Chapter 1: An Introduction to Computer Science

Invitation to Computer Science, C++ Version, Third Edition
Objectives

In this chapter, you will learn about:

- The definition of computer science
- Algorithms
- A brief history of computing
- Organization of the text
Introduction

Common misconceptions about computer science:

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

- Gibbs and Tucker definition of computer science
  - The study of algorithms, including their:
    - Formal and mathematical properties
    - Hardware realizations
    - Linguistic realizations
    - Applications
The Definition of Computer Science (continued)

- Computer scientist designs and develops algorithms to solve problems
- Operations involved in designing algorithms:
  - Formal and mathematical properties
    - Studying the behavior of algorithms to determine whether they are correct and efficient
  - Hardware realizations
    - Designing and building computer systems that are able to execute algorithms
The Definition of Computer Science (continued)

- Linguistic realizations
  - Designing programming languages and translating algorithms into these languages

- Applications
  - Identifying important problems and designing correct and efficient software packages to solve these problems
The Definition of Computer Science (continued)

- **Algorithm**
  - Dictionary definition
    - Procedure for solving a mathematical problem in a finite number of steps that frequently involves repetition of an operation
    - A step-by-step method for accomplishing a task
  - Informal description
    - An ordered sequence of instructions that is guaranteed to solve a specific problem
The Definition of Computer Science (continued)

- An algorithm is a list that looks like
  - STEP 1: Do something
  - STEP 2: Do something
  - STEP 3: Do something
  - ...
  - ...
  - ...
  - ...
  - STEP N: Stop, you are finished
The Definition of Computer Science (continued)

- Categories of operations used to construct algorithms
  - Sequential operations
    - Carries out a single well-defined task; when that task is finished, the algorithm moves on to the next operation
  - Examples:
    - Add 1 cup of butter to the mixture in the bowl
    - Subtract the amount of the check from the current account balance
    - Set the value of x to 1
Conditional operations

- Ask a question and then select the next operation to be executed on the basis of the answer to that question

Examples

- If the mixture is too dry, then add one-half cup of water to the bowl
Conditional operations examples (continued):

- If the amount of the check is less than or equal to the current account balance, then cash the check; otherwise, tell the person that the account is overdrawn.

- If $x$ is not equal to 0, then set $y$ equal to $1/x$; otherwise, print an error message that says we cannot divide by 0.
The Definition of Computer Science (continued)

- Iterative operations
  - Tell us to go back and repeat the execution of a previous block of instructions
  - Examples
    - Repeat the previous two operations until the mixture has thickened
    - While there are still more checks to be processed, do the following five steps
    - Repeat steps 1, 2, and 3 until the value of y is equal to 11
The Definition of Computer Science (continued)

- If we can specify an algorithm to solve a problem, we can automate its solution.

- Computing agent:
  - The machine, robot, person, or thing carrying out the steps of the algorithm.
  - Does not need to understand the concepts or ideas underlying the solution.
The Formal Definition of an Algorithm

- **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

- **Unambiguous operation**
  - An operation that can be understood and carried out directly by the computing agent without needing to be further simplified or explained
The Formal Definition of an Algorithm (continued)

- A primitive operation (or a primitive) of the computing agent
  - Operation that is unambiguous for computing agent
  - Primitive operations of different individuals (or machines) vary
  - An algorithm must be composed entirely of primitives
- Effectively computable
  - Computational process exists that allows computing agent to complete that operation successfully
The Formal Definition of an Algorithm (continued)

- The result of the algorithm must be produced after the execution of a finite number of operations

  - Infinite loop
    - The algorithm has no provisions to terminate
    - A common error in the designing of algorithms
The Importance of Algorithmic Problem Solving

- Algorithmic solutions can be:
  - Encoded into some appropriate language
  - Given to a computing agent to execute

- The computing agent
  - Would mechanically follow these instructions and successfully complete the task specified
  - Would not have to understand
    - Creative processes that went into discovery of solution
    - Principles and concepts that underlie the problem
The Early Period: Up to 1940

- 3,000 years ago: Mathematics, logic, and numerical computation
  - Important contributions made by the Greeks, Egyptians, Babylonians, Indians, Chinese, and Persians
- 1614: Logarithms
  - Invented by John Napier to simplify difficult mathematical computations
- Around 1622: First slide rule created
1672: The Pascaline
- Designed and built by Blaise Pascal
- One of the first mechanical calculators
- Could do addition and subtraction

1674: Leibnitz’s Wheel
- Constructed by Gottfried Leibnitz
- Mechanical calculator
- Could do addition, subtraction, multiplication, and division
Figure 1.4
The Pascaline: One of the Earliest Mechanical Calculators
The Early Period: Up to 1940 (continued)

- 1801: The Jacquard loom
  - Developed by Joseph Jacquard
  - Automated loom
  - Used punched cards to create desired pattern

- 1823: The Difference Engine
  - Developed by Charles Babbage
  - Did addition, subtraction, multiplication, and division to 6 significant digits
  - Solved polynomial equations and other complex mathematical problems
The Early Period: Up to 1940 (continued)

- 1823: The Difference Engine
  - Developed by Charles Babbage
  - Capabilities:
    - Addition, subtraction, multiplication, and division to 6 significant digits
    - Solve polynomial equations and other complex mathematical problems
Figure 1.5
Drawing of the Jacquard Loom
The Early Period: Up to 1940 (continued)

