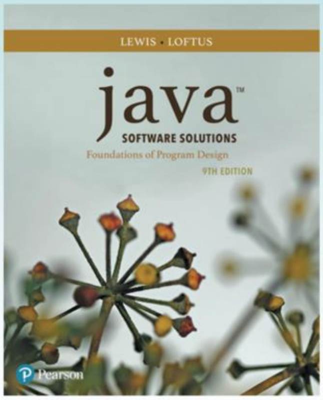
Chapter 1
Introduction



Java Software Solutions
Foundations of Program Design
9th Edition

John Lewis
William Loftus

Focus of the Course

- Object-Oriented Software Development
  - problem solving
  - program design, implementation, and testing
  - object-oriented concepts
    - classes
    - objects
    - encapsulation
    - inheritance
    - polymorphism
  - graphical user interfaces
  - the Java programming language

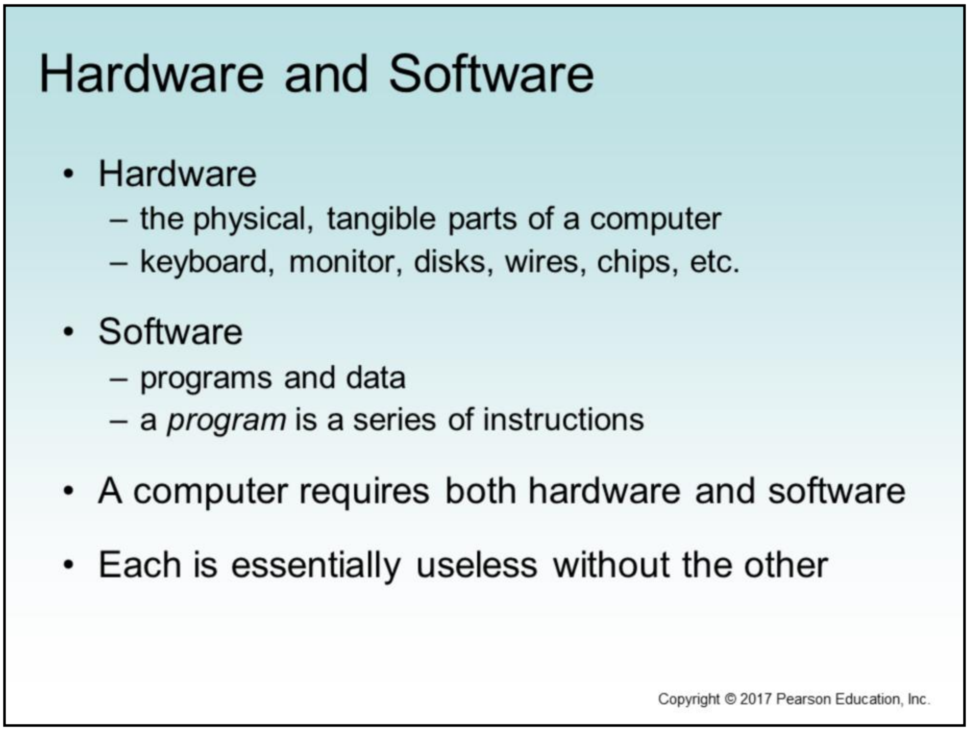
Introduction

- We start with the fundamentals of computer processing
- Chapter 1 focuses on:
  - components of a computer
  - how computers store and manipulate information
  - computer networks
  - the Internet and the World Wide Web
  - programming and programming languages
  - an introduction to Java
  - an overview of object-oriented concepts

-Why focus on components of a computer when this is a software course?

Outline

- Computer Processing
- Hardware Components
- Networks
- The Java Programming Language
- Program Development
- Object-Oriented Programming



How specifically does software “control” hardware?

