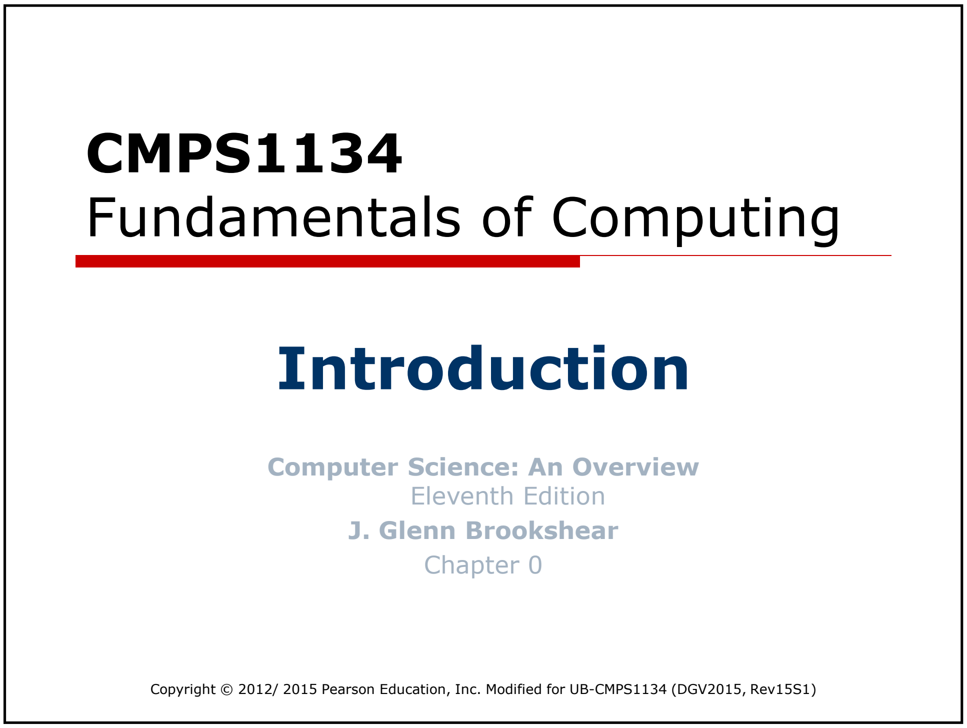
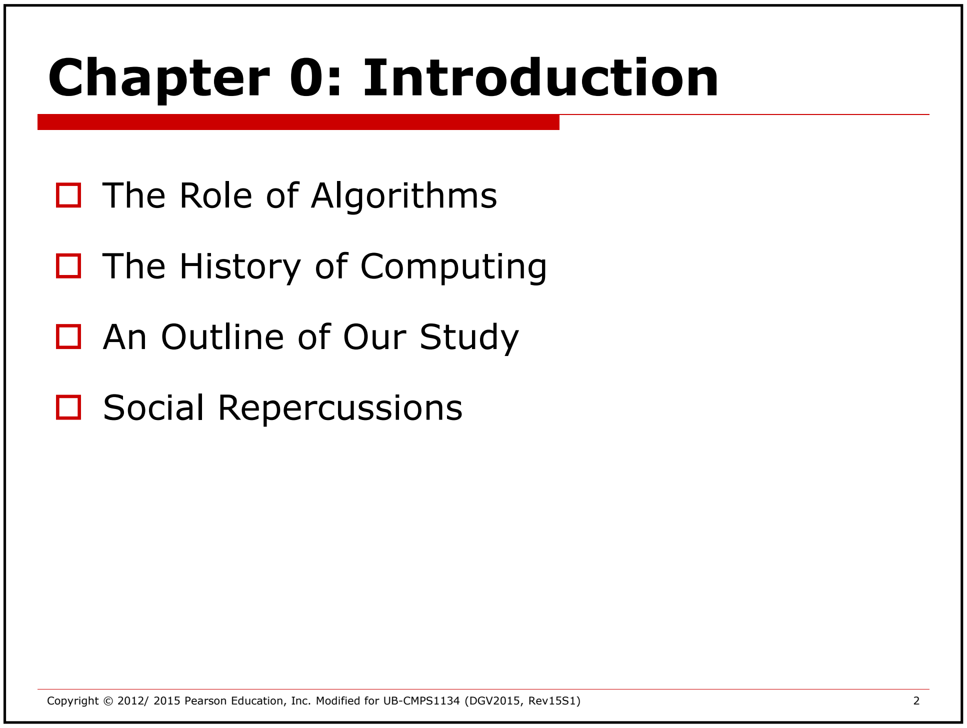


CMPS1134
Fundamentals of Computing

Introduction

Computer Science: An Overview
Eleventh Edition
J. Glenn Brookshear
Chapter 0



Chapter 0: Introduction

- □ The Role of Algorithms
- □ The History of Computing
- □ An Outline of Our Study
- □ Social Repercussions

The Role of Algorithms

- □ **Algorithm:** A set of steps that defines how a task is performed
- □ **Program:** A representation of an algorithm
- □ **Programming:** The process of developing a program
- □ **Software:** Programs and algorithms
- □ **Hardware:** Equipment

Figure 0.1 **An algorithm for a magic trick**

Effect: The performer places some cards from a normal deck of playing cards face down on a table and mixes them thoroughly while spreading them out on the table. Then, as the audience requests either red or black cards, the performer turns over cards of the requested color.

Secret and Patter:

Step 1. From a normal deck of cards, select ten red cards and ten black cards. Deal these cards face up in two piles on the table according to color.

Step 2. Announce that you have selected some red cards and some black cards.

Step 3. Pick up the red cards. Under the pretense of aligning them into a small deck, hold them face down in your left hand and, with the thumb and first finger of your right hand, pull back on each end of the deck so that each card is given a slightly *backward* curve. Then place the deck of red cards face down on the table as you say, "Here are the red cards in this stack."

Step 4. Pick up the black cards. In a manner similar to that in step 3, give these cards a slight *forward* curve. Then return these cards to the table in a face-down deck as you say, "And here are the black cards in this stack."

Step 5. Immediately after returning the black cards to the table, use both hands to mix the red and black cards (still face down) as you spread them out on the tabletop. Explain that you are thoroughly mixing the cards.

