Lecture #1
CSC 200-04L Fall 2018

Ambrose Lewis
tjl274@email.vccs.edu

Agenda

- An Overview of the History of Computers
- What is a computer?

An Overview of the History of Computers

- The first device known to carry out calculations was the abacus
- The abacus uses a system of sliding beads on a rack for addition and subtraction
- Blaise Pascal invented the calculating device called the Pascaline

An Overview of the History of Computers (Cont.)

- In 1819, Joseph Jacquard, a French weaver, discovered that the weaving instructions for his looms could be stored on cards with holes punched in them
- In the early and mid-1800s, Charles Babbage, an English mathematician and physical scientist, designed two calculating machines: the difference engine and the analytical engine

An Overview of the History of Computers (Cont.)

- The first computer-like machine was the Mark I
  - Built in 1944
  - Used punched cards to feed data into the machine
  - 52 feet long, weighed 50 tons, and had 750,000 parts
- In 1946, ENIAC (Electronic Numerical Integrator and Calculator) was built at the University of Pennsylvania
  - Contained 18,000 vacuum tubes and weighed some 30 tons

An Overview of the History of Computers (Cont.)

- In 1956, the invention of the transistors resulted in smaller, faster, more reliable, and more energy-efficient computers
- This era also saw the emergence of the software development industry with the introduction of FORTRAN and COBOL, two early programming languages
- In 1970, the microprocessor, an entire CPU on a single chip, was invented
- In 1977, Stephen Wozniak and Steven Jobs designed and built the first Apple computer in their garage
- In 1981, IBM introduced its personal computer (PC)

What is a computer?

- A computer is an electronic device capable of performing commands
- The 5 basic commands that a computer performs are:
  - input (get data)
  - output (display results)
  - store data
  - perform arithmetic operations
  - Perform logical operations

What is a computer?

- A computer has two major components:
  - Hardware: physical components of the computer system (the things you can touch)
  - Software: logical components of the computer system (used to carry out the instructions)
  - Firmware is “in-between” software
    - semi-permanently placed in hardware
    - it does not disappear when hardware is powered off
    - changed via special installation processes or with an administration tool

Computer Hardware Overview
Major Components

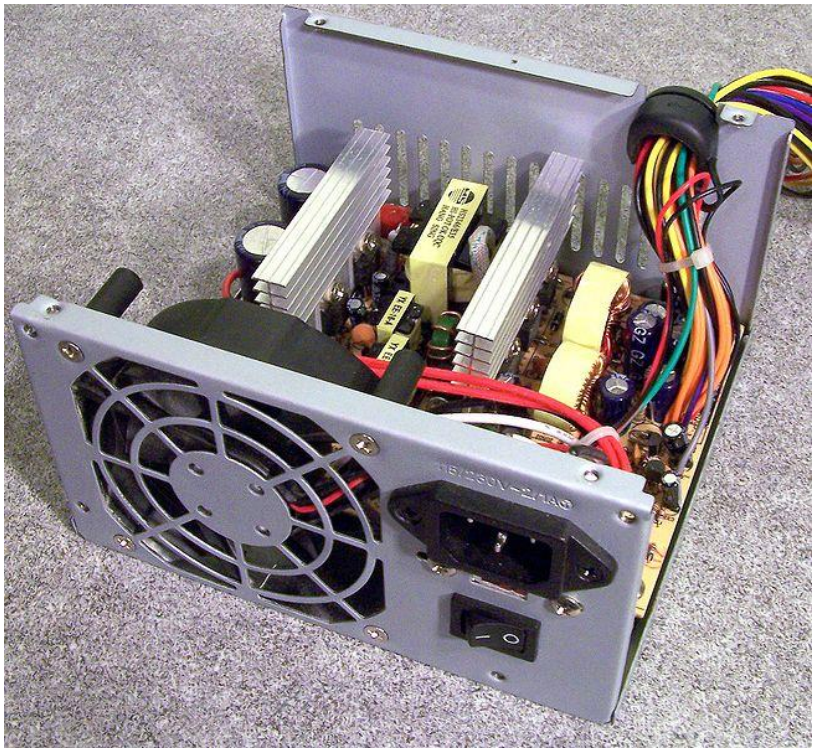
- Power Supply
- Motherboard
- CPU
- Memory
- Graphics Cards
- Input / Output Devices

http://misrarecords.com/donewaiting-ohios-main-independent-music-blog-will-cease-digital-operations-and-go-print-only-with-10th-anniversary-bash/smashed-computer/

Computer Autopsy Parts will be Displayed

Computer Hardware Overview
Power Supply

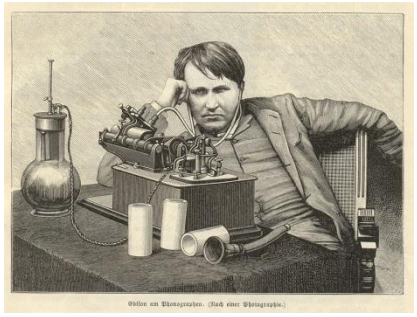
- Square metal box with a bunch of colored wired coming out, a connector for the power cord, and a fan
- Converts Alternating Current (AC) to DC
  - In US, 110-120 V at 60 Hz
- Provides Direct Current (DC) to other hardware inside the computer
  - Typically provides between 500 – 1400 Watts of power
  - Ratings are manufacturer specified, not always “apples to apples” comparisons



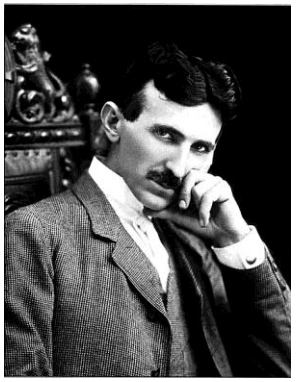
http://en.wikipedia.org/wiki/File:PSU-Open1.jpg

Computer Hardware Overview
Electrical Terms: Current

- "War of Currents":
  - DC versus AC
  - Between:
    - Edison (DC) versus
    - Westinghouse and Tesla (AC)
  - RIP Topsy the Elephant
- Analysis:
  - DC doesn't travel far and is at always the same voltage
  - AC travels far and is easily converted via transformers
  - AC uses a rotating magnet to vary voltage

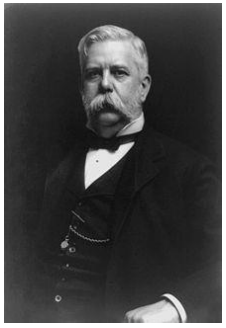


http://en.wikipedia.org/wiki/File:Edison.jpg



http://en.wikipedia.org/wiki/File:N.Tesla.JPG

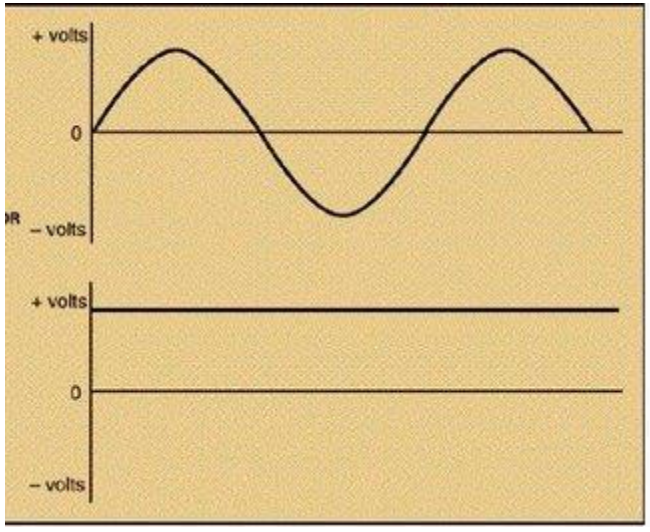
These Guys…



http://en.wikipedia.org/wiki/George_Westinghouse

AC

DC



http://3.bp.blogspot.com/_D4YflnTR13g/SXbCjpCMoCI/AAAAAAAAAnc/1CbOign7uXw/s400/untitled.bmp

