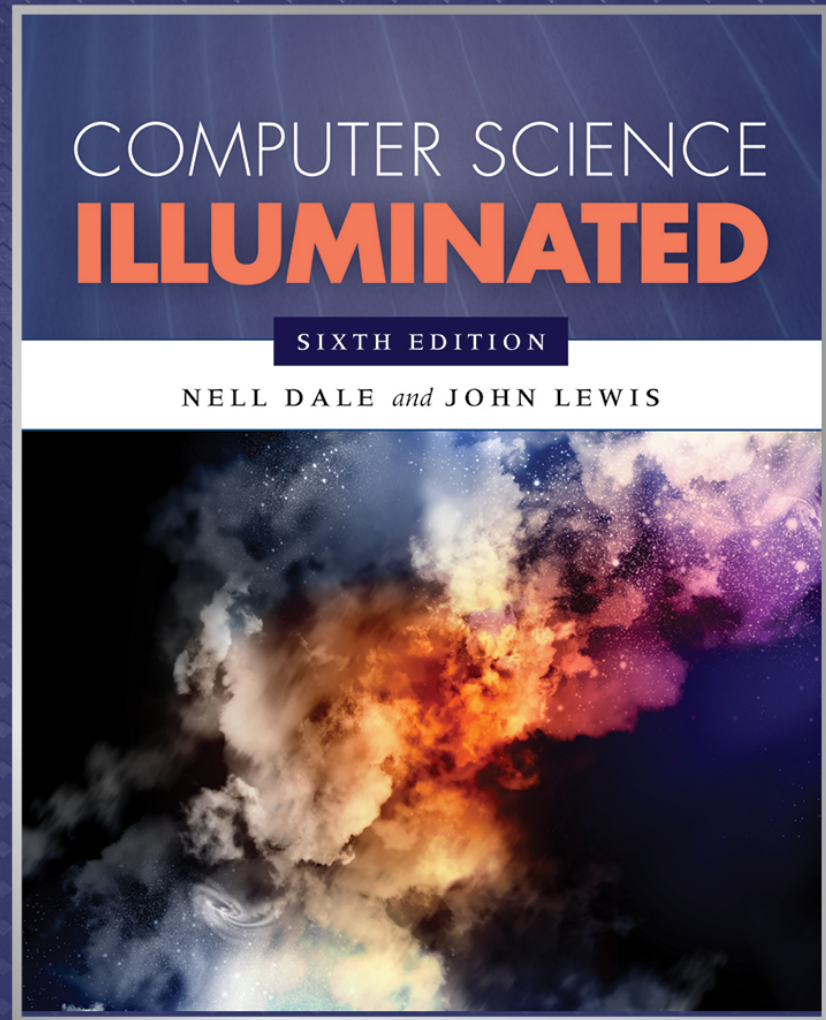


# Chapter 1

## The Big Picture





# Chapter Goals

- Describe the **layers** of a computer system
- Describe the concept of **abstraction** and its relationship to computing
- Describe the **history** of computer hardware and software
- Describe the **changing role** of the computer user
- Distinguish between **systems** programmers and **applications** programmers
- Distinguish between computing as a **tool** and computing as a **discipline**