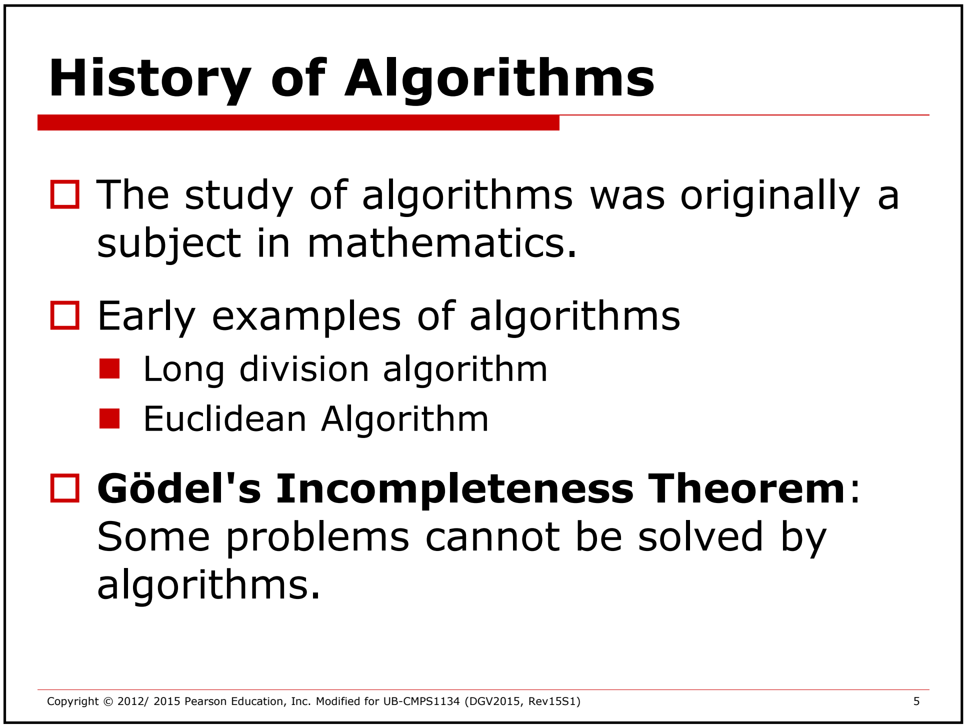
Step 6. As long as there are face-down cards on the table, repeatedly execute the following steps:

6.1. Ask the audience to request either a red or a black card.

6.2. If the color requested is red and there is a face-down card with a concave appearance, turn over such a card while saying, "Here is a red card."

6.3. If the color requested is black and there is a face-down card with a convex appearance, turn over such a card while saying, "Here is a black card."

6.4. Otherwise, state that there are no more cards of the requested color and turn over the remaining cards to prove your claim.



History of Algorithms

- The study of algorithms was originally a subject in mathematics.
- Early examples of algorithms
  - Long division algorithm
  - Euclidean Algorithm
- **Gödel's Incompleteness Theorem**: Some problems cannot be solved by algorithms.

5

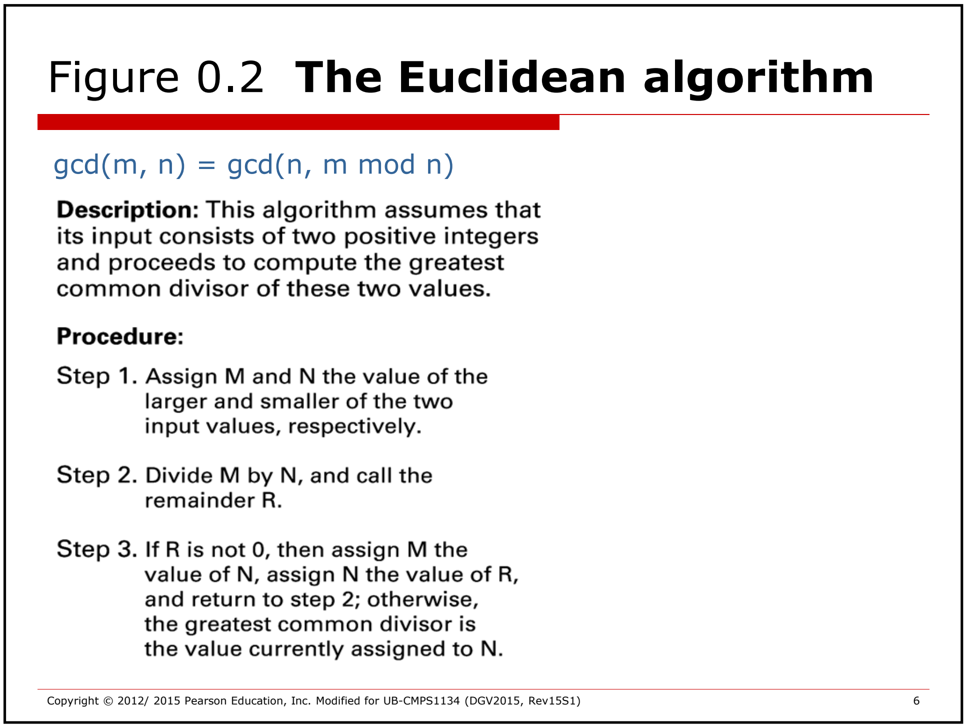


Figure 0.2 **The Euclidean algorithm**

gcd(m, n) = gcd(n, m mod n)

Description: This algorithm assumes that its input consists of two positive integers and proceeds to compute the greatest common divisor of these two values.

Procedure:

Step 1. Assign M and N the value of the larger and smaller of the two input values, respectively.

Step 2. Divide M by N, and call the remainder R.

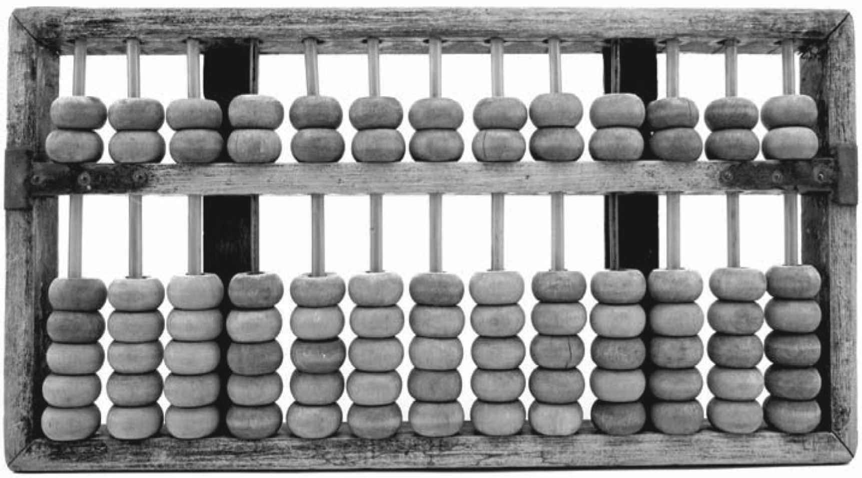
Step 3. If R is not 0, then assign M the value of N, assign N the value of R, and return to step 2; otherwise, the greatest common divisor is the value currently assigned to N.

