Chapter 1: Computer Systems

Presentation slides for

Java Software Solutions

Foundations of Program Design

Second Edition

by John Lewis and William Loftus

Java Software Solutions is published by Addison-Wesley-Longman

Focus of the Course

- **Object-Oriented Software Development**

  - **problem solving**
  - **program design and implementation**
  - **object-oriented concepts**
    - **objects**
    - **classes**
    - **interfaces**
    - **inheritance**
    - **polymorphism**
  - **graphics and Graphical User Interfaces**
  - **the Java programming language**

Computer Systems

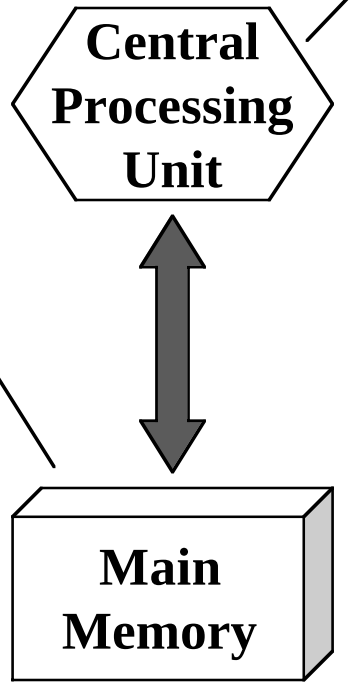
- **We first need to explore the fundamentals of computer processing**

- **Chapter 1 focuses on:**
  - **components of a computer**
  - **how those components interact**
  - **how computers store and manipulate information**
  - **computer networks**
  - **the Internet and the World-Wide Web**
  - **programming and programming languages**
  - **graphic systems**

Hardware and Software

- **Hardware**
  - **the physical, tangible parts of a computer**
  - **keyboard, monitor, wires, chips, data**
- **Software**
  - **programs and data**
  - **a *program* is a series of instructions**
- **A computer requires both hardware and software**
- **Each is essentially useless without the other**

CPU and Main Memory



Chip that executes program commands

Intel Pentium III
Sun Sparc Processor

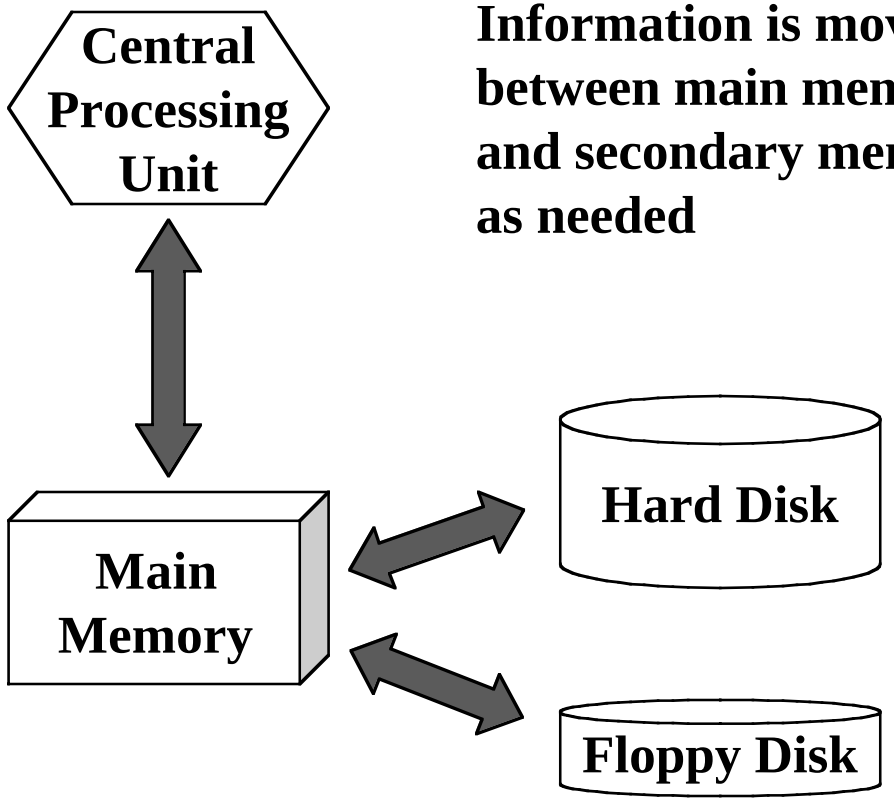
Primary storage area for programs and data that are in active use