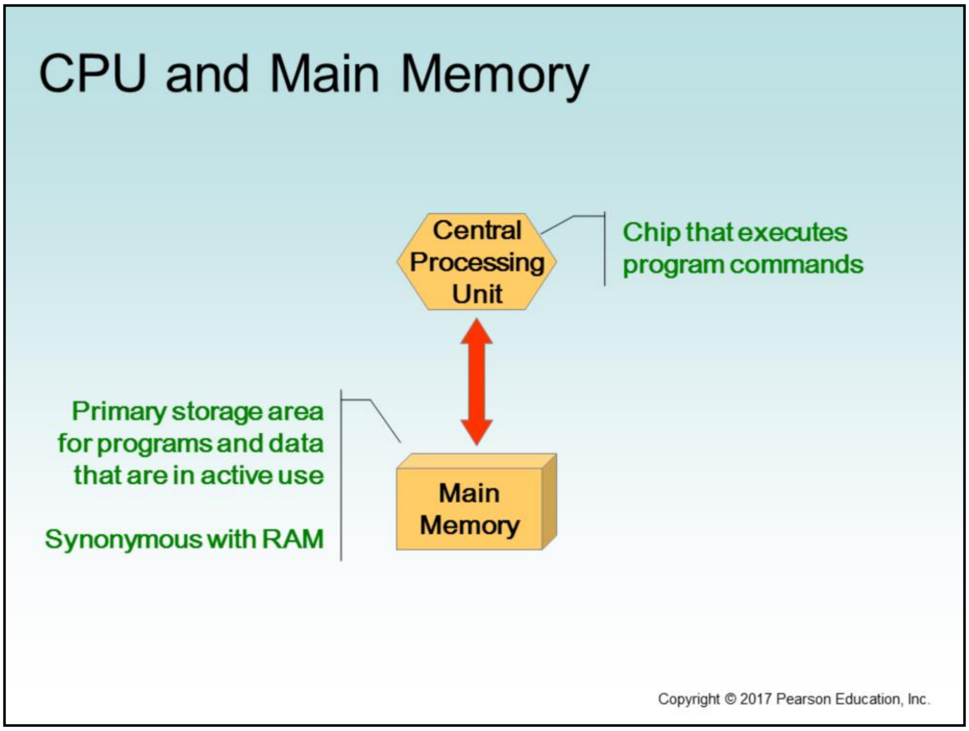
-Software instructions converted to “machine code”

-Consists of 0 and 1, 0 is off, 1 is on

-Off/on controls the state of a silicon transistor (hardware)

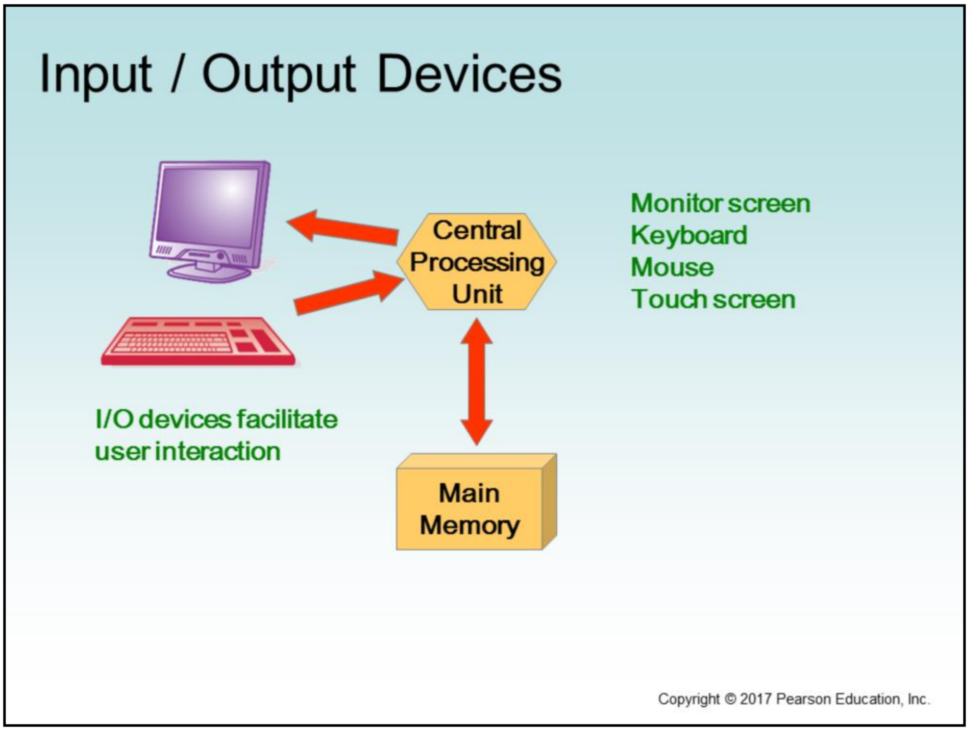
-Millions of transistors in chips work together to solve boolean calculations