# Computing Systems

**Computing systems are dynamic!**

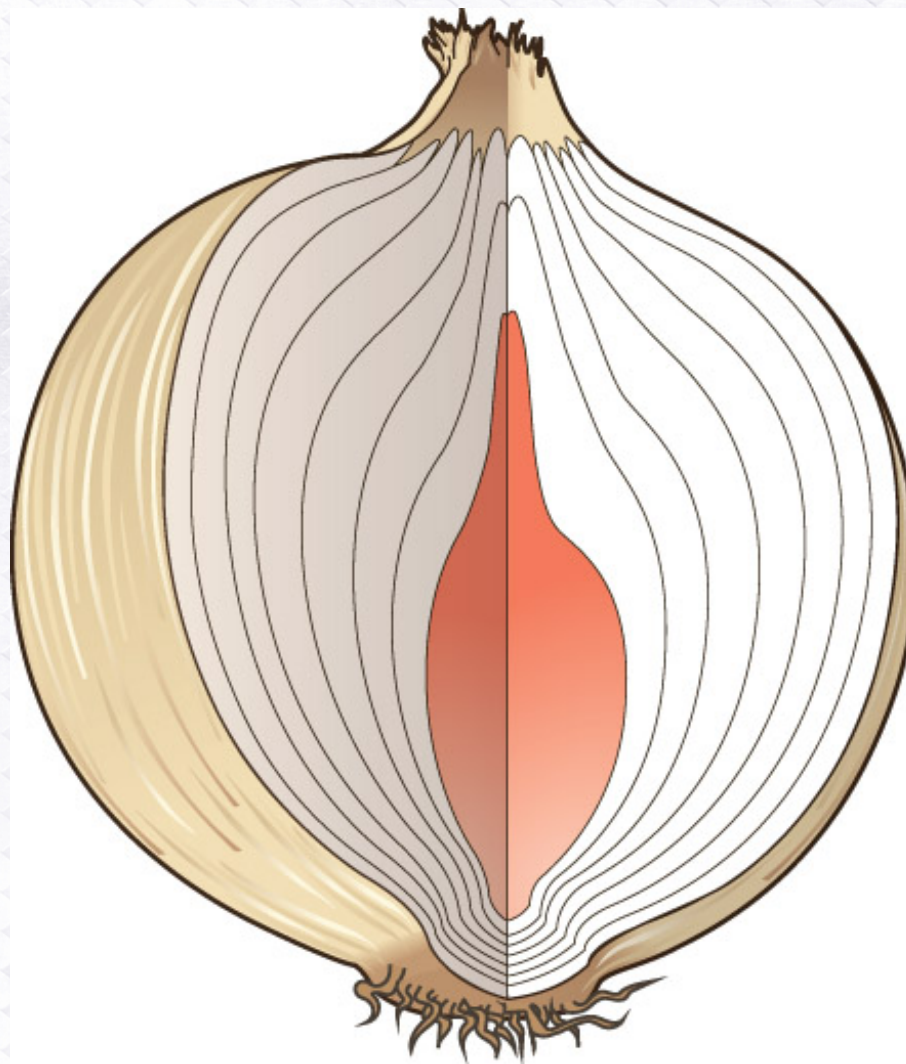
*What is the difference between **hardware** and **software**?*



# Computing Systems

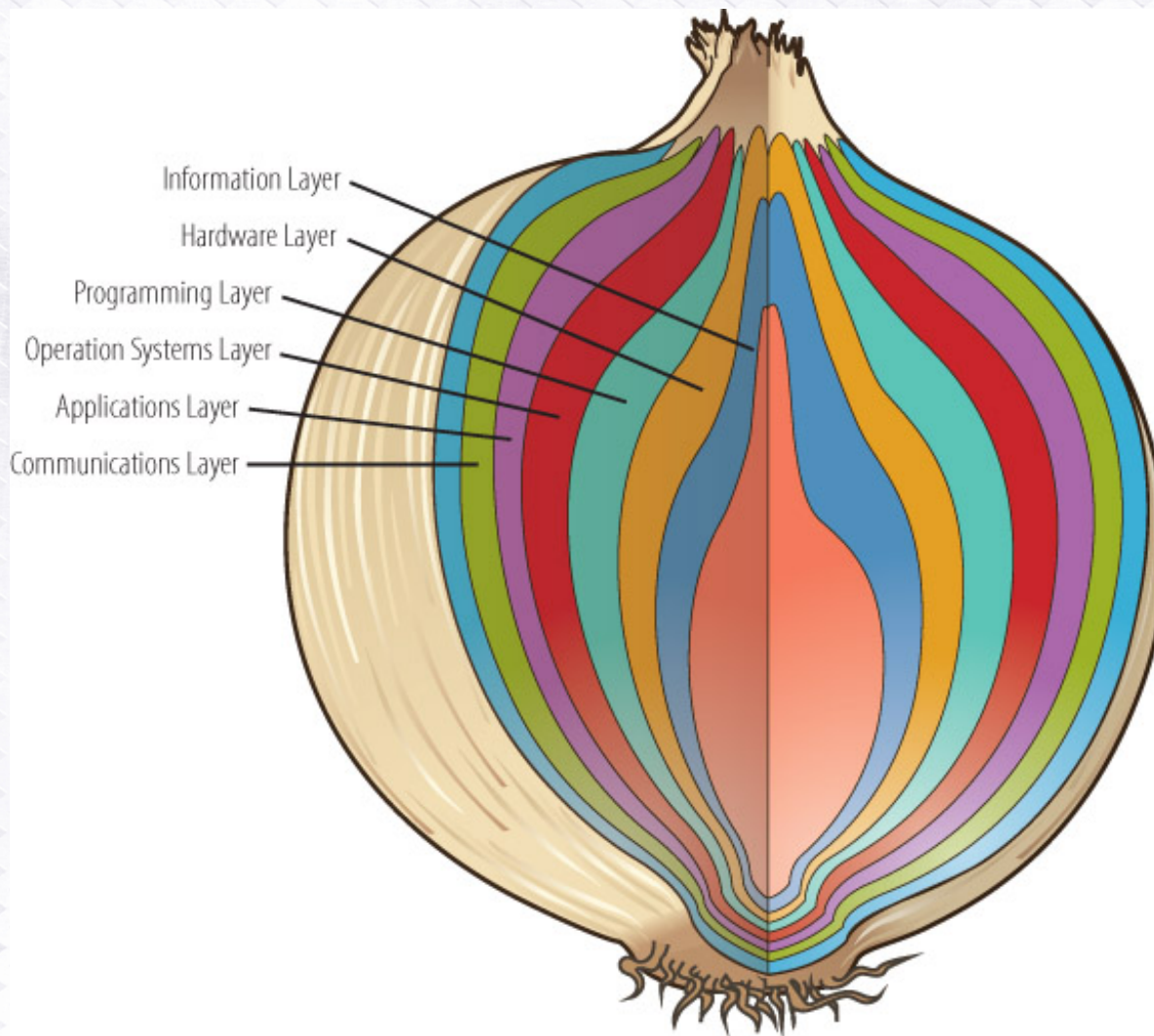
**Hardware** The physical elements of a computing system (printer, circuit boards, wires, keyboard...)