Synonymous with RAM

Secondary Memory Devices

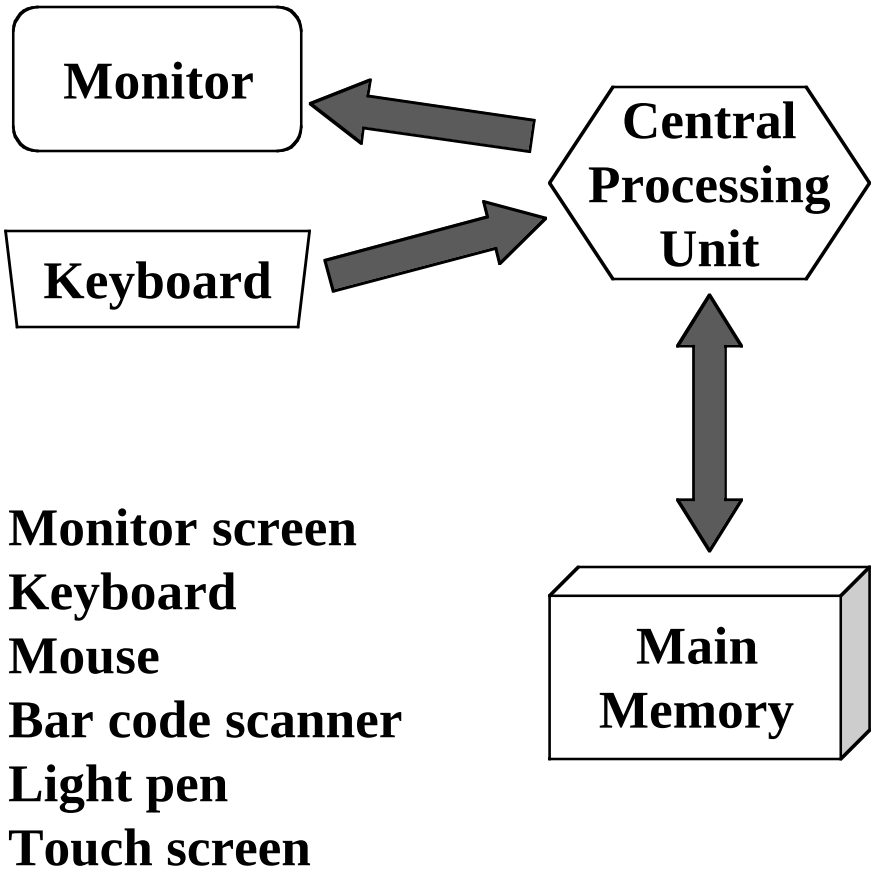
Secondary memory devices provide long-term storage

Information is moved between main memory and secondary memory as needed



Hard disks
Floppy disks
ZIP disks
Writable CDs
Tapes

Input / Output Devices



I/O devices allow user interaction

Monitor screen
Keyboard
Mouse
Bar code scanner
Light pen
Touch screen

Software Categories

- **Operating System**
  - **controls all machine activities**
  - **provides the user interface to the computer**
  - **manages resources such as the CPU and memory**
  - **Windows 98, Windows NT, Unix, Linux, Mac OS**
- **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)**

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 voltage sent to the speaker**

- ***Digital***
  - **the information is broken down into pieces, and each piece is represented separately**
  - **music on a compact disc - the disc stores numbers representing specific voltage levels sampled at various points**

Digital Information

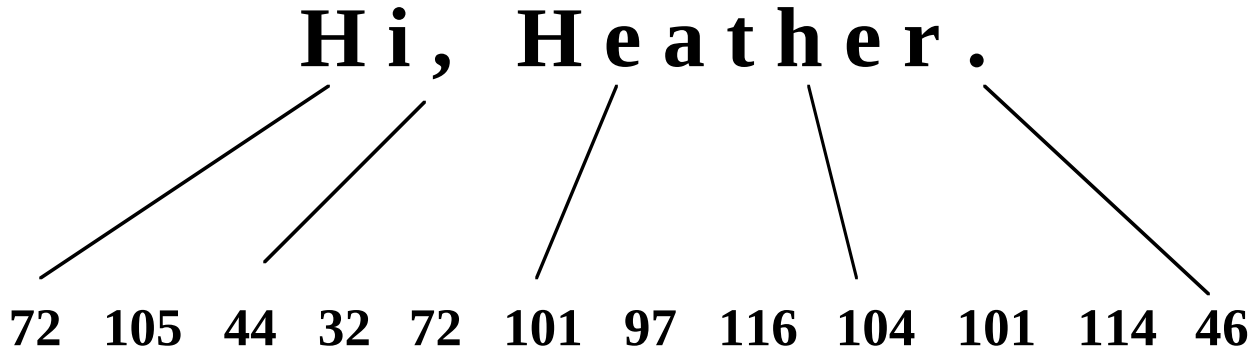
- **Computers store all information digitally:**
  - **numbers**
  - **text**
  - **graphics and images**
  - **audio**
  - **video**
  - **program instructions**

