Chapter 1
An Introduction to 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

• Abu Ja’far Muhammad ibn Musa Al-Khowarizmi (AD 780-850?), Persian Author
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, \( a_{m-1} a_{m-2} \ldots a_0 \) and \( b_{m-1} b_{m-2} \ldots b_0 \).

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

*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, \( c_m c_{m-1} c_{m-2} \ldots c_0 \).

**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

Shampooing instructions:
  – STEP 1 Wet hair
  – STEP 2 Lather
  – STEP 3 Rinse
  – STEP 4 Repeat
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

**Finding 100\textsuperscript{th} Prime Number?**

• STEP 1 Generate a list $L$ of all the prime numbers: $L_1, L_2, L_3, \ldots$
• STEP 2 Sort the list $L$ in ascending order
• STEP 3 Print out the 100th element in the list, $L_{100}$
• STEP 4 Stop
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
<table>
<thead>
<tr>
<th>Step</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Wet your hair</td>
</tr>
<tr>
<td>2</td>
<td>Set the value of WashCount to 0</td>
</tr>
<tr>
<td>3</td>
<td>Repeat Steps 4 through 6 until the value of WashCount equals 2</td>
</tr>
<tr>
<td>4</td>
<td>Lather your hair</td>
</tr>
<tr>
<td>5</td>
<td>Rinse your hair</td>
</tr>
<tr>
<td>6</td>
<td>Add 1 to the value of WashCount</td>
</tr>
<tr>
<td>7</td>
<td>Stop, you have finished shampooing your hair</td>
</tr>
</tbody>
</table>

A correct solution to the shampooing problem
### FIGURE 1.4

<table>
<thead>
<tr>
<th>Step</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Wet your hair</td>
</tr>
<tr>
<td>2</td>
<td>Lather your hair</td>
</tr>
<tr>
<td>3</td>
<td>Rinse your hair</td>
</tr>
<tr>
<td>4</td>
<td>Lather your hair</td>
</tr>
<tr>
<td>5</td>
<td>Rinse your hair</td>
</tr>
<tr>
<td>6</td>
<td>Stop, you have finished shampooing your hair</td>
</tr>
</tbody>
</table>

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
Quick Quiz 1

• 1. Which kind of operation is “Add water until the cup is full”?
• 2. (True or false) All algorithms are known, computer scientists simply select the correct algorithm for each new problem.
• 3. Operations that a given computing agent can perform are called ______________.
• 4. List at least two flaws in the “algorithm” below.
  • Given a jar full of jelly beans,
  • Pick a jelly bean from the jar
  • Add one to the total count
  • Repeat until the jar is empty
Food for Thought - Practice Problems

Get a copy of the instructions that describe how to do the following and decide if they are algorithms:

- Register for classes at the beginning of the semester.
- Use the online computer catalog to see what is available in the college library on a given subject.
- Use the copying machine in your building.
- Log on to the World Wide Web.
- Add someone as a friend to your Facebook account.
• Seventeenth century: automation/simplification of arithmetic for scientific research
  – John Napier invented logarithms as a way to simplify difficult mathematical computations (1614)
  – The first slide rule appeared around 1622
  – Blaise Pascal designed and built a mechanical calculator named the Pascaline (1672)
  – Gottfried Leibnitz constructed a mechanical calculator called Leibnitz’s Wheel (1674)
The Pascaline, one of the earliest mechanical calculators

Source: Computer History Museum
A Brief History of Computing
The Early Period: Up to 1940 (continued)

• Seventeenth century devices
  – Could represent numbers
  – Could perform arithmetic operations on numbers
  – Did not have a memory to store information
  – Were not programmable (a user could not provide a sequence of actions to be executed by the device)
A Brief History of Computing

The Early Period: Up to 1940 (continued)

• Nineteenth century devices
  – Joseph Jacquard designed an automated loom that
    used punched cards to create patterns (1801)
  – Herman Hollerith (1880s on)
    • Designed programmable card-processing machines
      to read, tally, and sort data on punched cards for the
      U.S. Census Bureau
    • Founded company that became IBM in 1924
      – Computer Tabulating Recording Company -> IBM
A Brief History of Computing
The Early Period: Up to 1940 (continued)