**Software** The programs that provide the instructions for a computer to execute



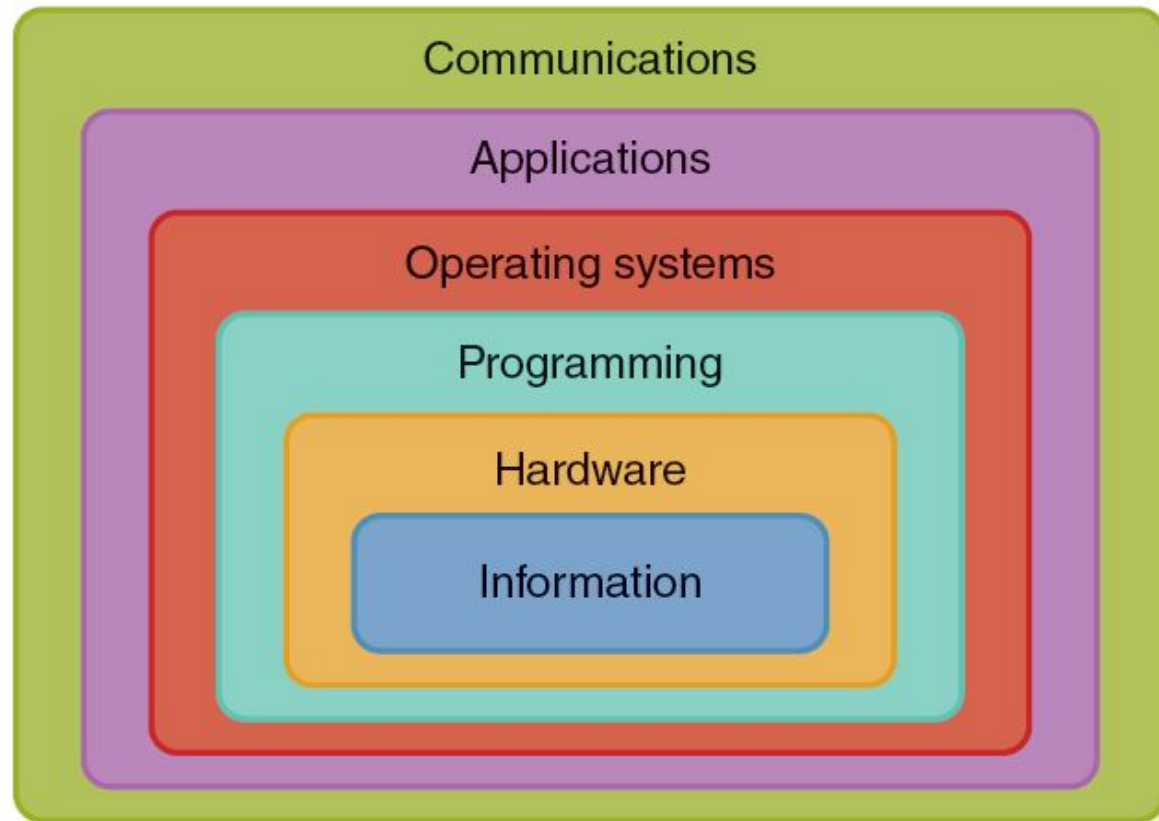
**FIGURE 01.UN02: Computer Science Illuminated**