- **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**



Binary Numbers

- **Once information is 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 only have to represent two states**
- **A single bit can represent two possible states, like a light bulb that is either on (1) or off (0)**
- **Combinations of bits are used to store values**

Bit Combinations

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 combinations

Bit Combinations

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

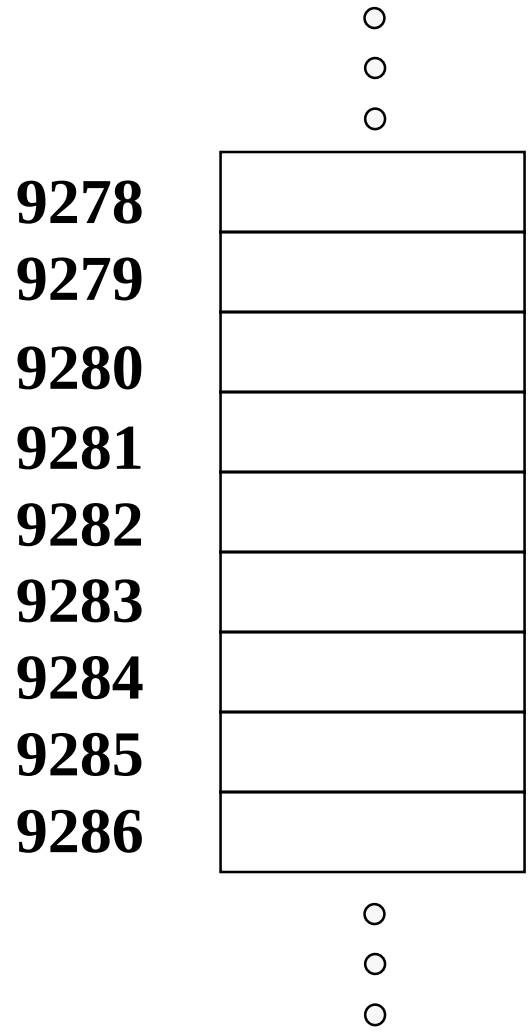
How many items can be represented by

- 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

A Computer Specification

- **Consider the following specification for a personal computer:**
  - **600 MHz Pentium III Processor**
  - **256 MB RAM**
  - **16 GB Hard Disk**
  - **24x speed CD ROM Drive**
  - **17” Multimedia Video Display with 1280 x 1024 resolution**
  - **56 KB Modem**
- **What does it all mean?**