6

The History of Computing

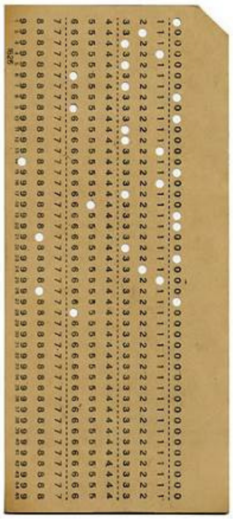
- Early computing devices
  - Abacus: positions of beads represent numbers
  - Gear-based machines (1600s-1800s)
    - Positions of gears represent numbers
    - Blaise Pascal, Wilhelm Leibniz, Charles Babbage (production was not financially feasible because of technology of the time)

Figure 0.3 **Chinese Wooden Abacus**



Early Data Storage

- □ Punched cards
  - First used in Jacquard Loom (1801) to store patterns for weaving cloth
  - Storage of programs in Babbage’s Analytical Engine
  - Popular through the 1970’s
- □ Gear positions
  - Gears set for initial data
  - Output is final gear position



Early Computers

- □ Based on mechanical relays
  - 1940: Stibitz at Bell Laboratories
  - 1944: Mark I: Howard Aiken and IBM at Harvard
- □ Based on vacuum tubes
  - 1937-1941: Atanasoff-Berry at Iowa State
  - 1940s: Colossus: secret German code-breaker (do not confuse with Turing’s Bombe)
  - 1940s: ENIAC (Electronic Numerical Integrator and Computer): Mauchly & Eckert at U. of Penn.

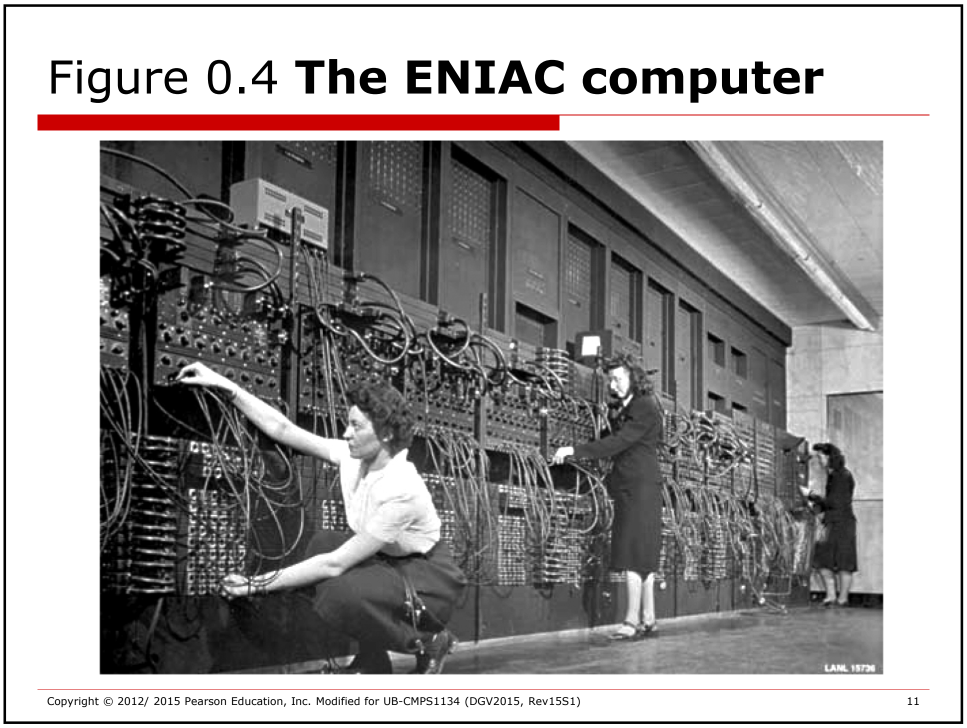
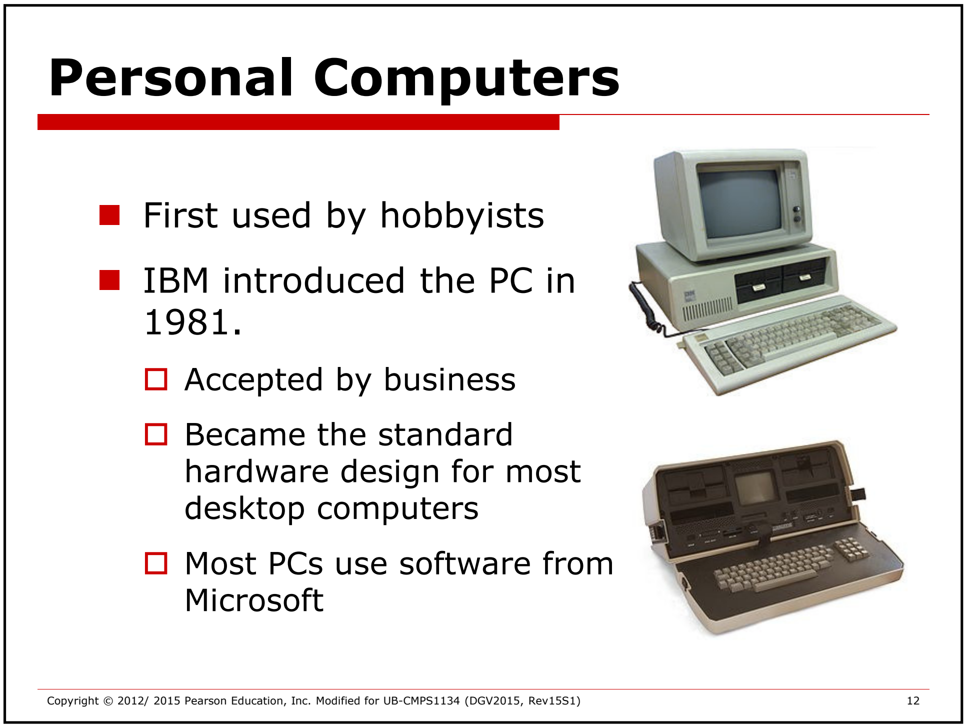
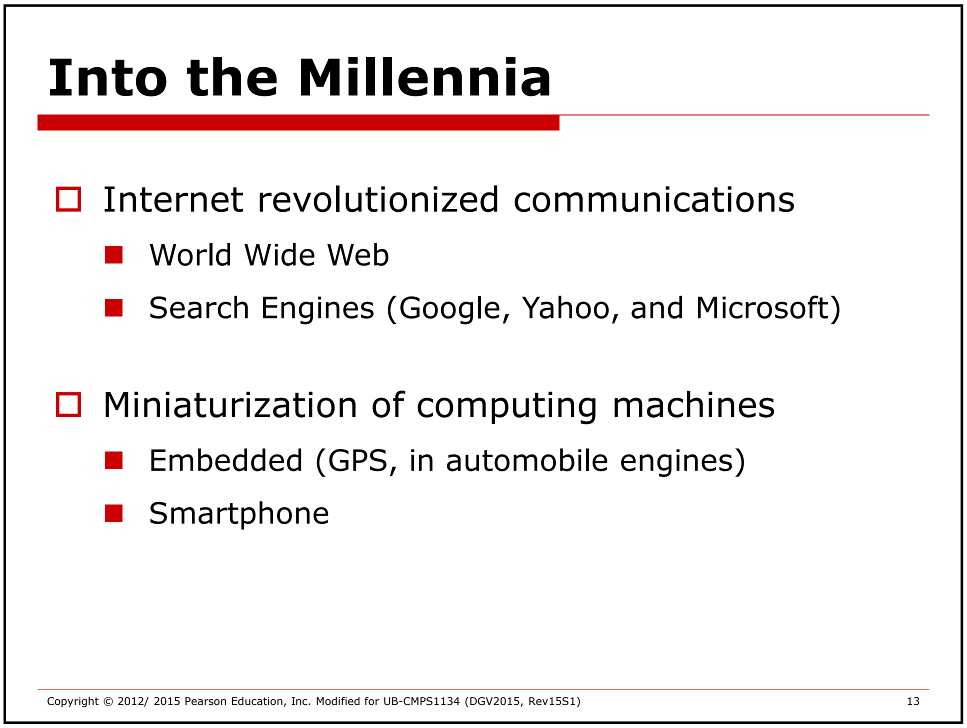