**FIGURE 01.UN10: Computer Science Illuminated**

# Layers of a Computing System



**FIGURE 1.1** The layers of a computing system



# Abstraction

**Abstraction** A mental model that removes complex details

***This is a key concept. Abstraction will reappear throughout the text – be sure you understand it!***



# Internal and Abstract View



**FIGURE 1.2** A car engine and the abstraction that allows us to use it

© aospan/Shutterstock, Inc.; © Syda Productions/Shutterstock, Inc.

# History



© vencavolrab/iStock/Thinkstock



# Early History of Computing

## Abacus

An early device to record numeric values

## Blaise Pascal

Mechanical device to add, subtract, divide & multiply

## Joseph Jacquard

Jacquard's Loom, the punched card

## Charles Babbage

Analytical Engine



## **BOX 01.BOXUN02: Ada Lovelace, the First Programmer?**

**Courtesy of U.S. Army.**

© Eky Studio/Shutterstock, Inc. Copyright © 2016 by Jones & Bartlett Learning, LLC an Ascend Learning Company  
[www.jblearning.com](http://www.jblearning.com)



# Early History of Computing

## Ada Lovelace

First Programmer, the loop

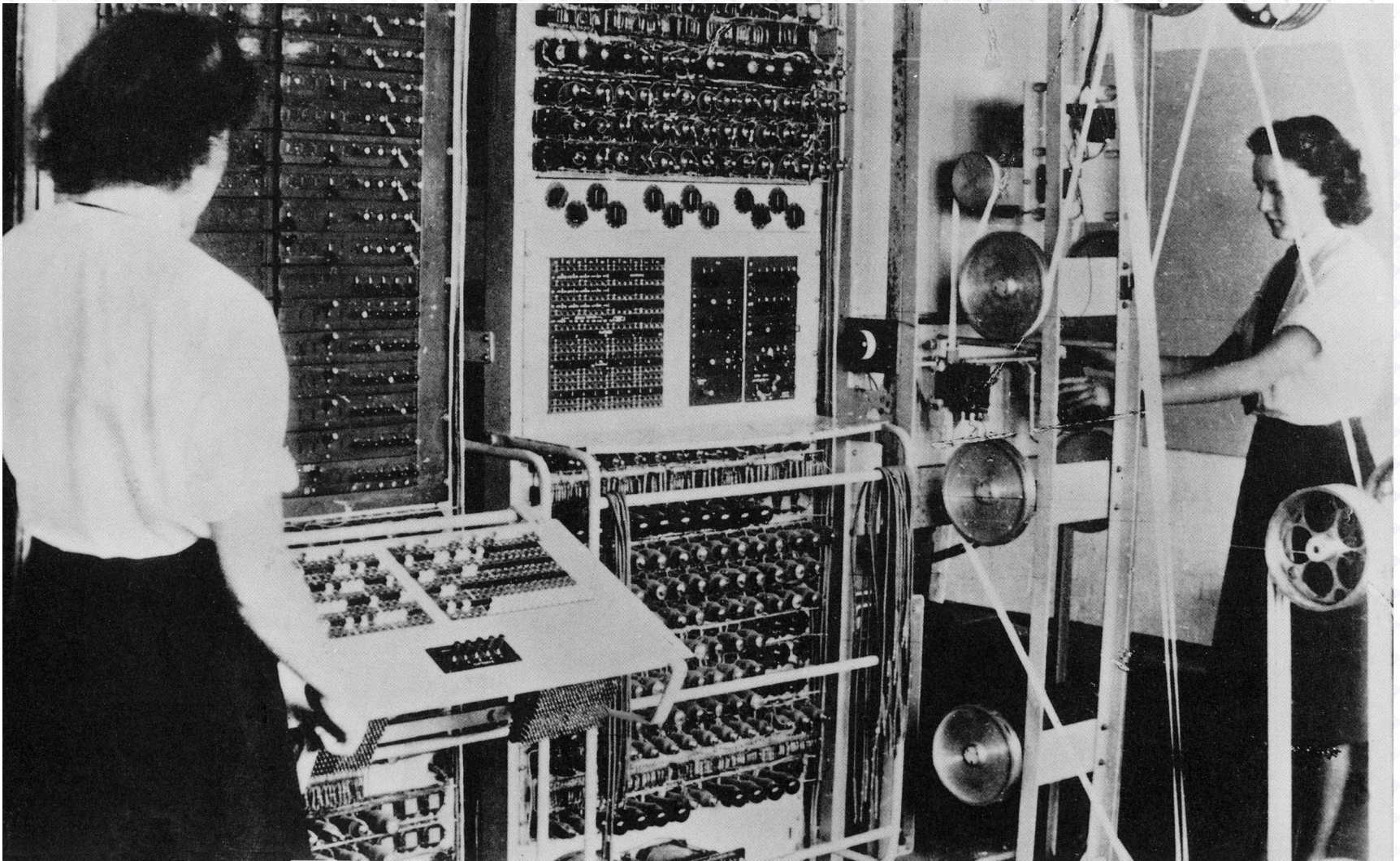
## Alan Turing

Turing Machine, Artificial Intelligence Testing

## Harvard Mark I, ENIAC, UNIVAC I

Early computers launch new era in mathematics, physics, engineering and economics





**FIGURE 01.F04: The Colossus, the first all-programmable digital computer**



# First Generation Hardware (1951-1959)

## Vacuum Tubes

Large, not very reliable, generated a lot of heat

## Magnetic Drum

Memory device that rotated under a read/write head