• Charles Babbage
  – Difference Engine designed and built in 1823
    • Could do addition, subtraction, multiplication, and division to six significant digits
    • Could solve polynomial equations and other complex mathematical problems

  – Analytical Engine, designed but never built
    • Mechanical, programmable machine similar to a modern computer
A Brief History of Computing
The Early Period: Up to 1940 (continued)

_Babbage’s Term_ Modern Terminology

- **mill** → arithmetic/logic unit
- **store** → memory
- **operator** → processor
- **output unit** → input/output

Ada Augusta Byron – First programmer
A Brief History of Computing
The Early Period: Up to 1940 (continued)

• Nineteenth century devices
  – Were mechanical, not electrical
  – Had many features of modern computers:
    • Representation of numbers or other data
    • Operations to manipulate the data
    • Memory to store values in a machine-readable form
    • Programmable: sequences of instructions could be pre-designed for complex operations
A Brief History of Computing
The Birth of Computers: 1940–1950

• ABC system (Atanasoff-Berry Computer) (1942)
• Mark I (1944)
  – Electromechanical computer used a mix of relays, magnets, and gears to process and store data (binary, memory 72, * 4 s)
• Colossus (1943)
  – General-purpose computer built by Alan Turing for British Enigma project - German Enigma code
• ENIAC (Electronic Numerical Integrator and Calculator) (1946) - Eckert and Mauchly
  – First publicly known fully electronic computer
  – Firing tables, 18k tubes, 100X10’, 30 ton, * 4 ms)
Photograph of the ENIAC computer
A Brief History of Computing
The Birth of Computers: 1940–1950 (continued)

• John Von Neumann
  – Proposed a radically different computer design based on a model called the *stored program computer*
  – Research group at the University of Pennsylvania built one of the first stored program computers, called EDVAC, in 1951
  – UNIVAC-1, a version of EDVAC, *first commercially-sold computer* – Echert/ Mauckley
  – Virtually all modern computers use the *Von Neumann architecture*
A Brief History of Computing
The Modern Era: 1950 to the Present

• First generation of computing (1950-1957)
  – Similar to EDVAC
  – Vacuum tubes for processing and storage
  – Large, expensive, and delicate
  – Required trained users and special environments

• Second generation (1957–1965)
  – Transistors and magnetic cores instead of vacuum tubes
  – Era of FORTRAN and COBOL: high-level programming languages
  – The occupation called programmer was born.
A Brief History of Computing
The Modern Era: 1950 to the Present (continued)

• Third generation (1965 to 1975)
  – Era of the integrated circuit
  – Birth of the first minicomputer: desk-sized, not room-sized computers – PDP-1 (DEC Corp)
  – Birth of the software industry

• Fourth generation (1975 to 1985)
  – The first microcomputers: desktop machines (Altair 8800 – 1975)
  – Development of widespread computer networks
  – Electronic mail, graphical user interfaces, and embedded systems
A Brief History of Computing
The Modern Era: 1950 to the Present (continued)

• Source: University of Hawai’i at Hilo Graphics Services
A Brief History of Computing
The Modern Era: 1950 to the Present (continued)

• Fifth generation (1985–?)
  – Massively parallel processors capable of quadrillions \(10^{15}\) of computations per second
    • Non-Von-Neuman Architectures
  – Handheld digital devices
  – Powerful multimedia user interfaces incorporating sound, voice recognition, images, video, television
  – Wireless communications
  – Massive storage devices
  – Ubiquitous computing
## Organization of the Text

**Computer science is the study of algorithms including:**

1. Their formal and mathematical properties,
2. Their hardware realizations,
3. Their linguistic realizations,
4. Their applications.

**Levels of the text:**

1. **Level 1:** The Algorithmic Foundations of Computer Science
2. **Level 2:** The Hardware World
   - Level 3: The Virtual Machine
3. **Level 4:** The Software World
4. **Level 5:** Applications
5. **Level 6:** Social Issues
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