Figure 0.4 **The ENIAC computer**



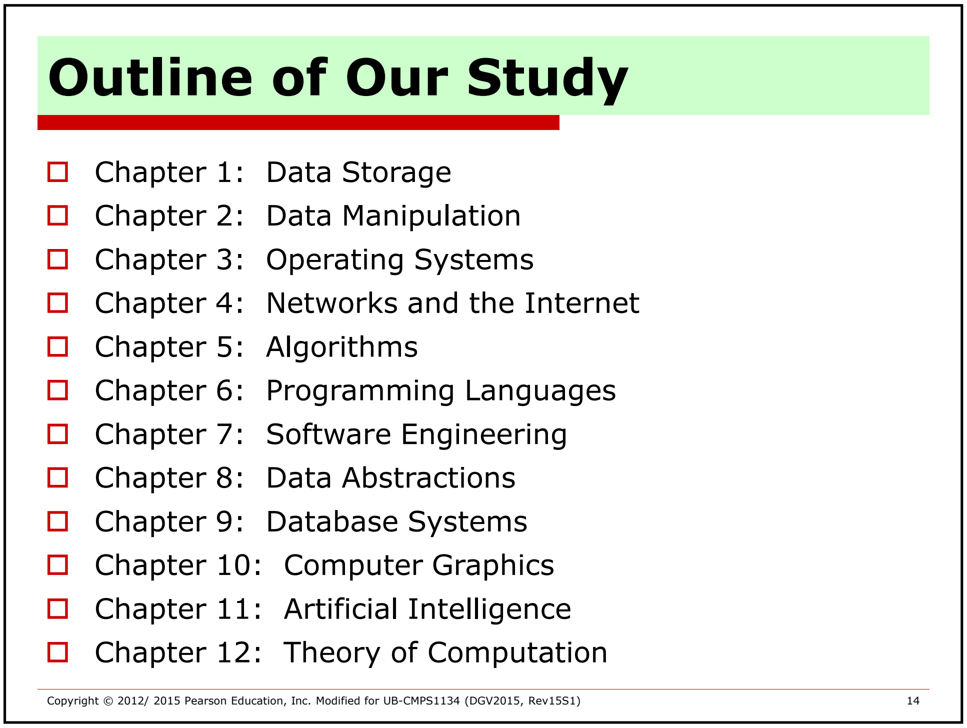
Personal Computers

- First used by hobbyists
- IBM introduced the PC in 1981.
  - □ Accepted by business
  - □ Became the standard hardware design for most desktop computers
  - □ Most PCs use software from Microsoft



Into the Millennia

- □ Internet revolutionized communications
  - World Wide Web
  - Search Engines (Google, Yahoo, and Microsoft)
- □ Miniaturization of computing machines
  - Embedded (GPS, in automobile engines)
  - Smartphone



Outline of Our Study

- □ Chapter 1: Data Storage
- □ Chapter 2: Data Manipulation
- □ Chapter 3: Operating Systems
- □ Chapter 4: Networks and the Internet
- □ Chapter 5: Algorithms
- □ Chapter 6: Programming Languages
- □ Chapter 7: Software Engineering
- □ Chapter 8: Data Abstractions
- □ Chapter 9: Database Systems
- □ Chapter 10: Computer Graphics
- □ Chapter 11: Artificial Intelligence
- □ Chapter 12: Theory of Computation

The Overarching Themes of Computer Science

□ Computing technology effects:

- Governments ability to exert control
- Global economics
- Scientific research
- Revolutionized the role of data collection, storage, and applications
- Provided new means of communication and interaction
- Repeatedly challenged the status quo

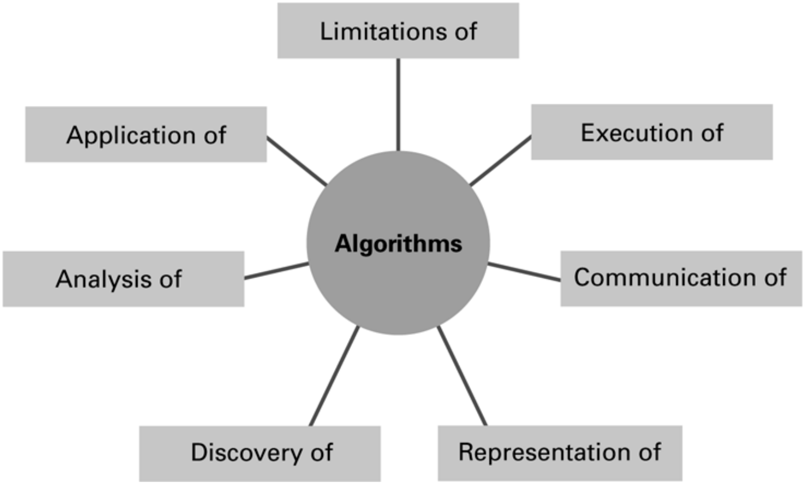
Seven “Big Ideas” that unite computer science