## Card Readers → Magnetic Tape Drives

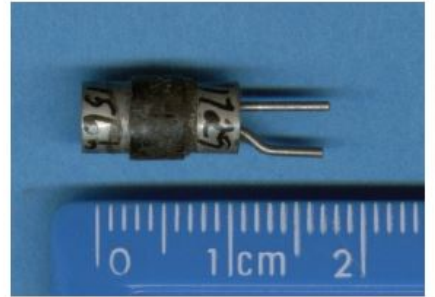
Sequential auxiliary storage devices



# Second Generation Hardware (1959-1965)

## Transistor

Replaced vacuum tube, fast, small, durable, cheap



**FIGURE 1.7** A transistor, which replaced the vacuum tube

Courtesy of Dr. Andrew Wylie

## Magnetic Cores

Replaced magnetic drums, information available instantly

## Magnetic Disks

Replaced magnetic tape, data can be accessed directly



# Third Generation Hardware (1965-1971)

## Integrated Circuits

Replaced circuit boards, smaller, cheaper, faster, more reliable

## Transistors

Now used for memory construction

## Terminal

An input/output device with a keyboard and screen

# Fourth Generation Hardware (1971-?)

## Large-scale Integration

Great advances in chip technology

## PCs, the Commercial Market, Workstations

Personal Computers and Workstations emerge

New companies emerge: Apple, Sun, Dell ...

## Laptops, Tablet Computers, and Smart Phones

Everyone has his/her own portable computer



# Parallel Computing and Networking

## Parallel Computing

Computers rely on interconnected central processing and/or memory units that increase processing speed

## Networking

Ethernet connects small computers to share resources  
File servers connect PCs in the late 1980s

## ARPANET and LANs → Internet

# What's Next?

## Quantum Computing

Utilising Quantum Mechanics to develop a computer which used quantum-bits (qubits)

## Qubits

Replacing traditional bits which can either be 0 or 1 a qubit can represent both a zero and a one at the same time!