Not These Guys…

http://www.sweetslyrics.com/images/img_gal/61_ac_dc-4.jpg

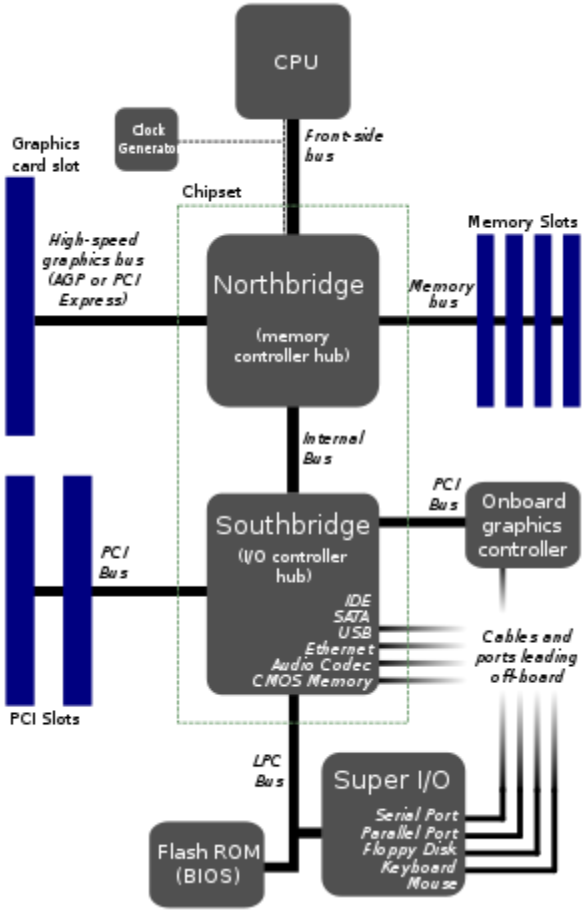
Computer Hardware Overview Electrical Terms: Volt, Current, & Watts

- Conductors (discuss examples of good & bad conductors)
- Volt = electric potential
  - Voltage measures the force potential (compare a 12 volt car battery to a 1.5 volt flashlight battery)
- Current = electricity moving through a wire in a circuit is current
  - Watts = measure of power
  - Watts = volts * amps
  - Power is a measure of electrical usage per second

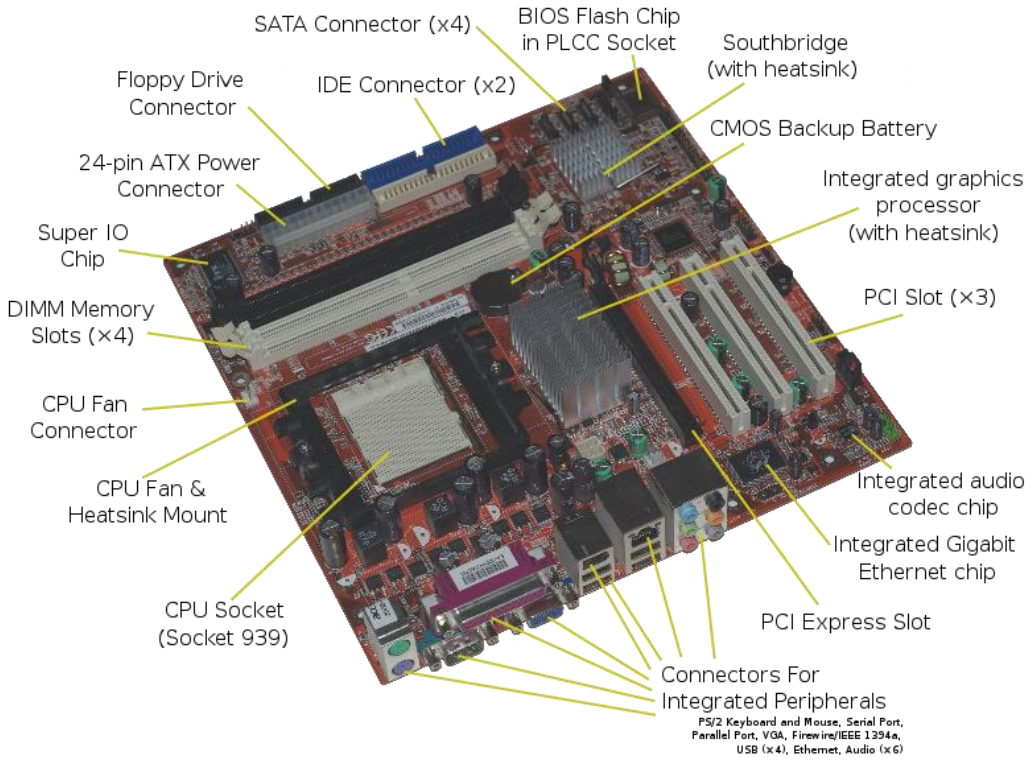
Computer Hardware Overview Motherboard

- Motherboard is the central printed circuit board (PCB) of a personal computer (PC)
- Connects everything together
- Bus = subsystem that transfers data between components
  - Front Side Bus (for CPU, RAM, and Video Card/PCI Xpress)
  - Slower Bus for I/O devices (USB, audio, etc.)
- Clock Generator to synchronize the various components
- Flash ROM for BIOS firmware
- Slots for expansion cards (PCI, PCI Express, etc)
- CPU socket
- Memory sockets
- PS2 connector (keyboard & mouse), network, etc

Computer Hardware Overview Motherboard



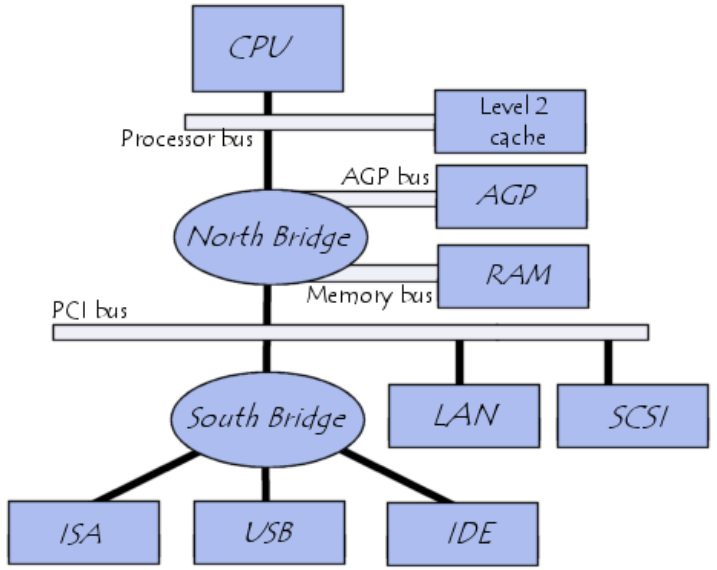
http://images1.wikia.nocookie.net/__cb20090614012254/bmet/images/0/0e/300px-Motherboard_diagram.svg.png



http://en.wikipedia.org/wiki/File:Acer_E360_Socket_939_motherboard_by_Foxconn.svg

Data Bus

- A communications system within a computer that exchanges data between components
  - For example, the bus moves data between main memory and the CPU
- Computers can have multiple buses with different capabilities such as size (how many bits can be sent at once), speed, etc.



http://ccm.net/contents/375-computer-bus-what-is-it

Computer Hardware Overview
CPU

- "brain" of the computer
  - execute a sequence of stored instructions called a program
  - typically large scale integrated circuits (IC) with multiple connecting pins. Usually has a heat sink attached
  - Two main parts:
    - ALU = arithmetic logic unit
      - Performs arithmetic & logic operations
    - CU = control unit
      - Extracts instructions from memory and decodes/executes them

http://www.build-your-own-computer.net/image-files/cpu-reviews-01.jpg

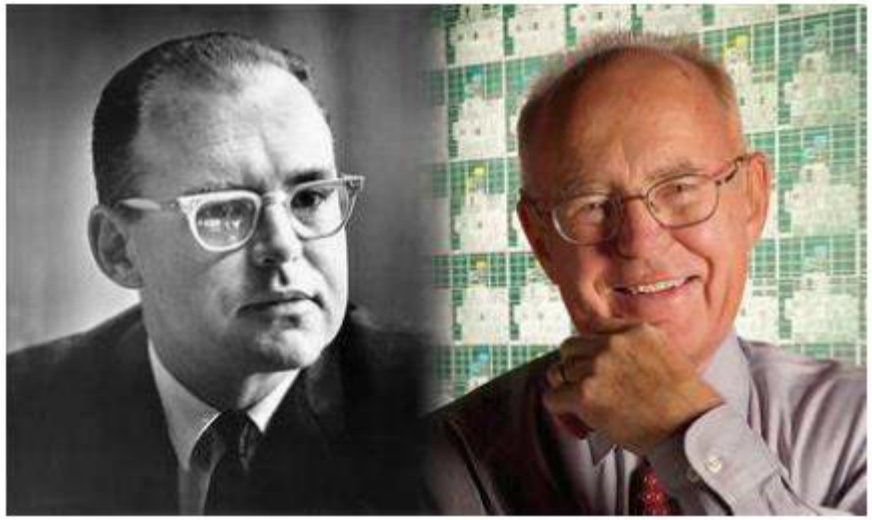
http://product-image.tradeindia.com/00219944/b/0/Aluminium-Heat-Sink.jpg