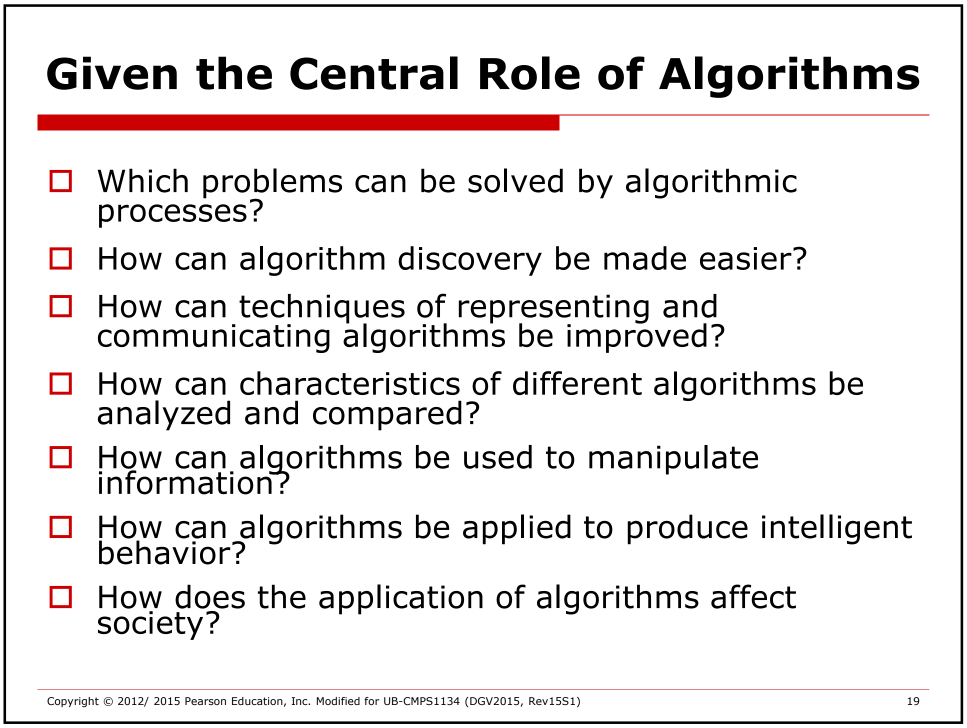
- Algorithms
- Abstraction
- Creativity
- Data
- Programing
- Internet
- Impact

Algorithms

- Computer Science is the science of algorithms
- Draws from other subjects, including
  - Mathematics
  - Engineering
  - Psychology
  - Business Administration
  - Linguistics

Figure 0.5 **The central role of algorithms in computer science**

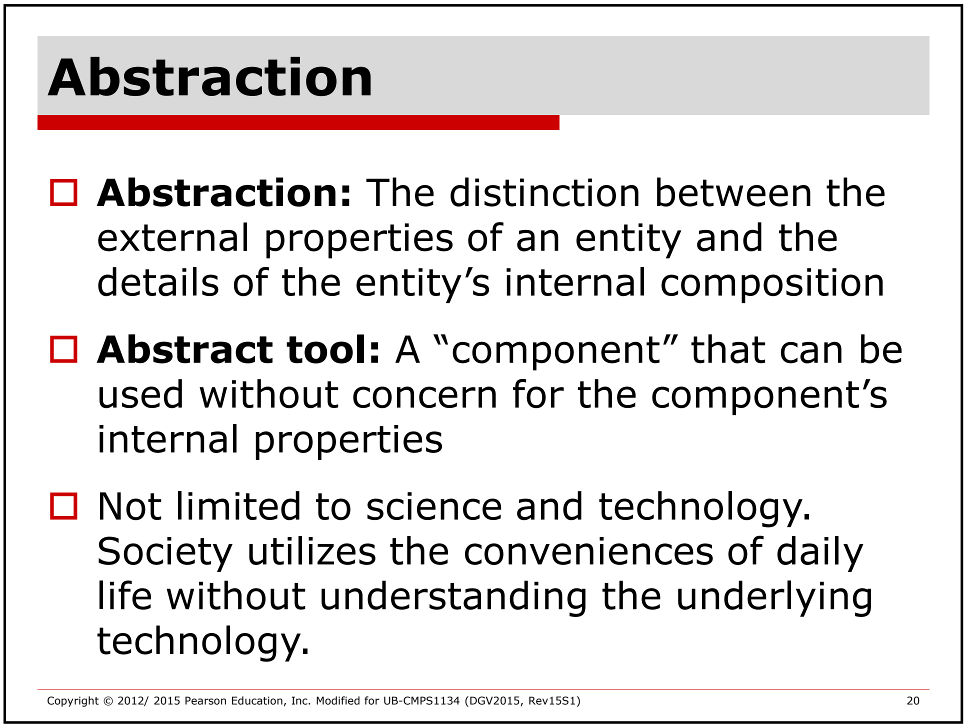




Given the Central Role of Algorithms

- Which problems can be solved by algorithmic processes?
- How can algorithm discovery be made easier?
- How can techniques of representing and communicating algorithms be improved?
- How can characteristics of different algorithms be analyzed and compared?
- How can algorithms be used to manipulate information?
- How can algorithms be applied to produce intelligent behavior?
- How does the application of algorithms affect society?

