

Chapter 1

An Introduction to Computer Science



**INVITATION TO
Computer Science**

6TH
EDITION

Objectives

After studying this chapter, students will be able to:

- Understand the definition of computer science
- Write down everyday algorithms and evaluate them to determine if they are ambiguous or not effectively computable

Objectives (continued)

After studying this chapter, students will be able to:

- Understand the roots of modern computer science in mathematics thousands of years old and in mechanical machines hundreds of years old
- Summarize the key points in the historical development of modern electronic computers
- Map the organization of the text onto the definition of computer science

Introduction

- Misconceptions
 - Computer science is:
 - The study of computers
 - The study of how to write computer programs
 - The study of the uses and applications of computers and software

The Definition of Computer Science

- **Computer science** is the study of algorithms, including:
 - Their formal and mathematical properties
 - Their hardware realizations
 - Their linguistic realizations
 - Their applications

The Definition of Computer Science (continued)

- Algorithm
 - Informally, “an ordered sequence of instructions that is guaranteed to solve a specific problem.”
- Operations used to construct algorithms
 - Sequential operations
 - Conditional operations
 - Iterative operations

FIGURE 1.1

- Step 1** If the clock and calendar are not correctly set, then go to page 9 of the instruction manual and follow the instructions there before proceeding to Step 2
- Step 2** Place a blank disc into the DVR disc slot
- Step 3** Repeat Steps 4 through 7 for each program that you want to record
- Step 4** Enter the channel number that you want to record and press the button labeled CHAN
- Step 5** Enter the time that you want recording to start and press the button labeled TIME-START
- Step 6** Enter the time that you want recording to stop and press the button labeled TIME-FINISH. This completes the programming of one show
- Step 7** If you do not want to record anything else, press the button labeled END-PROG
- Step 8** Turn off your DVR. Your DVR is now in TIMER mode, ready to record

Programming your DVR: An example of an algorithm

FIGURE 1.2

Given: $m \geq 1$ and two positive numbers each containing m digits, $a_{m-1} a_{m-2} \dots a_0$ and $b_{m-1} b_{m-2} \dots b_0$

Wanted: $c_m c_{m-1} c_{m-2} \dots c_0$, where $c_m c_{m-1} c_{m-2} \dots c_0 = (a_{m-1} a_{m-2} \dots a_0) + (b_{m-1} b_{m-2} \dots b_0)$

Algorithm:

- Step 1** Set the value of *carry* to 0
- Step 2** Set the value of i to 0
- Step 3** While the value of i is less than or equal to $m - 1$, repeat the instructions in Steps 4 through 6
- Step 4** Add the two digits a_i and b_i to the current value of *carry* to get c_i
- Step 5** If $c_i \geq 10$, then reset c_i to $(c_i - 10)$ and reset the value of *carry* to 1; otherwise, set the new value of *carry* to 0
- Step 6** Add 1 to i , effectively moving one column to the left
- Step 7** Set c_m to the value of *carry*
- Step 8** Print out the final answer, $c_m c_{m-1} c_{m-2} \dots c_0$
- Step 9** Stop

Algorithm for adding two m -digit numbers

The Definition of Computer Science (continued)

- Why are formal algorithms so important in computer science?
 - If we can specify an algorithm to solve a problem, then we can automate its solution
- Computing agent
 - Machine, robot, person, or thing carrying out the steps of the algorithm
- Unsolved problems
 - Some problems are unsolvable, some solutions are too slow, and some solutions are not yet known

Algorithms

- The Formal Definition of an Algorithm
 - A well-ordered collection of unambiguous and effectively computable operations that, when executed, produces a result and halts in a finite amount of time

Algorithms (continued)

- Well-ordered collection
 - Upon completion of an operation we always know which operation to do next
- Ambiguous statements
 - Go back and do it again (Do *what* again?)
 - Start over (From *where*?)

Algorithms (continued)

- Unambiguous operation, or **primitive**
 - Can be understood by the computing agent without having to be further defined or simplified
- It is not enough for an operation to be understandable
 - It must also be *doable* (**effectively computable**) by the computing agent

Algorithms (continued)

- Algorithm
 - Result must be produced after the execution of a finite number of operations
 - Result may be a number, text, a light, picture, sound, or a change in the computing agent's environment
- Infinite loop
 - Runs forever
 - Usually a mistake



FIGURE 1.3

Step	Operation
1	Wet your hair
2	Set the value of <i>WashCount</i> to 0
3	Repeat Steps 4 through 6 until the value of <i>WashCount</i> equals 2
4	Lather your hair
5	Rinse your hair
6	Add 1 to the value of <i>WashCount</i>
7	Stop, you have finished shampooing your hair

A correct solution to the shampooing problem



FIGURE 1.4

Step	Operation
1	Wet your hair
2	Lather your hair
3	Rinse your hair
4	Lather your hair
5	Rinse your hair
6	Stop, you have finished shampooing your hair

Another correct solution to the shampooing problem

Algorithms (continued)

- The Importance of Algorithmic Problem Solving
 - “Industrial revolution” of 19th century
 - Mechanized and automated repetitive physical tasks
 - “Computer revolution” of the 20th and 21st centuries
 - Mechanized and automated repetitive mental tasks
 - Through algorithms and computer hardware
- See: “A Brief History of Computing”

Organization of the Text

Computer science is the study of algorithms including:	Levels of the text:
1. Their formal and mathematical properties,	Level 1: The Algorithmic Foundations of Computer Science
2. Their hardware realizations,	Level 2: The Hardware World Level 3: The Virtual Machine
3. Their linguistic realizations,	Level 4: The Software World
4. Their applications.	Level 5: Applications Level 6: Social Issues

FIGURE 1.9



Summary

- Computer science is the **study of algorithms**
- An **algorithm** is a well-ordered collection of unambiguous and effectively computable operations that, when executed, produces a result and halts in a finite amount of time
- If we can **specify** an algorithm to solve a problem, then we can **automate** its **solution**
- Computers developed from mechanical calculating devices to modern electronic marvels of miniaturization