Computer Hardware Overview
CPU Steps

- 4 steps:
  - Fetch
    - Retrieve an instruction from memory
  - Decode
    - Translate from number into operation
    - Opcode = operation code
    - Parameters (static value, memory address)
  - Execute
    - Do the work
  - Writeback
    - Write result back to memory

Computer Hardware Overview
Moore's Law

- The number of transistors that can be placed inexpensively on an integrated circuit doubles approximately every two years.
- See http://en.wikipedia.org/wiki/Moore's_law
- In 1965 from Gordon Moore, the co-founder of Intel

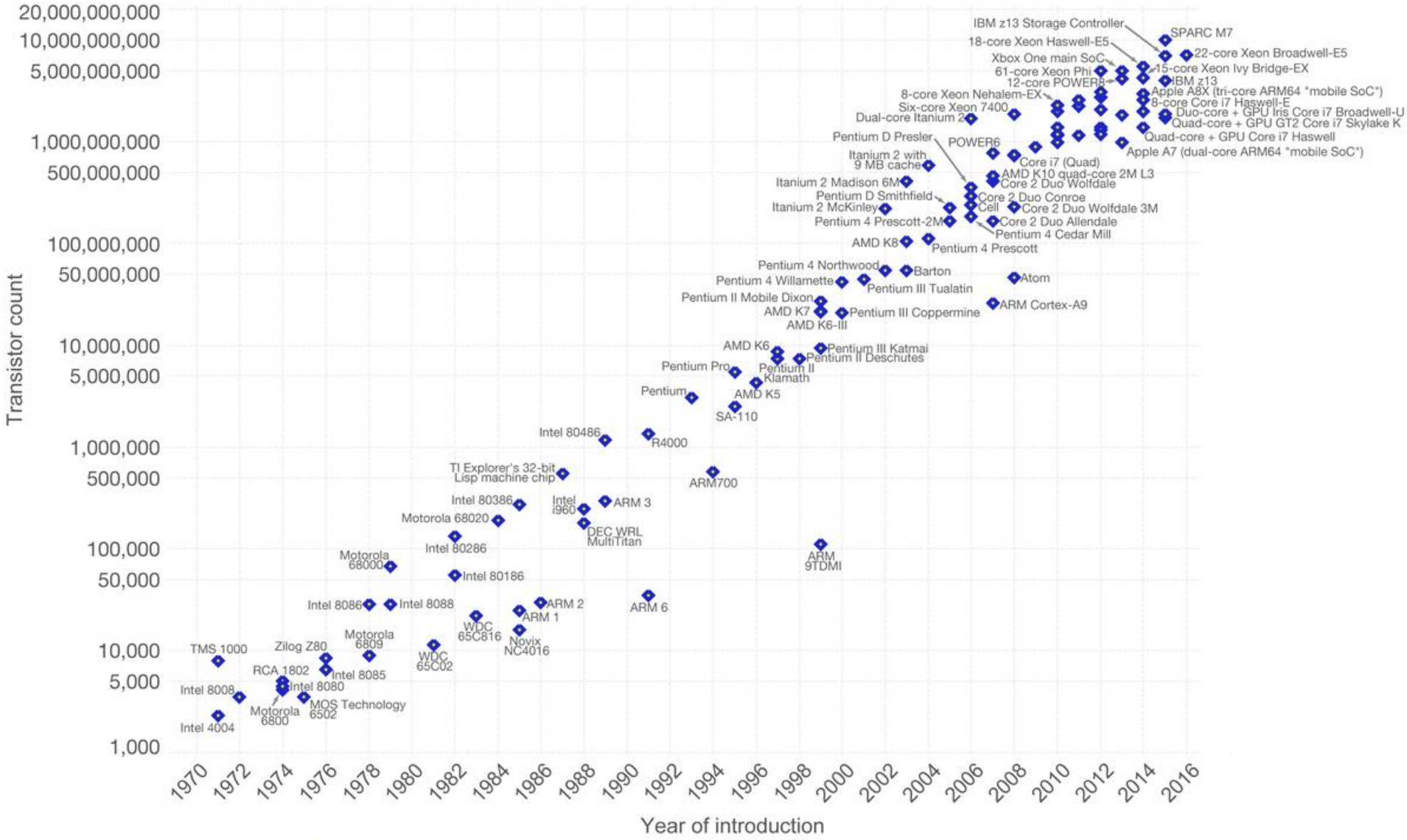


http://www.rentoid.com/blog/wp-content/uploads/2009/08/gordon-moore.jpg

Moore's Law – The number of transistors on integrated circuit chips (1971-2016)

Our World in Data

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important as other aspects of technological progress – such as processing speed or the price of electronic products – are strongly linked to Moore's law.

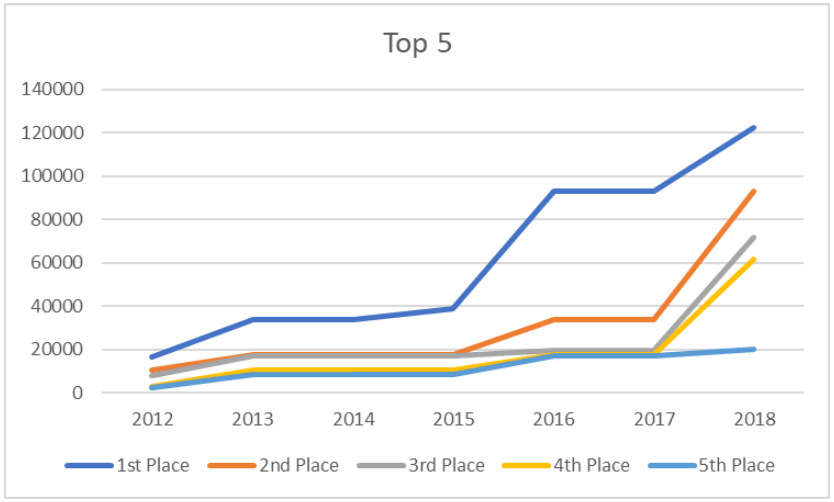


Data source: Wikipedia (https://en.wikipedia.org/wiki/Transistor_count)

The data visualization is available at OurWorldinData.org. There you find more visualizations and research on this topic.