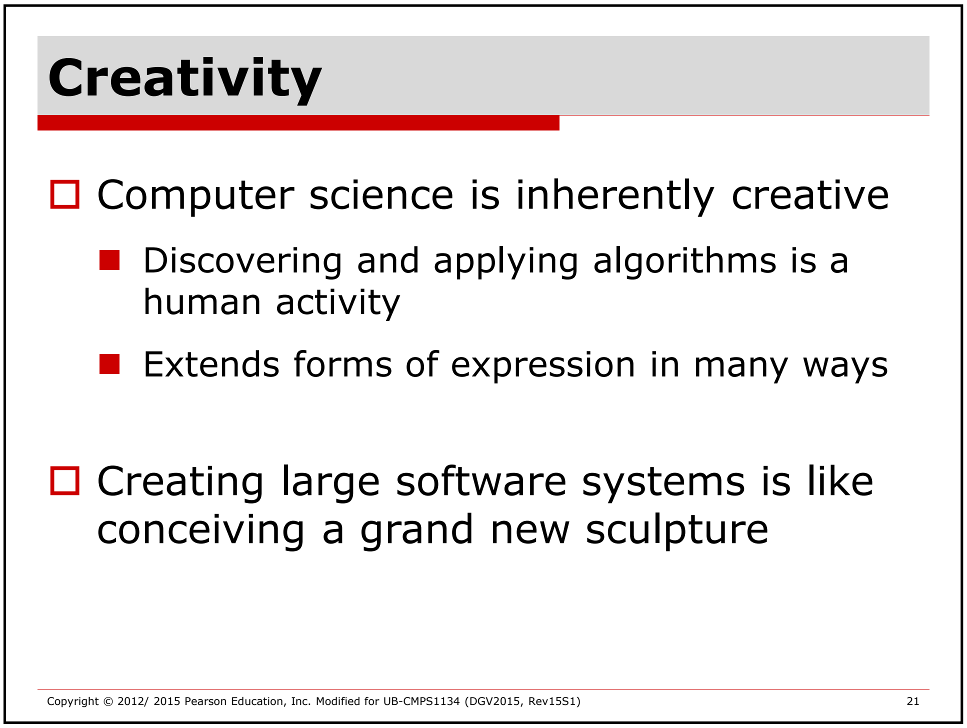
19



Abstraction

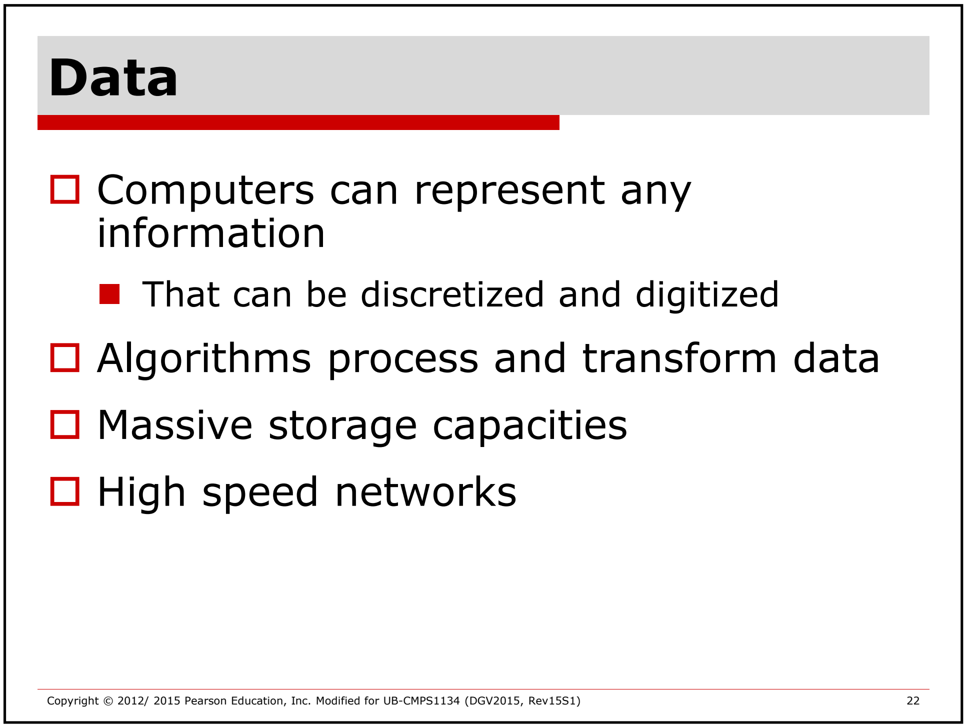
- **Abstraction:** The distinction between the external properties of an entity and the details of the entity's internal composition
- **Abstract tool:** A "component" that can be used without concern for the component's internal properties
- Not limited to science and technology. Society utilizes the conveniences of daily life without understanding the underlying technology.

20



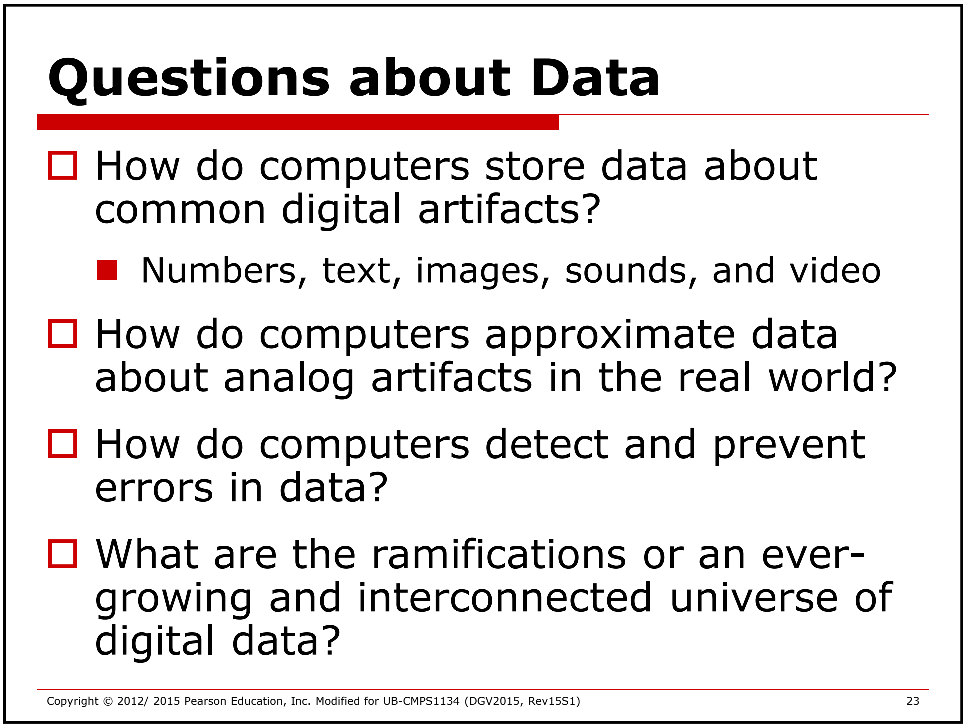
Creativity

- Computer science is inherently creative
  - Discovering and applying algorithms is a human activity
  - Extends forms of expression in many ways
- Creating large software systems is like conceiving a grand new sculpture



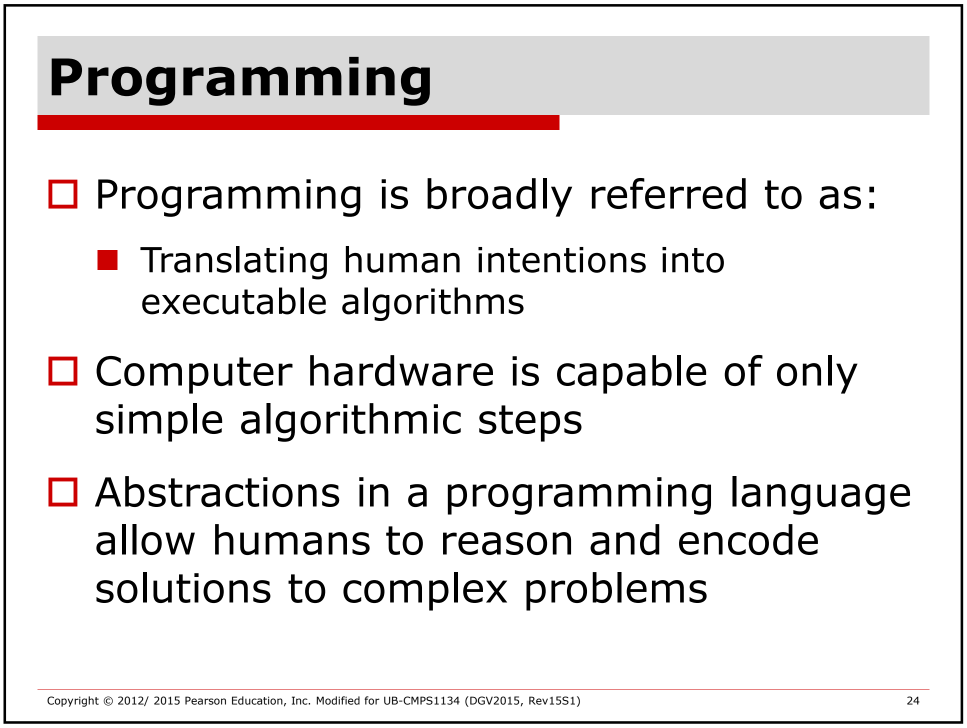
Data

- Computers can represent any information
  - That can be discretized and digitized
- Algorithms process and transform data
- Massive storage capacities
- High speed networks