- 1830s: The Analytic Engine
  - Designed by Charles Babbage
  - More powerful and general-purpose computational machine
  - Components were functionally similar to the four major components of today’s computers
    - Mill (modern terminology: arithmetic/logic unit)
    - Store (modern terminology: memory)
    - Operator (modern terminology: processor)
    - Output (modern terminology: input/output)
The Early Period: Up to 1940 (continued)

- 1890: U.S. census carried out with programmable card processing machines
  - Built by Herman Hollerith
  - These machines could automatically read, tally, and sort data entered on punched cards
The Birth of Computers: 1940–1950

- Development of electronic, general-purpose computers
  - Did not begin until after 1940
  - Was fueled in large part by needs of World War II
- Early computers
  - Mark I
  - ENIAC
  - ABC system
  - Colossus
  - Z1
Figure 1.6
Photograph of the ENIAC Computer
The Birth of Computers: 
1940–1950

- Stored program computer model
  - Proposed by John Von Neumann in 1946
  - Stored binary algorithm in the computer’s memory along with the data
  - Is known as the Von Neumann architecture
  - Modern computers remain, fundamentally, Von Neumann machines
- First stored program computers
  - EDVAC
  - EDSAC
The Modern Era: 1950 to the Present

- First generation of computing (1950-1959)
  - Used vacuum tubes to store data and programs
  - Each computer was multiple rooms in size
  - Computers were not very reliable
The Modern Era: 1950 to the Present (continued)

- Second generation of computing (1959-1965)
  - Replaced vacuum tubes by transistors and magnetic cores
  - Dramatic reduction in size
    - Computer could fit into a single room
  - Increase in reliability of computers
  - Reduced costs of computers
  - High-level programming languages
    - The programmer occupation was born
The Modern Era: 1950 to the Present (continued)

- Third generation of computing (1965-1975)
  - Used integrated circuits rather than individual electronic components
  - Further reduction in size and cost of computers
    - Computers became desk-sized
    - First minicomputer developed
  - Software industry formed
The Modern Era: 1950 to the Present (continued)

- Fourth generation of computing (1975-1985)
  - Reduced to the size of a typewriter
  - First microcomputer developed
  - Desktop and personal computers common
  - Appearance of
    - Computer networks
    - Electronic mail
    - User-friendly systems (Graphical user interfaces)
    - Embedded systems
Figure 1.7
The Altair 8800, the World’s First Microcomputer
The Modern Era: 1950 to the Present (continued)

- Fifth generation of computing (1985-?)
  - Recent developments
    - Massively parallel processors
    - Handheld devices and other types of personal digital assistants (PDAs)
    - High-resolution graphics
    - Powerful multimedia user interfaces incorporating sound, voice recognition, touch, photography, video, and television
The Modern Era: 1950 to the Present (continued)

- Recent developments (continued)
  - Integrated global telecommunications incorporating data, television, telephone, FAX, the Internet, and the World Wide Web
  - Wireless data communications
  - Massive storage devices
  - Ubiquitous computing
<table>
<thead>
<tr>
<th>Generation</th>
<th>Approximate Dates</th>
<th>Major Advances</th>
</tr>
</thead>
<tbody>
<tr>
<td>First</td>
<td>1950–1957</td>
<td>First commercial computers&lt;br&gt;First symbolic programming languages&lt;br&gt;Use of binary arithmetic, vacuum tubes for storage&lt;br&gt;Punched card input/output</td>
</tr>
<tr>
<td>Second</td>
<td>1957–1965</td>
<td>Transistors and core memories&lt;br&gt;First disks for mass storage&lt;br&gt;Size reduction, increased reliability, lower costs&lt;br&gt;First high-level programming languages&lt;br&gt;First operating systems</td>
</tr>
<tr>
<td>Third</td>
<td>1965–1975</td>
<td>Integrated circuits&lt;br&gt;Further reduction in size and cost, increased reliability&lt;br&gt;First minicomputers&lt;br&gt;Time-shared operating systems&lt;br&gt;Appearance of the software industry&lt;br&gt;First set of computing standards for compatibility between systems</td>
</tr>
</tbody>
</table>

Figure 1.8
Some of the Major Advancements in Computing
| Fourth  | 1975–1985 | Large-scale and very-large-scale integrated circuits  
|        |           | Further reduction in size and cost, increased reliability  
|        |           | First microcomputers  
|        |           | Growth of new types of software and of the software industry  
|        |           | Computer networks  
|        |           | Graphical user interfaces  
| Fifth   | 1985–?   | Ultra-large-scale integrated circuits  
|        |           | Supercomputers and parallel processors  
|        |           | Laptops and handheld computers  
|        |           | Wireless computing  
|        |           | Massive external data storage devices  
|        |           | Ubiquitous computing  
|        |           | High-resolution graphics, visualization, virtual reality  
|        |           | Worldwide networks  
|        |           | Multimedia user interfaces  

**Figure 1.8**
Some of the Major Advancements in Computing
Organization of the Text

- This book is divided into six separate sections called levels

- Each level addresses one aspect of the definition of computer science

- Computer science/Algorithms
Organization of the Text

- **Level 1: The Algorithmic Foundations of Computer Science**
  - Chapters 1, 2, 3

- **Level 2: The Hardware World**
  - Chapters 4, 5

- **Level 3: The Virtual Machine**
  - Chapters 6, 7
Organization of the Text

- Level 4: The Software World
  - Chapters 8, 9, 10, 11
- Level 5: Applications
  - Chapters 12, 13, 14
- Level 6: Social Issues
  - Chapter 15
Figure 1.9
Organization of the Text into a Six-Layer Hierarchy
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.