Computer Hardware Overview
Moore's Law: Top500.org

Rank	Site	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
1	DOE/SC/Oak Ridge National Laboratory United States	**Summit** - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband IBM	2,282,544	122,300.0	187,659.3	8,806
2	National Supercomputing Center in Wuxi China	**Sunway TaihuLight** - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway NRCPC	10,649,600	93,014.6	125,435.9	15,371
3	DOE/NNSA/LLNL United States	**Sierra** - IBM Power System S922LC, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband IBM	1,572,480	71,610.0	119,193.6	
4	National Super Computer Center in Guangzhou China	**Tianhe-2A** - TH-IVB-FEP Cluster, Intel Xeon E5-2692v2 12C 2.2GHz, TH Express-2, Matrix-2000 NUDT	4,981,760	61,444.5	100,678.7	18,482
5	National Institute of Advanced Industrial Science and Technology (AIST) Japan	**AI Bridging Cloud Infrastructure (ABCI)** - PRIMERGY CX2550 M4, Xeon Gold 6148 20C 2.4GHz, NVIDIA Tesla V100 SXM2, Infiniband EDR Fujitsu	391,680	19,880.0	32,576.6	1,649
6	Swiss National Supercomputing Centre (CSCS) Switzerland	**Piz Daint** - Cray XC50, Xeon E5-2690v3 12C 2.6GHz, Aries interconnect , NVIDIA Tesla P100 Cray Inc.	361,760	19,590.0	25,326.3	2,272
7	DOE/SC/Oak Ridge National Laboratory United States	**Titan** - Cray XK7, Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA K20x Cray Inc.	560,640	17,590.0	27,112.5	8,209
8	DOE/NNSA/LLNL United States	**Sequoia** - BlueGene/Q, Power BQC 16C 1.60 GHz, Custom IBM	1,572,864	17,173.2	20,132.7	7,890
9	DOE/NNSA/LANL/SNL United States	**Trinity** - Cray XC40, Intel Xeon Phi 7250 68C 1.4GHz, Aries interconnect Cray Inc.	979,968	14,137.3	43,902.6	3,844



https://www.top500.org/list/2018/06/

Computer Hardware Overview

Memory

- In yea olden days, memory was magnetic "drum" based…this was slow
- Today, we use RAM
  - Random Access Memory
    - CPU can access any memory address at any cycle
  - Integrated circuit based, each IC holds a bit in a single capacitor
    - charged or discharged represents a bit
    - Capacitors need to be refreshed, making it Dynamic RAM (DRAM)
    - DRAM is used in PCs, laptops, Wii, Xbox, etc.
  - RAM is volatile
    - requires power to maintain the stored information
    - No power, no data
    - Non-volatile memory is slower, more expensive (flash memory)

http://en.wikipedia.org/wiki/File:Pamiec_bebnowa_1.jpg

http://www.ciscomonkeys.com/monkeystore/images/16dram.jpg

Computer Hardware Overview
Memory Speed

- Fast memory is expensive, slow memory is cheaper
- In order from fastest to slowest:
  1. CPU Register [~16 registers / CPU]
  2. Level 1 Cache (CPU Cache) [up to 2KB]
  3. Level 2 Cache [up to 2MB]
  4. Main Memory [up to 8GB]
  5. Virtual Memory (disk based) [practically unlimited]

Next slide talks about sizes...

Computer Hardware Overview
A Discussion on Sizes...

TABLE 1-1 Binary Units

Unit	Symbol	Bits/Bytes
Byte		8 bits
Kilobyte	KB	2^{10} bytes = 1024 bytes
Megabyte	MB	1024 KB = 2^{10} KB = 2^{20} bytes = 1,048,576 bytes
Gigabyte	GB	1024 MB = 2^{10} MB = 2^{30} bytes = 1,073,741,824 bytes
Terabyte	TB	1024 GB = 2^{10} GB = 2^{40} bytes = 1,099,511,627,776 bytes
Petabyte	PB	1024 TB = 2^{10} TB = 2^{50} bytes = 1,125,899,906,842,624 bytes
Exabyte	EB	1024 PB = 2^{10} PB = 2^{60} bytes = 1,152,921,504,606,846,976 bytes
Zettabyte	ZB	1024 EB = 2^{10} EB = 2^{70} bytes = 1,180,591,620,717,411,303,424 bytes

Size: Bit < Byte < Kilobyte < Megabyte < Gigabyte

Based on Java Programming: From Problem Analysis to Program Design, 5e by D. S. Malik

Computer Hardware Overview
A Discussion on Sizes…

- A little perspective…How big is a Trillion?
- http://www.grc.nasa.gov/WWW/k-12/Numbers/Math/Mathematical_Thinking/how_big_is_a_trillion.htm describes this by looking at time…a trillion seconds
- How Many Seconds in a year?
  - 60 sec/min, 60 min/hour, 24 hour/day = 86,400 sec/day
  - 365.25 days/year = 31,557,600 sec/year
  - 1 Trillion seconds = 1*10^12 = 1,000,000,000,000
  - (1*10^12 sec)/ (31,557,600 sec/year) = 31,546 years!

Computer Hardware Overview
Graphics Cards

- Expansion card or built into Motherboard
- Generates output images to display
- Very powerful, includes features like 3D, lighting & shading, etc
- GPU = graphics processor unit
  - Multiple pipelines to work in parallel
  - Can be used to supplement CPU using software like CUDA or OpenCL

Computer Hardware Overview
Input / Output Devices

- Input:
  - Network
  - CD/ROM
  - Keyboard/Mouse
  - Sensors (especially Cell Phones)

This is an example of a 3D Mouse

http://images.techtree.com/ttimages/story/88590_scn1.jpg

- Output:
  - Hard Disk
  - CD/DVD Burner
  - USB Stick
  - etc

http://www.pcguide.com/ref/hdd/z_ibm_ultrastar36zx.jpg

Computer Hardware Overview
CPU & Memory Interaction

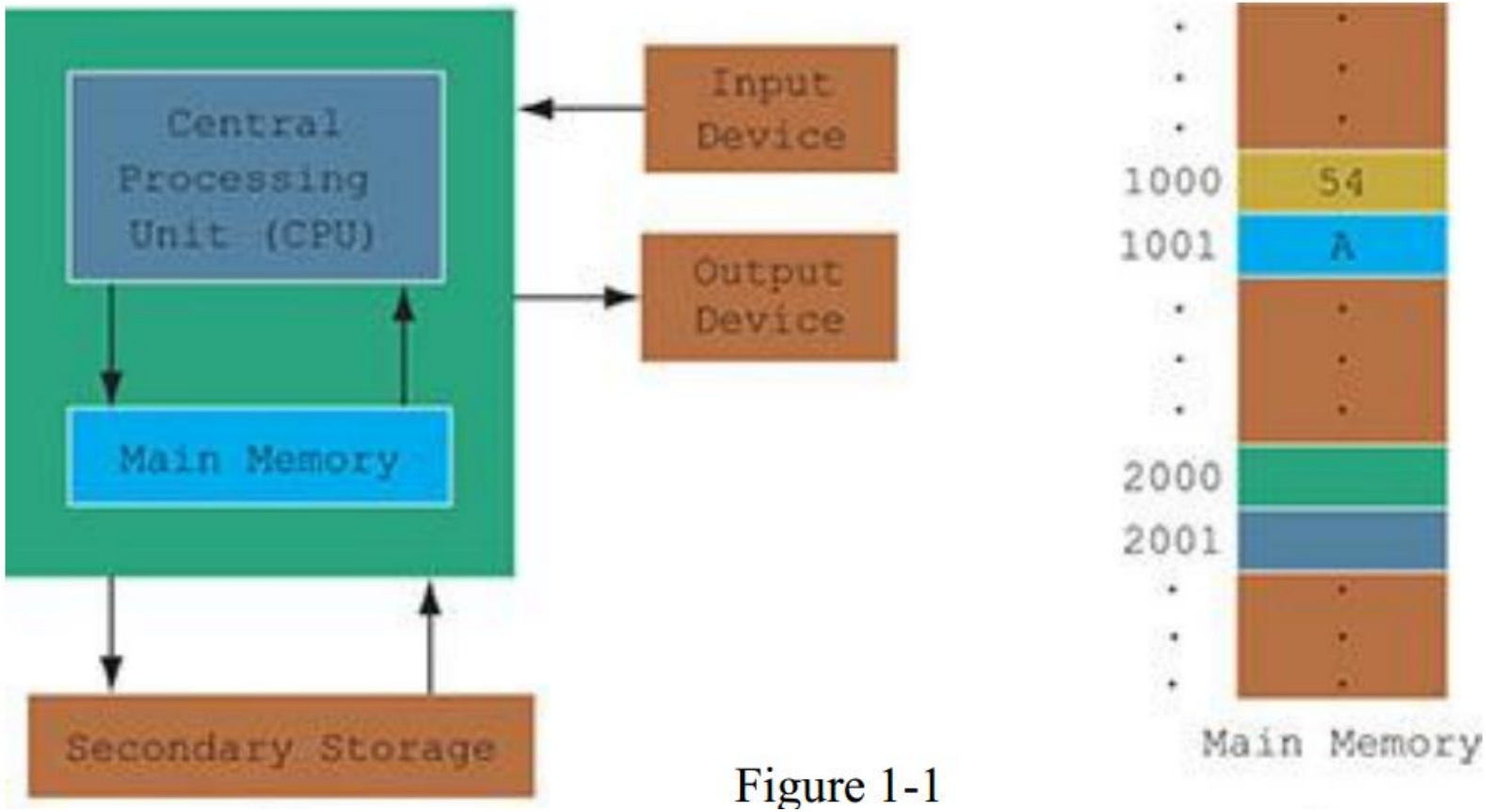
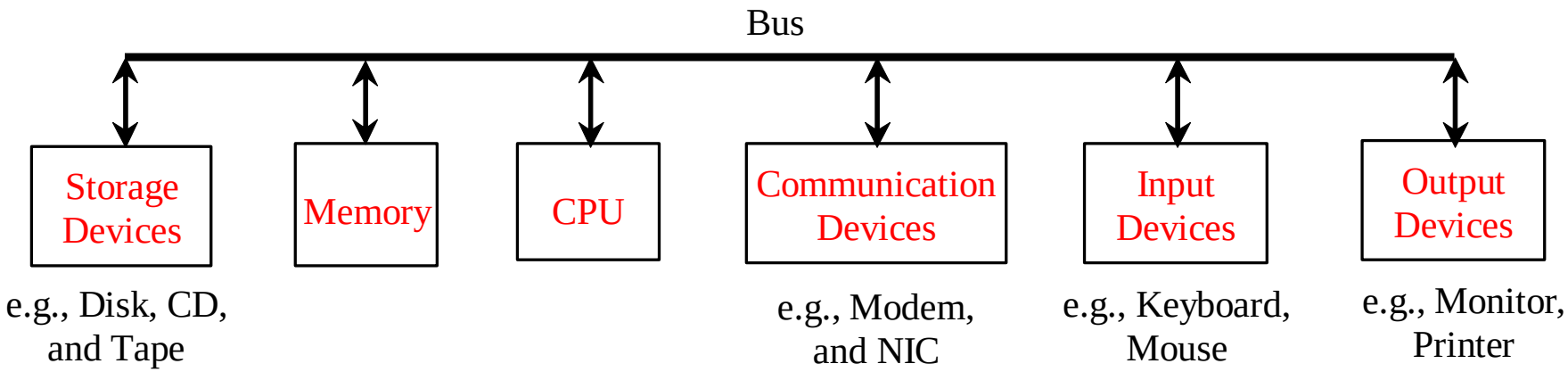