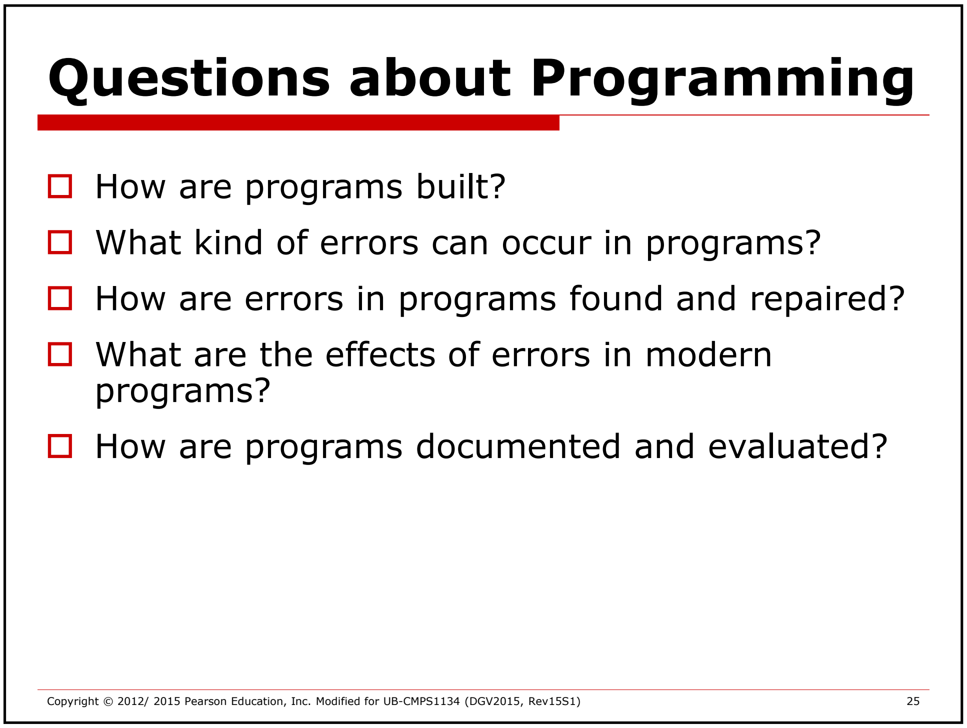
Questions about Data

- □ How do computers store data about common digital artifacts?
  - Numbers, text, images, sounds, and video
- □ How do computers approximate data about analog artifacts in the real world?
- □ How do computers detect and prevent errors in data?
- □ What are the ramifications or an ever-growing and interconnected universe of digital data?



Programming

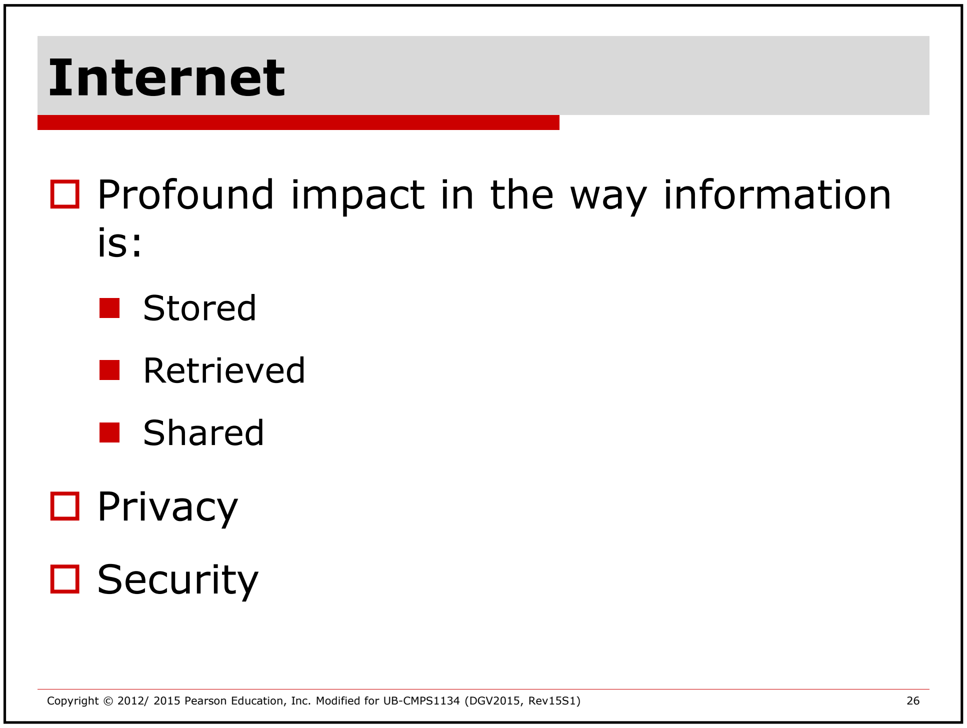
- □ Programming is broadly referred to as:
  - Translating human intentions into executable algorithms
- □ Computer hardware is capable of only simple algorithmic steps
- □ Abstractions in a programming language allow humans to reason and encode solutions to complex problems



Questions about Programming

- How are programs built?
- What kind of errors can occur in programs?
- How are errors in programs found and repaired?
- What are the effects of errors in modern programs?
- How are programs documented and evaluated?

25



Internet

- Profound impact in the way information is:
  - Stored
  - Retrieved
  - Shared
- Privacy
- Security

26

Impact

□ Social, ethical, legal impacts including:

- Security concerns
- Issues of software ownership and liabilities
- Social impact of database technology
- Consequences of artificial intelligence