(Schrödinger's cat)



# SCHRODINGER'S CAT IS A • L • I • V • E



# First Generation Software (1951-1959)

## Machine Language

Computer programs written in binary (1s and 0s)

## Assembly Languages and Translators

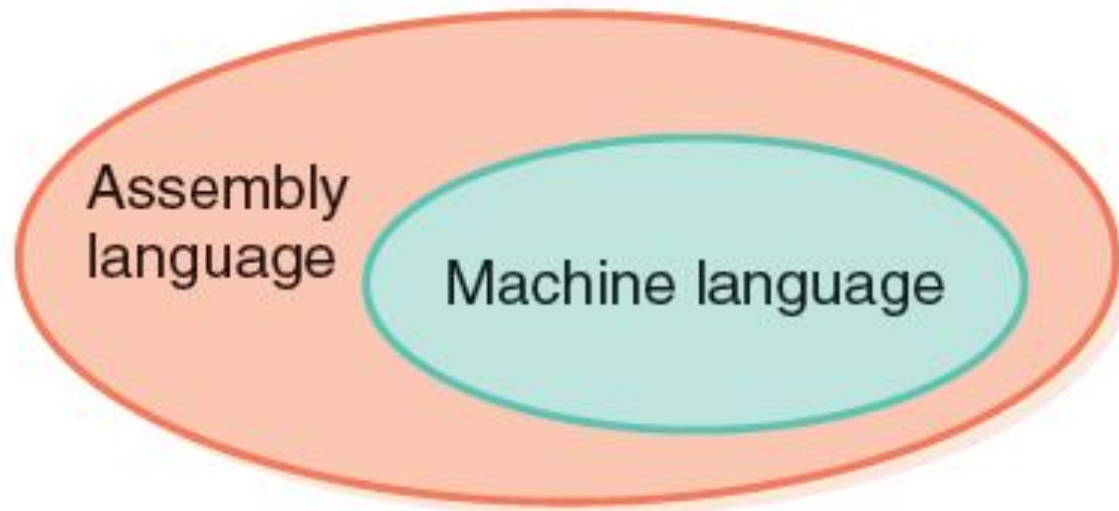
Programs written using mnemonics, which were translated into machine language

## Programmer Changes

Programmers divide into two groups: application programmers and systems programmers