Figure 1-1

Based on Java Programming: From Problem Analysis to Program Design, 5e by D. S. Malik

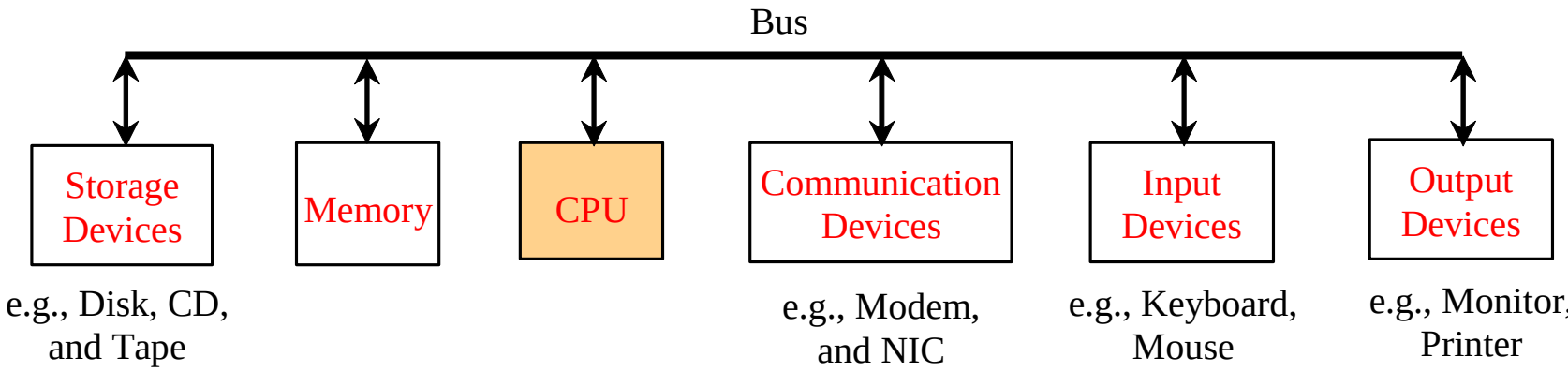
What is a Computer?

A computer consists of a CPU, memory, hard disk, floppy disk, monitor, printer, and communication devices.



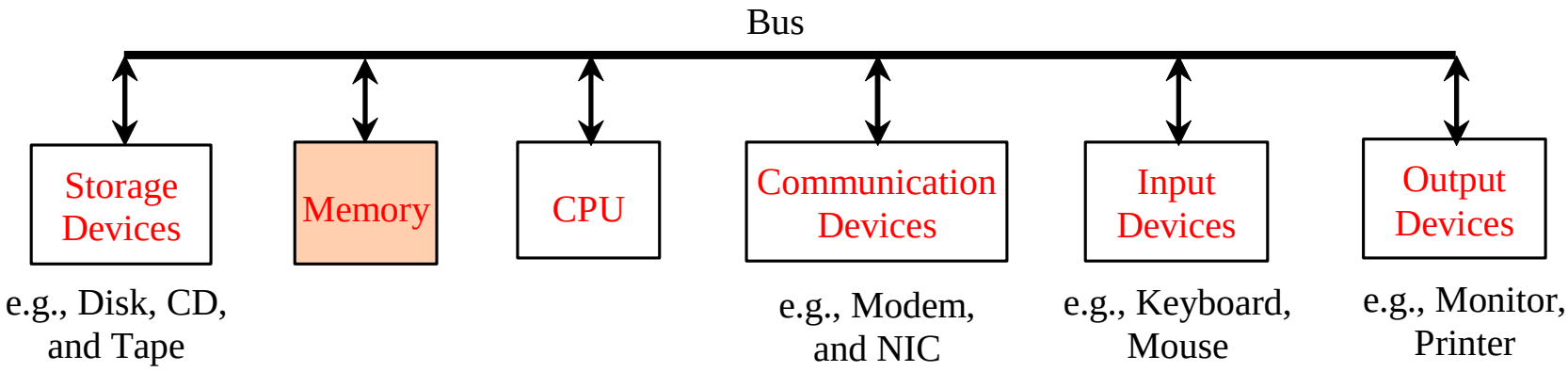
CPU

The central processing unit (CPU) is the brain of a computer. It retrieves instructions from memory and executes them. The CPU speed is measured in megahertz (MHz), with 1 megahertz equaling 1 million pulses per second. The speed of the CPU has been improved continuously. If you buy a PC now, you can get an Intel Pentium 4 Processor at 3 gigahertz (1 gigahertz is 1000 megahertz).



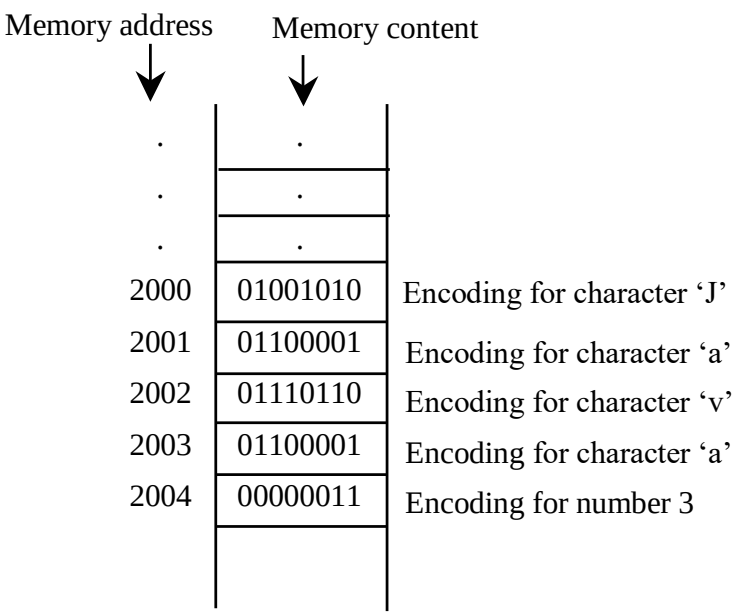
Memory

Memory is to store data and program instructions for CPU to execute. A memory unit is an ordered sequence of bytes, each holds eight bits. A program and its data must be brought to memory before they can be executed. A memory byte is never empty, but its initial content may be meaningless to your program. The current content of a memory byte is lost whenever new information is placed in it.



