Chapter 1: An Introduction to Computer Science

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

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

(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

(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

Objectives

In this chapter, you will learn about:

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

- Categorizes 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

The Definition of Computer Science (continued)

- 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
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

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

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 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

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 (continued)

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

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

1823: 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

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

- 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

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

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

Figure 1.7
The Altair 8800, the World’s First Microcomputer

Figure 1.8
Some of the Major Advancements in Computing
Figure 1.8  Some of the Major Advancements in Computing

Fifth  1985–7
- Ultra-large-scale integrated circuits
- Supercomputers and parallel processors
- Laptops and handheld computers
- Massive external data storage devices
- Multius programming languages
- High-resolution graphics, visualization, virtual reality
- Worldwide networks
- Multimedia user interfaces

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
- Level 4: The Software World
  - Chapters 8, 9, 10, 11
- Level 5: Applications
  - Chapters 12, 13, 14
- Level 6: Social Issues
  - Chapter 15

Organization of the Text

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