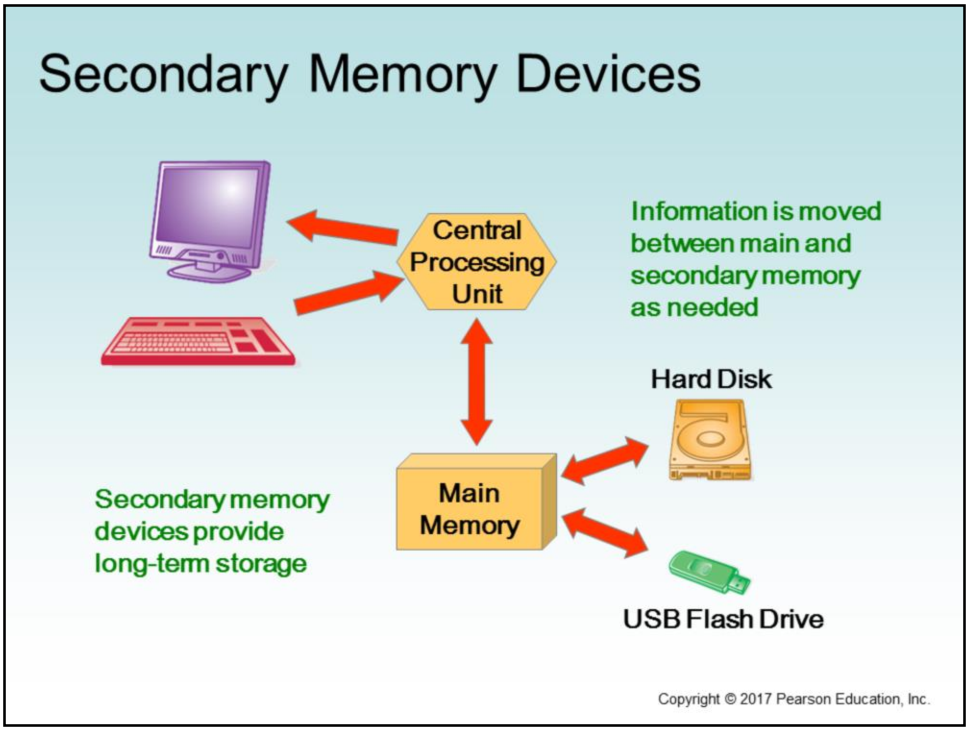
-Good book Petzhold “Code” explains the process



-Primary storage area also referred to as RAM (Random access memory)

-Instructions (as 0 or 1) stored in RAM, accessed by CPU during processing


Input / Output Devices
Central Processing Unit
Monitor screen
Keyboard
Mouse
Touch screen
I/O devices facilitate user interaction
Main Memory
Copyright © 2017 Pearson Education, Inc.



-Secondary memory devices also referred to as hard disk or storage

-Know difference between RAM (main memory) and storage (secondary)

-Process of moving program from disk to memory for execution – loading

Software Categories

- Operating System
  - controls all machine activities
  - provides the user interface to the computer
  - manages resources such as the CPU and memory
  - Windows, Mac OS, Unix, Linux,
- Application program
  - generic term for any other kind of software
  - word processors, missile control systems, games
- Most operating systems and application programs have a *graphical user interface* (GUI)

-Operating system: interface between user and computer

-Interface doesn’t have to be graphical (e.g. Linux/DOS command line)

-Android operating system is Linux

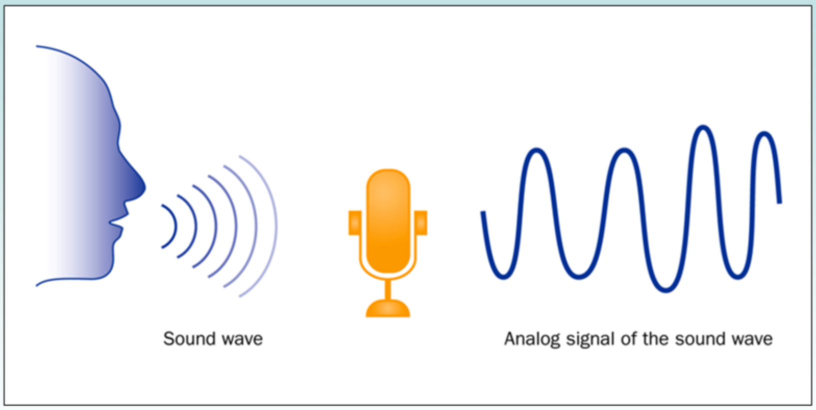
-Applications: programs we’ll write in this course