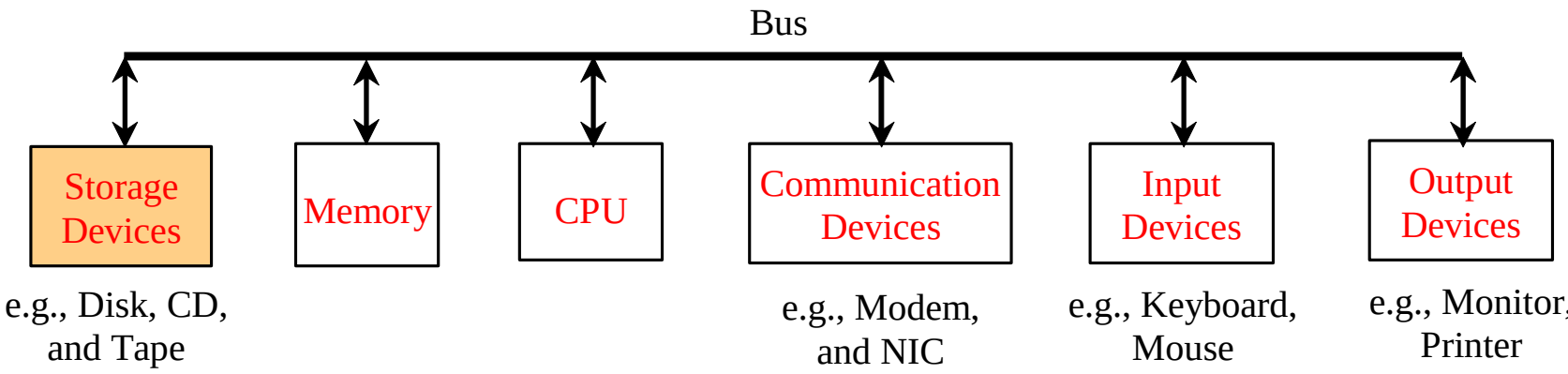
How Data is Stored?

Data of various kinds, such as numbers, characters, and strings, are encoded as a series of bits (zeros and ones). Computers use zeros and ones because digital devices have two stable states, which are referred to as *zero* and *one* by convention. The programmers need not to be concerned about the encoding and decoding of data, which is performed automatically by the system based on the encoding scheme. The encoding scheme varies. For example, character 'J' is represented by 01001010 in one byte. A small number such as three can be stored in a single byte. If computer needs to store a large number that cannot fit into a single byte, it uses a number of adjacent bytes. No two data can share or split a same byte. A byte is the minimum storage unit.



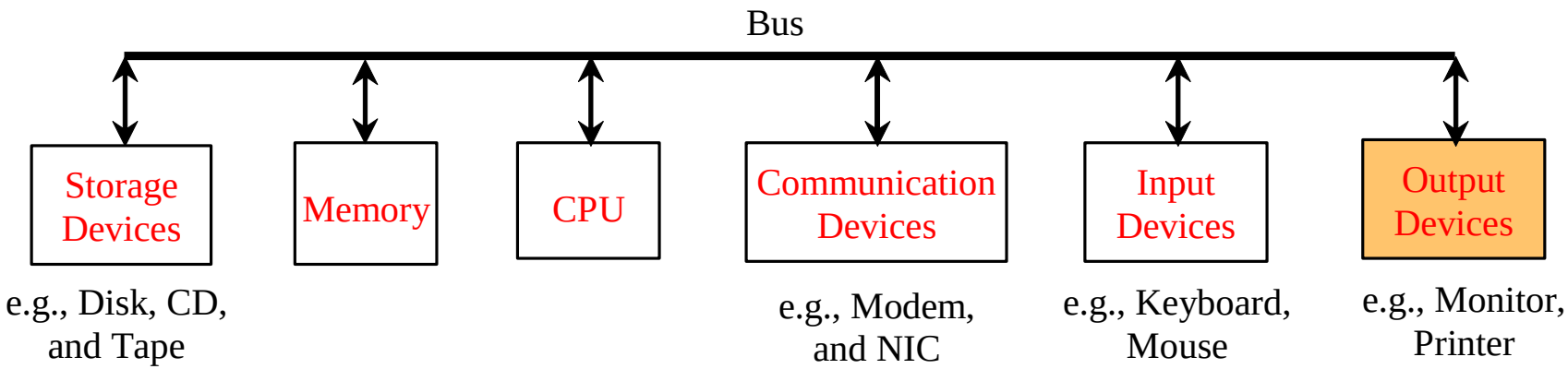
Storage Devices

Memory is volatile, because information is lost when the power is off. Programs and data are permanently stored on storage devices and are moved to memory when the computer actually uses them. There are three main types of storage devices:Disk drives (hard disks and floppy disks), CD drives (CD-R and CD-RW), and Tape drives.



Output Devices: Monitor

The monitor displays information (text and graphics). The resolution and dot pitch determine the quality of the display.



Monitor Resolution and Dot Pitch

resolution

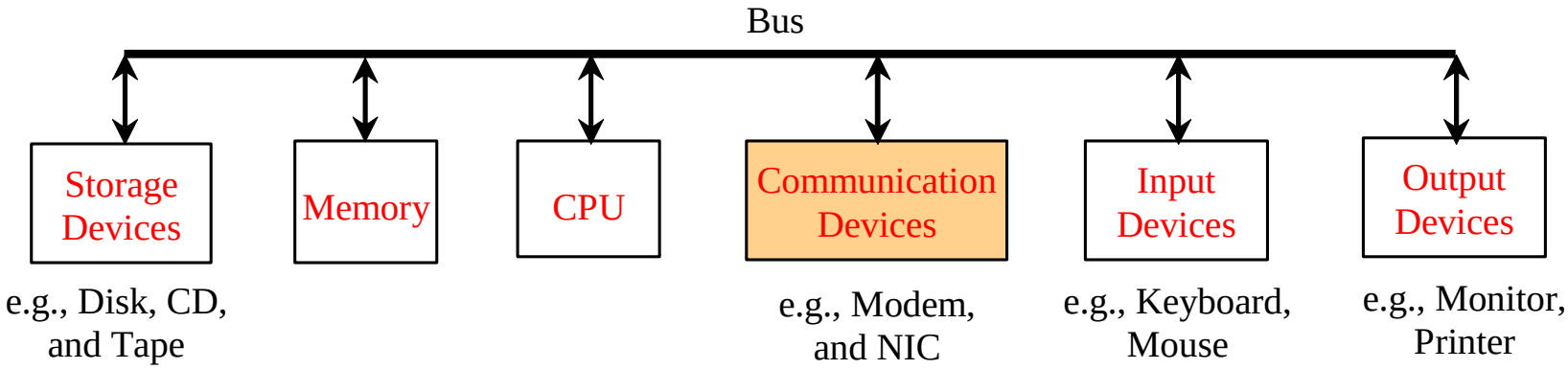
The *screen resolution* specifies the number of pixels in horizontal and vertical dimensions of the display device. *Pixels* (short for "picture elements") are tiny dots that form an image on the screen. A common resolution for a 17-inch screen, for example, is 1,024 pixels wide and 768 pixels high. The resolution can be set manually. The higher the resolution, the sharper and clearer the image is.

dot pitch

The *dot pitch* is the amount of space between pixels, measured in millimeters. The smaller the dot pitch, the sharper the display.

