CC BY 2.0

The study showed that grit and perseverance were better predictors of academic success and achievement than talent or IQ.

This personality trait was defined as "grit" by the psychologist Angela Duckworth.⁷ In a 2007 study Duckworth and colleagues found that individuals with high grit were able to maintain motivation in learning tasks despite failures. The study examined a cross-section of learning environments, such as GMAT scores in Ivy League universities, dropout rates at West Point, rankings in the National Spelling Bee, and general educational achievement for adults. What the results showed was that grit and perseverance were better predictors of academic success and achievement than talent or IQ.

Applying Grit

The concept of grit is an easy one to dismiss as something taken for granted. In our culture, we have a number of sayings and aphorisms that capture the essence of grit: "If at first you do not succeed, try, try again," or the famous quote by Thomas Edison: "Genius is one percent inspiration, ninety-nine percent perspiration."

Access. The future of education.
openstax.org



RICE



1

Introduction to Computer Science

Figure 1.1 Computing is everywhere, affecting everyone, for better and for worse. (credit: modification of "Whereas design is expansive, engineering is narrowing" by Jessie Huynh/Critically Conscious Computing, CC0)

Chapter Outline

- [**1.1** Computer Science](#)
- [**1.2** Computer Science across the Disciplines](#)
- [**1.3** Computer Science and the Future of Society](#)



Introduction

This textbook will introduce you to the exciting and complex world of computer science. In this chapter, you'll review the history of computer science, learn about its use in different fields, and explore how computer science will impact the future of society. Computer science is a powerful tool, and computer scientists have used their vast knowledge of technology to create and implement technology that has transformed societies around the world.

This book will also introduce the computational thinking aspects of problem-solving and analytical thinking that enable the study of algorithms, which are step-by-step instructions for solving specific problems or carrying out computations. Therefore, this book also covers algorithms and their realization via programming languages, computer systems architectures, networks, and operating systems. The book subsequently delves into computer science areas that enable the design and development of software solutions using high-level programming languages (i.e., coding languages designed to be more intuitive for humans), architectural styles and related models, data management systems, and software engineering. Finally, the book demonstrates how to leverage computer science realizations and areas to build modern end-to-end solutions to business and social problems. In particular, the book focuses on modern web applications development, cloud-native applications development, and hybrid Cloud/on-premise digital solutions. The various chapters emphasize how to achieve software solution qualities such as performance and scalability. The last chapter explains how to secure software applications and their applications in the context of various cyber threats. It also explains how to make the right decisions about using computers and information in society to navigate social, ethical, economic, and political issues that could result from the misuse of technology. To conclude this textbook, we'll introduce you to cybersecurity and help you understand why responsible computing is essential to promote

ethical behavior in computer science. The book is designed to help students grasp the full meaning of computer science as a tool that can help them think, build meaningful solutions to complex problems, and motivate their careers in information technology (IT).

You're already familiar with computer science. Whenever you use a laptop, tablet, cell phone, credit card reader, and other technology, you interact with items made possible by computer science. Computer science is a challenging field, and the outputs of computer science offer many benefits for society. At the same time, we have to be cautious about how we use computer science to ensure it impacts society in ethical ways. To help you understand this, the next section will explain how computer science came to be and discuss the field's potential.

1.1 Computer Science

Learning Objectives

By the end of this section, you will be able to:

- Discuss the history that led to the creation of computer science as a field
- Define computer science
- Assess what computer science can do, as well as what it should not do

The field of **computer science (CS)** is the study of **computing**, which includes all phenomena related to computers, such as the Internet. With foundations in engineering and mathematics, computer science focuses on studying algorithms. An **algorithm** is a sequence of precise instructions that enables computing. This includes components computers use to process information. By studying and applying algorithms, computer science creates applications and solutions that impact all areas of society. For example, computer science developed the programs that enable online shopping, texting with friends, streaming music, and other technological processes.

While computers are common today, they weren't always this pervasive. For those whose lives have been shaped by computer technology, it can sometimes seem like computer technology is ahistorical: computing often focuses on rapid innovation and improvement, wasting no time looking back and reflecting on the past. Yet the foundations of computer science defined over 50, and as much as 100, years ago very much shape what is possible with computing today.

The Early History of Computing

The first computing devices were not at all like the computers we know today. They were physical calculation devices such as the abacus, which first appeared in many societies across the world thousands of years ago. They allowed people to tally, count, or add numbers ([Figure 1.2](#)). Today, abaci are still used in some situations, such as helping small children learn basic arithmetic, keeping score in games, and as a calculating tool for people with visual impairments. However, abaci are not common today because of the invention of number systems such as the Arabic number system (0, 1, 2, 3, . . .), which included zero and place values that cannot be computed with abaci. The concept of an algorithm was also invented around this time. Algorithms use inputs and a finite number of steps to carry out arithmetic operations like addition, subtraction, multiplication, and division, and produce outputs used in computing. Today's computers still rely on the same foundations of numbers, calculations, and algorithms, except at the scale of billions of numbers and billions of calculations per second.

To introduce a concrete example of an algorithm, let us consider binary search algorithm, which is used to locate a number in a sorted array of integers efficiently. The algorithm operates by repeatedly dividing the search interval in half to perform the search. If the number being searched is less than the integer in the middle of the interval, the interval is narrowed to the lower half. In the alternative, the interval is narrowed to the upper half. The algorithm repeatedly checks until the number is found or the interval is empty.