Analog vs. Digital

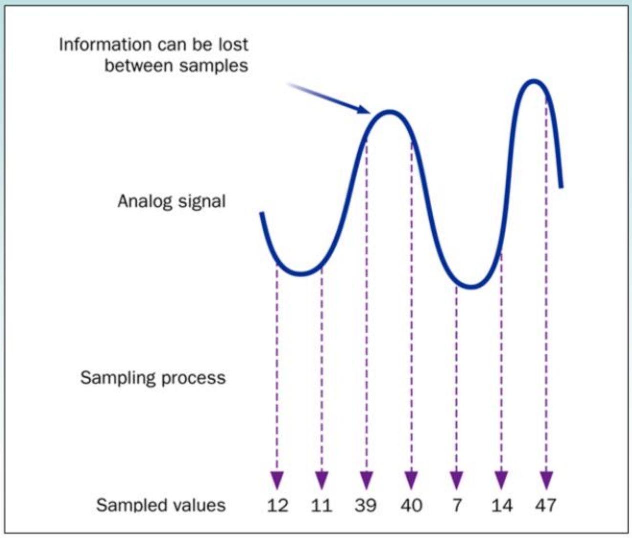
- There are two basic ways to store and manage data:
- *Analog*
  - continuous, in direct proportion to the data represented
  - music on a record album - a needle rides on ridges in the grooves that are directly proportional to the voltages sent to the speaker
- *Digital*
  - the information is broken down into pieces, and each piece is represented separately
  - *sampling* – record discrete values of the analog representation
  - music on a compact disc - the disc stores numbers representing specific voltage levels sampled at specific times

-Draw continuous analog signal (on board)
-Sample at regular intervals, creates a digital signal
-We can represent information digitally (discrete values)

Analog Information



Sampling


Information can be lost between samples
Analog signal
Sampling process
Sampled values
12
11
39
40
7
14
47

Digital Information

- Computers store all information digitally:
  - numbers
  - text
  - graphics and images
  - audio
  - video
  - program instructions
- In some way, all information is *digitized* - broken down into pieces and represented as numbers

Representing Text Digitally

- For example, every character is stored as a number, including spaces, digits, and punctuation
- Corresponding upper and lower case letters are separate characters

Hi, Heather.

72 105 44 32 72 101 97 116 104 101 114 46

Binary Numbers

- Once information has been digitized, it is represented and stored in memory using the *binary number system*
- A single binary digit (0 or 1) is called a *bit*
- Devices that store and move information are cheaper and more reliable if they have to represent only two states
- A single bit can represent two possible states, like a light bulb that is either on (1) or off (0)
- Permutations of bits are used to store values

-Digital information samples stored as numbers

-These numbers are in a binary form (two states, 0 or 1, off or on)

-Binary number system has a base value of 2 (two states)

-Such a system works well to control hardware chips, transistors

Bit Permutations

1 bit	2 bits	3 bits	4 bits	
0	00	000	0000	1000
1	01	001	0001	1001
	10	010	0010	1010
	11	011	0011	1011
		100	0100	1100
		101	0101	1101
		110	0110	1110
		111	0111	1111

Each additional bit doubles the number of possible permutations

-How does 0 and 1 translate to information?

-Book example, gears of a car (00-park,01-drive,10-rev,11-neutral)

-N bits represent 2^N unique possibilities (or permutations)

Bit Permutations

- Each permutation can represent a particular item
- There are 2^N permutations of N bits
- Therefore, N bits are needed to represent 2^N unique items

How many items can be represented by

1 bit ?	**2^1 = 2 items**
2 bits ?	**2^2 = 4 items**
3 bits ?	**2^3 = 8 items**
4 bits ?	**2^4 = 16 items**
5 bits ?	**2^5 = 32 items**

Quick Check

How many bits would you need to represent each of the 50 United States using a unique permutation of bits?

-Hint: Great example of a Quiz questions

Quick Check

How many bits would you need to represent each of the 50 United States using a unique permutation of bits?

Five bits wouldn't be enough, because 2^5 is 32.

Six bits would give us 64 permutations, and some wouldn't be used.

000000 Alabama
000001 Alaska
000010 Arizona
000011 Arkansas
000100 California
000101 Colorado

etc.

-Self-Review question 1.3:

44,000 (numbers/sec)*60(sec/min)*3(min) = 7,920,000 numbers needed to store 3 minute song