# Assembly/Machine

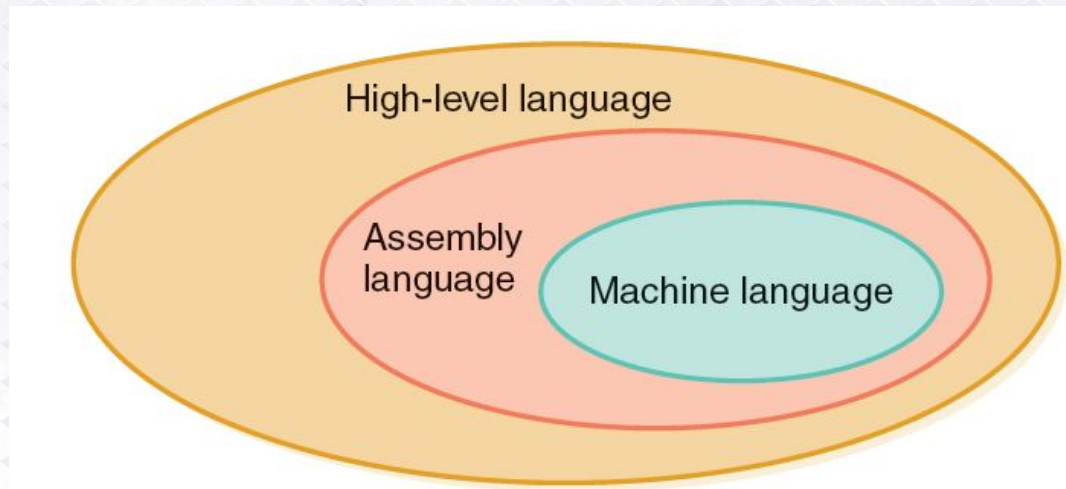


**FIGURE 1.8** Layers of languages at the end of the first generation

# Second Generation Software (1959-1965)

## High-level Languages

English-like statements made programming easier:  
Fortran, COBOL, Lisp



**FIGURE 1.9** Layers of language at the end of the second generation



# Third Generation Software (1965-1971)

## Systems Software

Utility programs

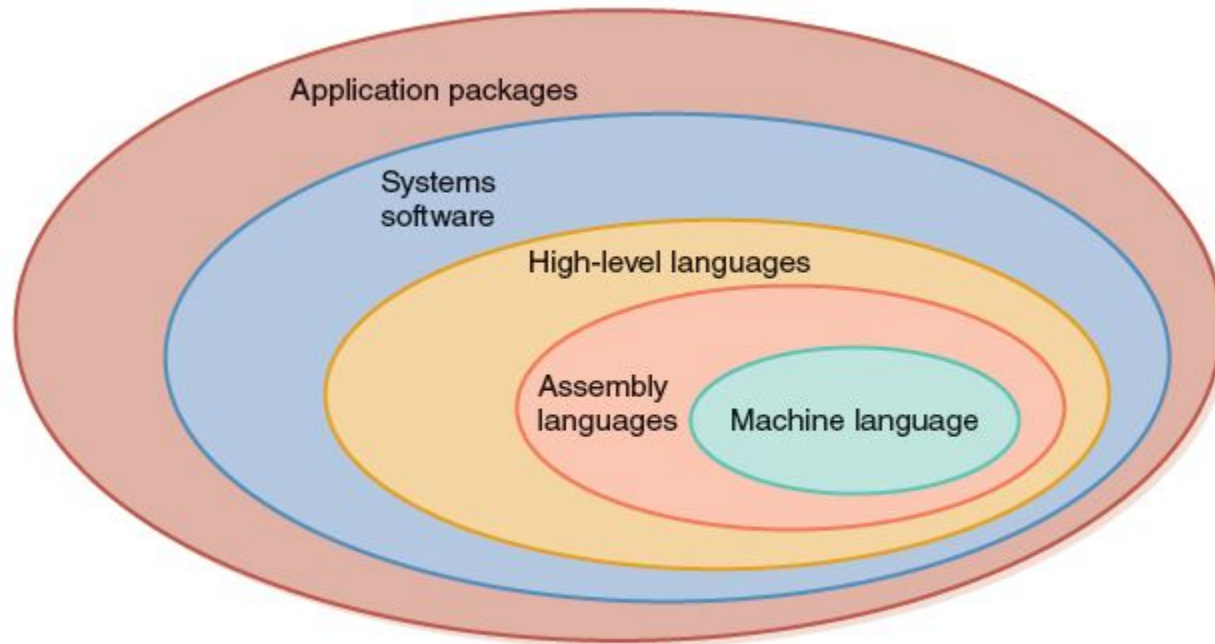
Language translators

Operating system, which decides which programs to run and when

## Separation between Users and Hardware

Computer programmers write programs to be used by general public (i.e., nonprogrammers)

# Third Generation Software (1965-1971)



**FIGURE 1.10** The layers of software surrounding the hardware continue to grow



# Fourth Generation Software (1971-1989)

## Structured Programming

Pascal

C++

## New Application Software for Users

Spreadsheets

Word processors

Database management systems

# **Fifth Generation Software (1990- present)**

## **Microsoft**

Windows operating system and other Microsoft application programs dominate the market

## **Object-Oriented Design**

Based on a hierarchy of data objects (i.e. Java)