Communication Devices

A *regular modem* uses a phone line and can transfer data in a speed up to 56,000 bps (bits per second). A *DSL* (digital subscriber line) also uses a phone line and can transfer data in a speed 20 times faster than a regular modem. A *cable modem* uses the TV cable line maintained by the cable company. A cable modem is as fast as a DSL. Network interface card (*NIC*) is a device to connect a computer to a local area network (LAN). The LAN is commonly used in business, universities, and government organizations. A typical type of NIC, called *10BaseT*, can transfer data at 10 mbps (million bits per second).



What is software?

- Software consists of programs written to perform specific tasks
- Two types of programs:
  - System programs
  - Application programs

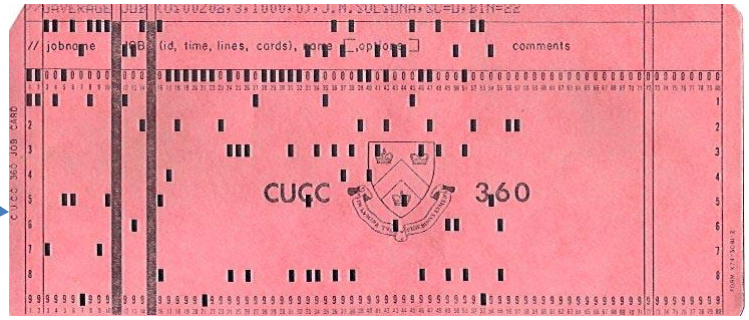
What is Software?

- Software is a sequence of instructions written to perform a specific task with a computer
- Computer programs are typically stored in non-volatile memory until requested either directly or indirectly to be executed by the computer user.
- Upon request, the program is loaded into random access memory (typically by an operating system) where it can be reached by the computer's central processor.
- The central processor then executes ("runs") the program, instruction by instruction, until termination.
  - A program in execution is called a process.
  - Termination is either by normal self-termination or by a software or hardware error.

Bullets based on http://en.wikipedia.org/wiki/Computer_program

Historical Ways to Start Software

- On “ye olden days”, software loaded from punch cards or paper tapes
- Set the address and begin running via front panel switches
- Later, programs were stored on magnetic tapes or disks
- Today, magnetic disks (hard disks), CD-ROM, DVD, or USB



http://www.columbia.edu/cu/computinghistory/card2.jpg

http://www.simulogics.com/nostalgia/DG/nova3_01.jpg

Programs

Computer *programs*, known as *software*, are instructions to the computer.

You tell a computer what to do through programs. Without programs, a computer is an empty machine. Computers do not understand human languages, so you need to use computer languages to communicate with them.

Programs are written using programming languages.

Programming Languages

Machine Language Assembly Language High-Level Language

Machine language is a set of primitive instructions built into every computer. The instructions are in the form of binary code, so you have to enter binary codes for various instructions. Program with native machine language is a tedious process. Moreover the programs are highly difficult to read and modify. For example, to add two numbers, you might write an instruction in binary like this:

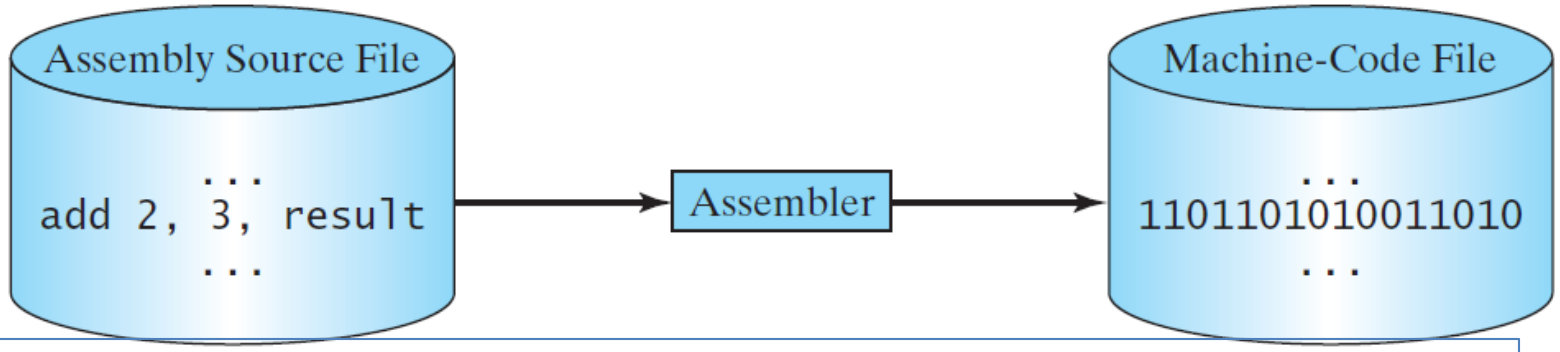
```
1101101010011010
```

Programming Languages

Machine Language Assembly Language High-Level Language

Assembly languages were developed to make programming easy. Since the computer cannot understand assembly language, however, a program called assembler is used to convert assembly language programs into machine code. For example, to add two numbers, you might write an instruction in assembly code like this:

ADDF3 R1, R2, R3



Programming Languages

Machine Language Assembly Language High-Level Language

The high-level languages are English-like and easy to learn and program. For example, the following is a high-level language statement that computes the area of a circle with radius 5:

```
area = 5 * 5 * 3.1415;
```

Popular High-Level Languages

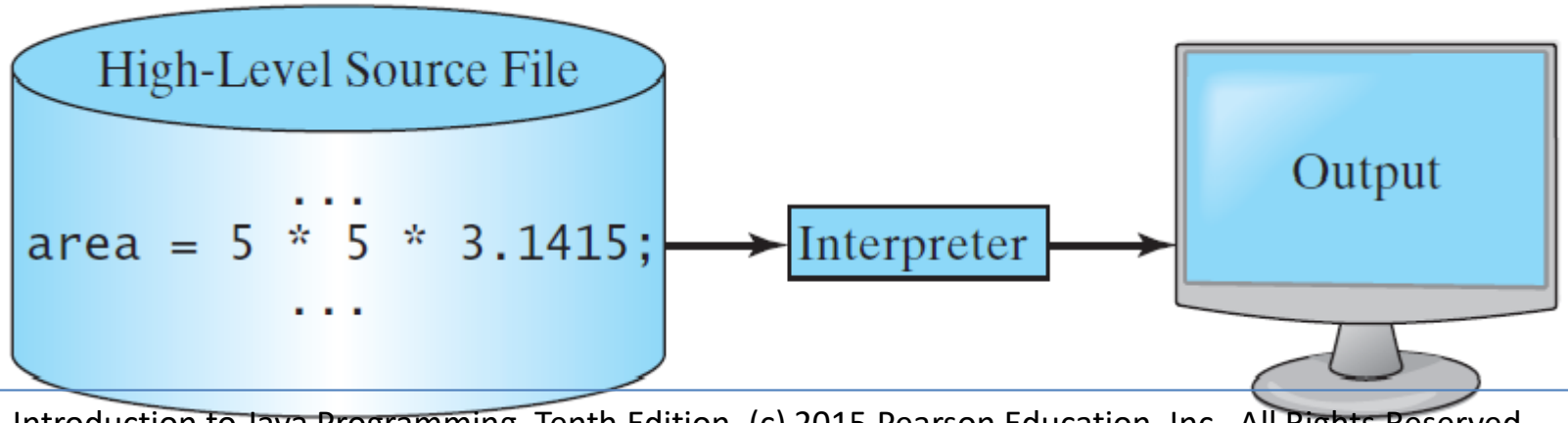
Language	Description
Ada	Named for Ada Lovelace, who worked on mechanical general-purpose computers. The Ada language was developed for the Department of Defense and is used mainly in defense projects.
BASIC	Beginner's All-purpose Symbolic Instruction Code. It was designed to be learned and used easily by beginners.
C	Developed at Bell Laboratories. C combines the power of an assembly language with the ease of use and portability of a high-level language.
C++	C++ is an object-oriented language, based on C.
C#	Pronounced "C Sharp." It is a hybrid of Java and C++ and was developed by Microsoft.
COBOL	COmmon Business Oriented Language. Used for business applications.
FORTRAN	FORmula TRANslation. Popular for scientific and mathematical applications.
Java	Developed by Sun Microsystems, now part of Oracle. It is widely used for developing platform-independent Internet applications.
Pascal	Named for Blaise Pascal, who pioneered calculating machines in the seventeenth century. It is a simple, structured, general-purpose language primarily for teaching programming.
Python	A simple general-purpose scripting language good for writing short programs.
Visual Basic	Visual Basic was developed by Microsoft and it enables the programmers to rapidly develop graphical user interfaces.