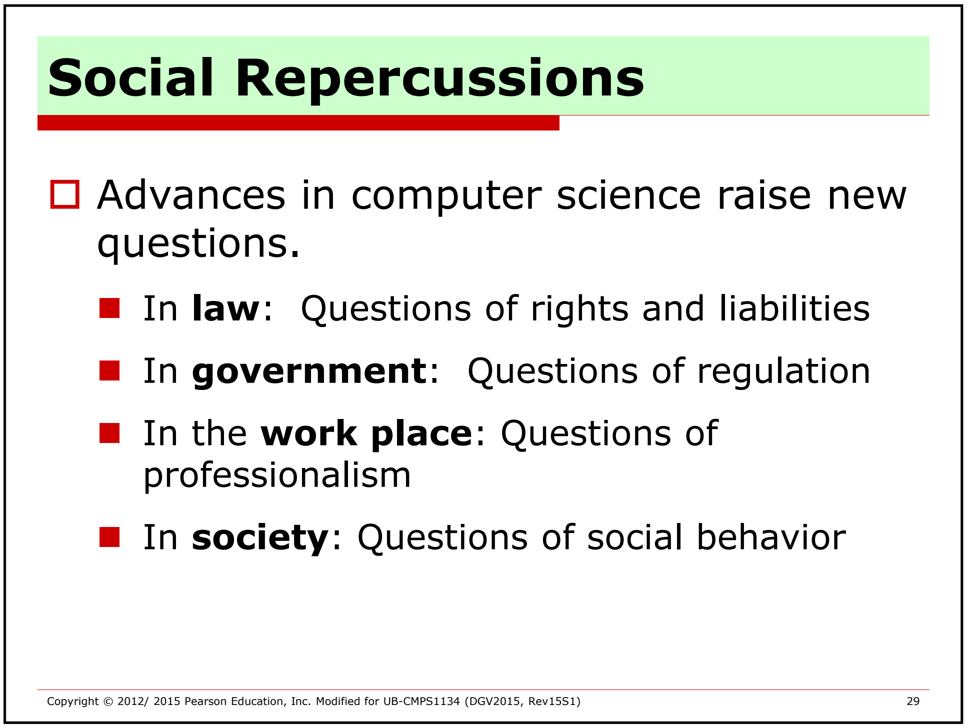
Impact

□ No “Correct” answers, instead increase awareness of:

- Various stakeholders
- Alternatives
- Short term and long term consequences

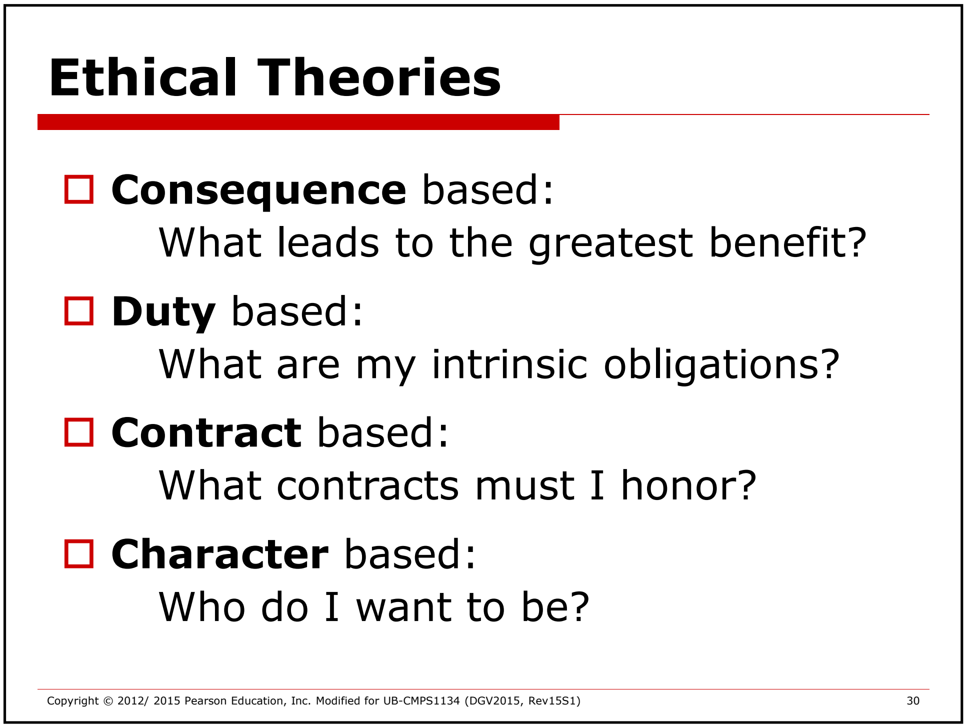
□ Character-based ethics

- “Good Behavior” is a consequence of “Good Character”



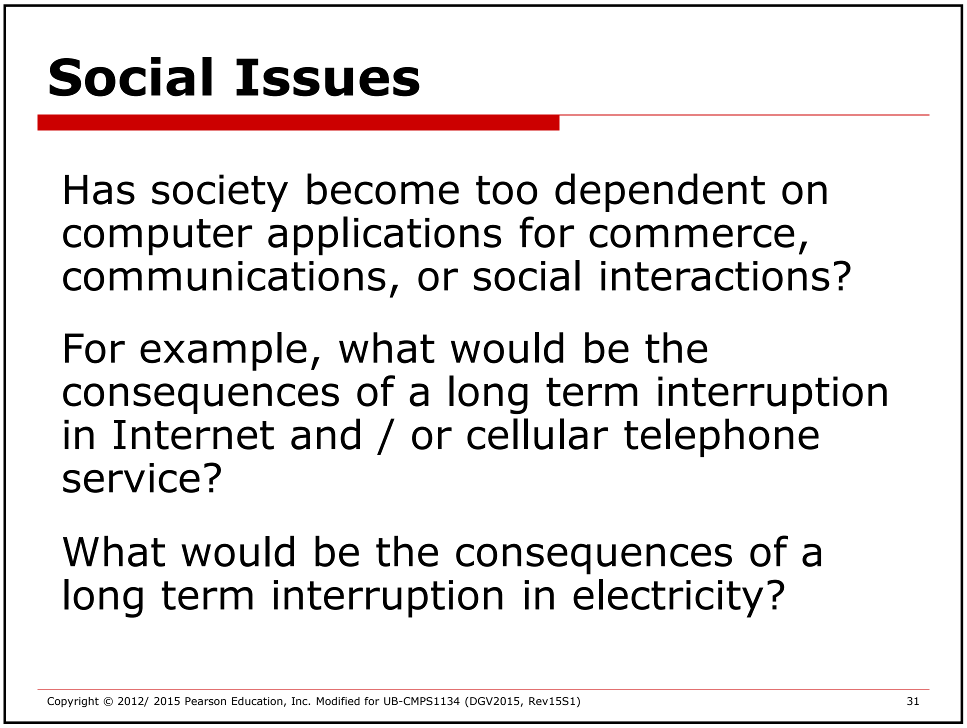
Social Repercussions

- □ Advances in computer science raise new questions.
  - In **law**: Questions of rights and liabilities
  - In **government**: Questions of regulation
  - In the **work place**: Questions of professionalism
  - In **society**: Questions of social behavior



Ethical Theories

- □ **Consequence** based:
  What leads to the greatest benefit?
- □ **Duty** based:
  What are my intrinsic obligations?
- □ **Contract** based:
  What contracts must I honor?
- □ **Character** based:
  Who do I want to be?



Social Issues

Has society become too dependent on computer applications for commerce, communications, or social interactions?

For example, what would be the consequences of a long term interruption in Internet and / or cellular telephone service?

What would be the consequences of a long term interruption in electricity?