## **World Wide Web**

Allows easy global communication through the Internet

## **New Users**

Today's user needs no computer knowledge



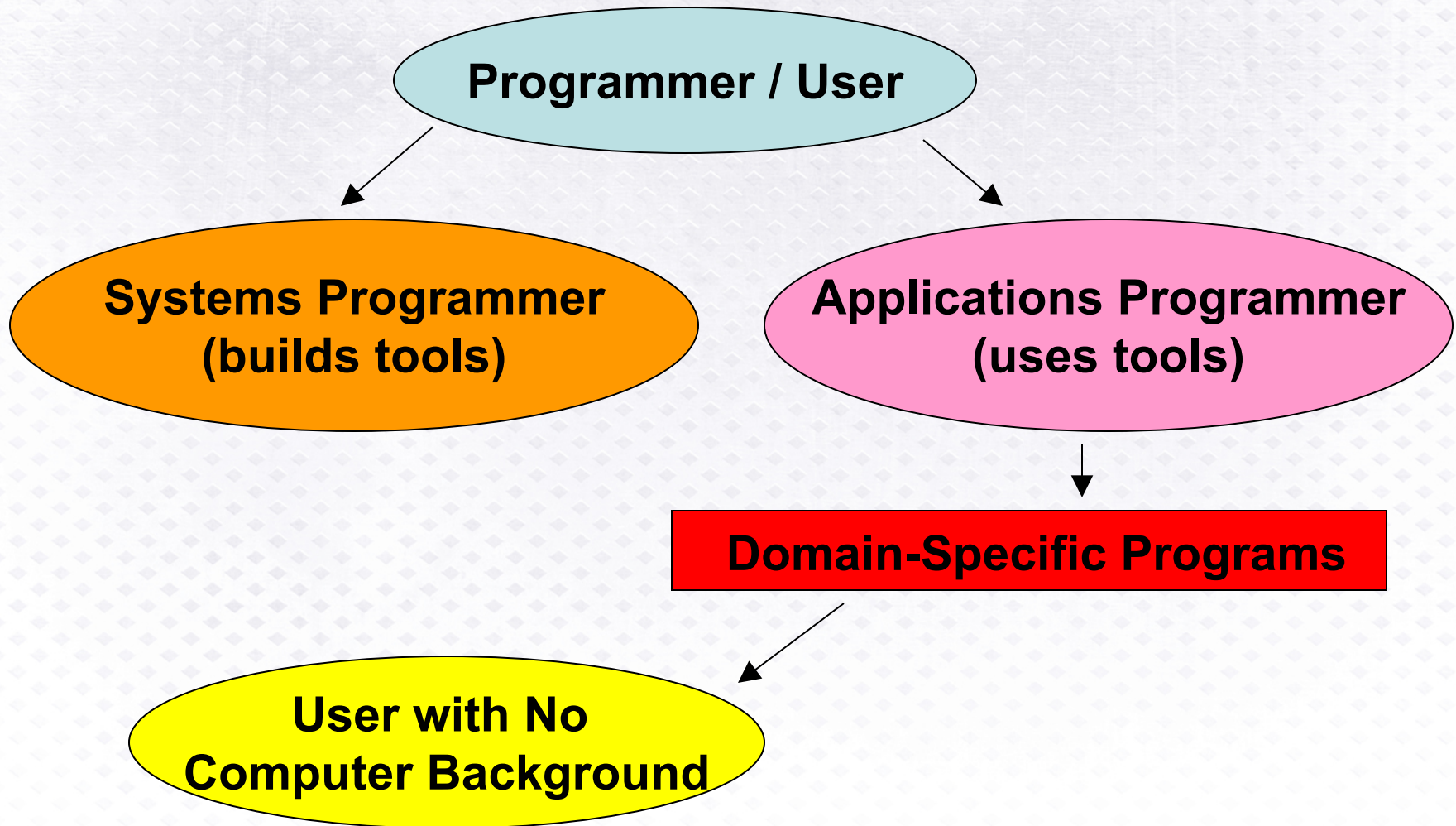
# Predictions

“I think there is a world market for maybe five computers” – Thomas Watson, chair of IBM, 1943.

“There is no reason anyone would want a computer in their home” – Ken Olsen, 1977.

“I predict that the Internet ... will go spectacularly supernova and in 1996 catastrophically collapse” – Bob Metcalfe, 1995.

# Computing as a Tool





# Computing as a Discipline

***What can be (efficiently) automated?***

## **Four Necessary Skills**

- Algorithmic Thinking
- Representation
- Programming
- Design

# Computing as a Discipline

***What do you think?***

*Is Computer Science a mathematical, scientific, or engineering discipline?*



# Examples of Systems Areas

- Algorithms and Data Structures
- Programming Languages
- Architecture
- Operating Systems
- Software Engineering
- Human-Computer Communication

# Examples of Application Areas

- Numerical and Symbolic Computation
- Databases and Information Retrieval
- Intelligent Systems
- Graphics and Visual Computing
- Net-Centric Computing
- Computational Science



# Who am I?



*Can you list three items on my resume?*

# Ethical Issues

## Digital Divide

*What disparity does the term “digital divide” describe?*

*What is the E-Rate program?*

*What is the One-Laptop-Per-Child program?*

*What additional challenge must developing nations face?*



# ***Do you know?***

?

*Who said the Analytical Engine “weaves algebraic patterns”?*

*Where did scientists build the first nanotube computer?*

*What computer company was launched in a garage?*

*What companies turned Jobs and Wozniak away?*

*Where is the Computer History Museum?*

*What is Room to Read?*

*When and where were the first CS Departments formed?*