Interpreting/Compiling Source Code

A program written in a high-level language is called a *source program* or *source code*. Because a computer cannot understand a source program, a source program must be translated into machine code for execution. The translation can be done using another programming tool called an *interpreter* or a *compiler*.

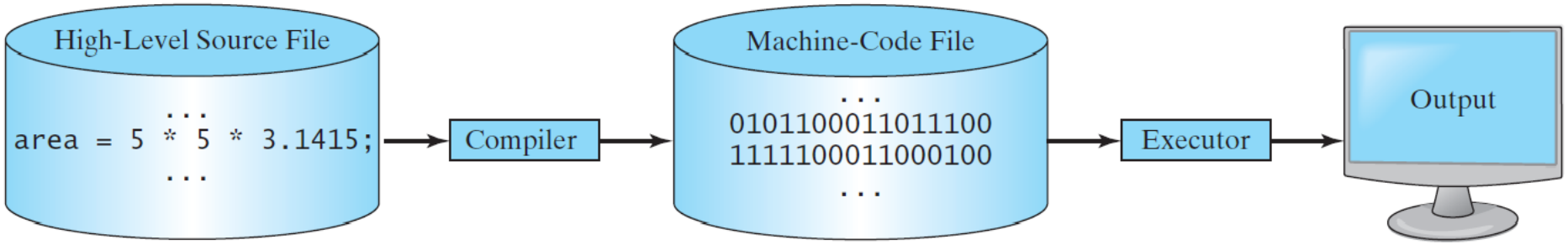
Interpreting Source Code

An interpreter reads one statement from the source code, translates it to the machine code or virtual machine code, and then executes it right away, as shown in the following figure. Note that a statement from the source code may be translated into several machine instructions.



Compiling Source Code

A compiler translates the entire source code into a machine-code file, and the machine-code file is then executed, as shown in the following figure.



What is an Operating System?

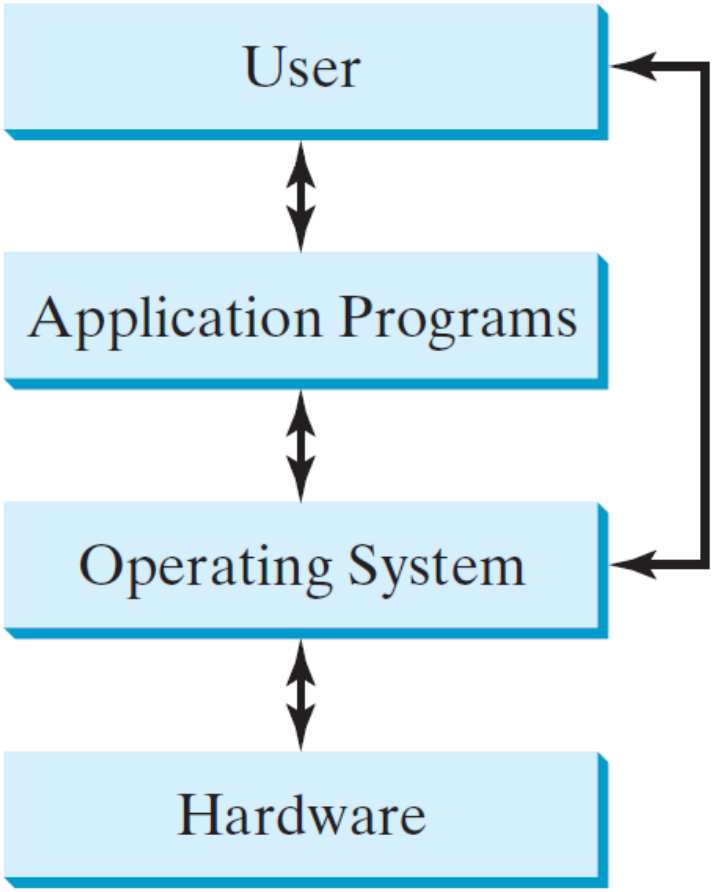
- OS monitors overall activity of the computer and provides services
- Example services
  - Memory management
  - Input/output
  - Activities
  - Storage management
- Why powerful?
  - Services don’t have to be written by each application

Operating Systems

The *operating system* (OS) is a program that manages and controls a computer's activities.

The popular operating systems for general-purpose computers are Microsoft Windows, Mac OS, and Linux.

Application programs, such as a Web browser or a word processor, cannot run unless an operating system is installed and running on the computer.



Why Java?

The answer is that Java enables users to develop and deploy applications on the Internet for servers, desktop computers, and small hand-held devices. The future of computing is being profoundly influenced by the Internet, and Java promises to remain a big part of that future. Java is the Internet programming language.

- Java is a general purpose programming language.
- Java is the Internet programming language.

Java, Web, and Beyond

- Java can be used to develop standalone applications.
- Java can be used to develop applications running from a browser.
- Java can also be used to develop applications for hand-held devices.
- Java can be used to develop applications for Web servers.

Characteristics of Java

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

www.cs.armstrong.edu/liang/JavaCharacteristics.pdf

Characteristics of Java

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

Java is partially modeled on C++, but greatly simplified and improved. Some people refer to Java as "C++--" because it is like C++ but with more functionality and fewer negative aspects.

Characteristics of Java

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

Java is inherently object-oriented. Although many object-oriented languages began strictly as procedural languages, Java was designed from the start to be object-oriented. Object-oriented programming (OOP) is a popular programming approach that is replacing traditional procedural programming techniques.

One of the central issues in software development is how to reuse code. Object-oriented programming provides great flexibility, modularity, clarity, and reusability through encapsulation, inheritance, and polymorphism.

Characteristics of Java

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

Distributed computing involves several computers working together on a network. Java is designed to make distributed computing easy. Since networking capability is inherently integrated into Java, writing network programs is like sending and receiving data to and from a file.

Characteristics of Java

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

You need an interpreter to run Java programs. The programs are compiled into the Java Virtual Machine code called bytecode. The bytecode is machine-independent and can run on any machine that has a Java interpreter, which is part of the Java Virtual Machine (JVM).

Characteristics of Java

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

Java compilers can detect many problems that would first show up at execution time in other languages.

Java has eliminated certain types of error-prone programming constructs found in other languages.

Java has a runtime exception-handling feature to provide programming support for robustness.

Characteristics of Java

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

Java implements several security mechanisms to protect your system against harm caused by stray programs.

Characteristics of Java

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

Write once, run anywhere

With a Java Virtual Machine (JVM), you can write one program that will run on any platform.

Characteristics of Java

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

Because Java is architecture neutral, Java programs are portable. They can be run on any platform without being recompiled.

Characteristics of Java

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

Java's performance is sometimes criticized. The execution of the bytecode is never as fast as it would be with a compiled language, such as C++. Because Java is interpreted, the bytecode is not directly executed by the system, but is run through the interpreter. However, its speed is more than adequate for most interactive applications, where the CPU is often idle, waiting for input or for data from other sources.

CPU speed has increased dramatically in the past few years, and this trend will continue. There are many ways to improve performance. Users of the earlier Sun Java Virtual Machine certainly noticed that Java was slow. However, the new JVM is significantly faster. The new JVM uses the technology known as *just-in-time compilation*. It compiles bytecode into native machine code, stores the native code, and reinvokes the native code when its bytecode is executed. Sun recently developed the Java HotSpot Performance Engine, which includes a compiler for optimizing the frequently used code. The HotSpot Performance Engine can be plugged into a JVM to dramatically boost its performance.

http://www.cs.armstrong.edu/liang/JavaCharacteristics.pdf

Characteristics of Java

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

Multithread programming is smoothly integrated in Java, whereas in other languages you have to call procedures specific to the operating system to enable multithreading.

Characteristics of Java

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

Java was designed to adapt to an evolving environment. New code can be loaded on the fly without recompilation. There is no need for developers to create, and for users to install, major new software versions. New features can be incorporated transparently as needed.