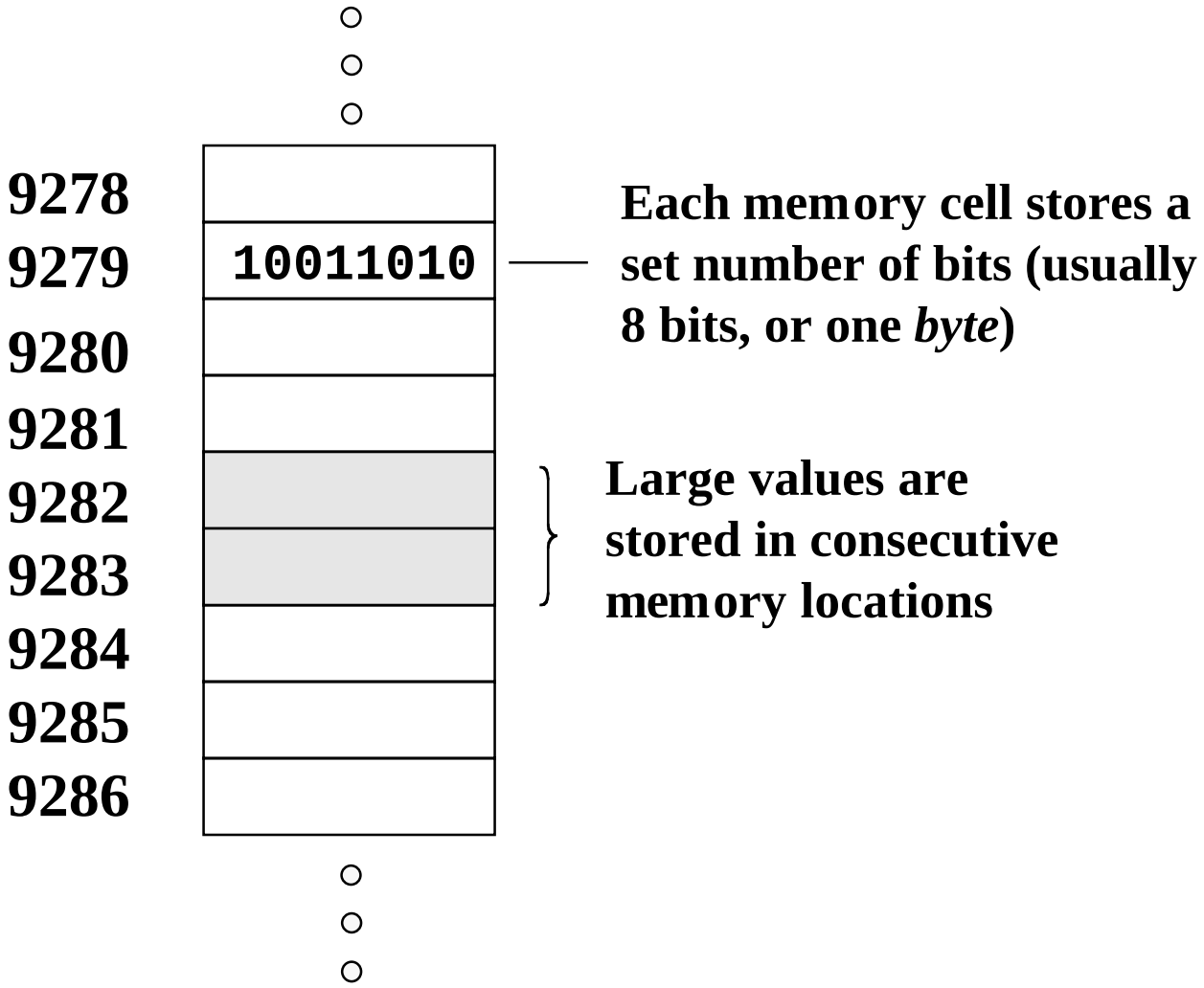
Memory



Main memory is divided into many memory locations (or *cells*)

Each memory cell has a numeric *address*, which uniquely identifies it

Storing Information



Storage Capacity

- **Every memory device has a *storage capacity*, indicating the number of bytes it can hold**
- **Capacities are expressed in various units:**

Unit	Symbol	Number of Bytes
kilobyte	KB	2^{10} = 1024
megabyte	MB	2^{20} (over 1 million)
gigabyte	GB	2^{30} (over 1 billion)
terabyte	TB	2^{40} (over 1 trillion)

Memory

- **Main memory is *volatile* - stored information is lost if the electric power is removed**
- **Secondary memory devices are *nonvolatile***

- **Main memory and disks are *direct access* devices - information can be reached directly**
- **The terms direct access and *random access* are often used interchangeably**
- **A magnetic tape is a *sequential access* device since its data is arranged in a linear order - you must get by the intervening data in order to access other information**

RAM vs. ROM

- ***RAM*** **- Random Access Memory (direct access)**
- ***ROM*** **- Read-Only Memory**

- **The terms RAM and main memory are basically interchangeable**

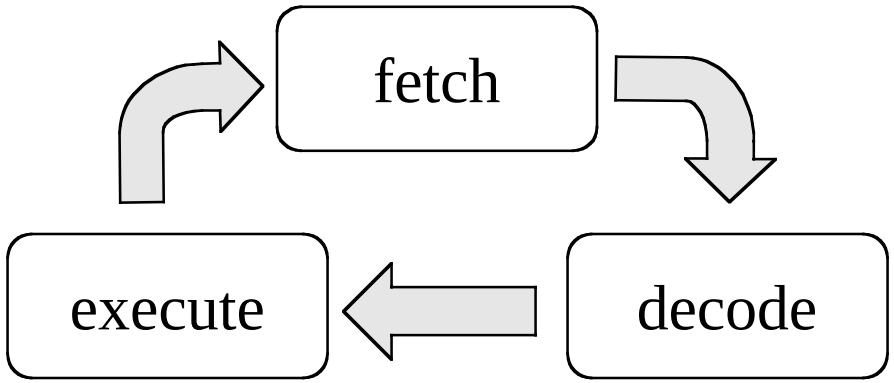
- **ROM could be a set of memory chips, or a separate device, such as a CD ROM**

- **Both RAM and ROM are random (direct) access devices!**
- **RAM should probably be called Read-Write Memory**

The Central Processing Unit

- **A CPU is also called a *microprocessor***
- **It continuously follows the *fetch-decode-execute cycle:***

Retrieve an instruction from main memory

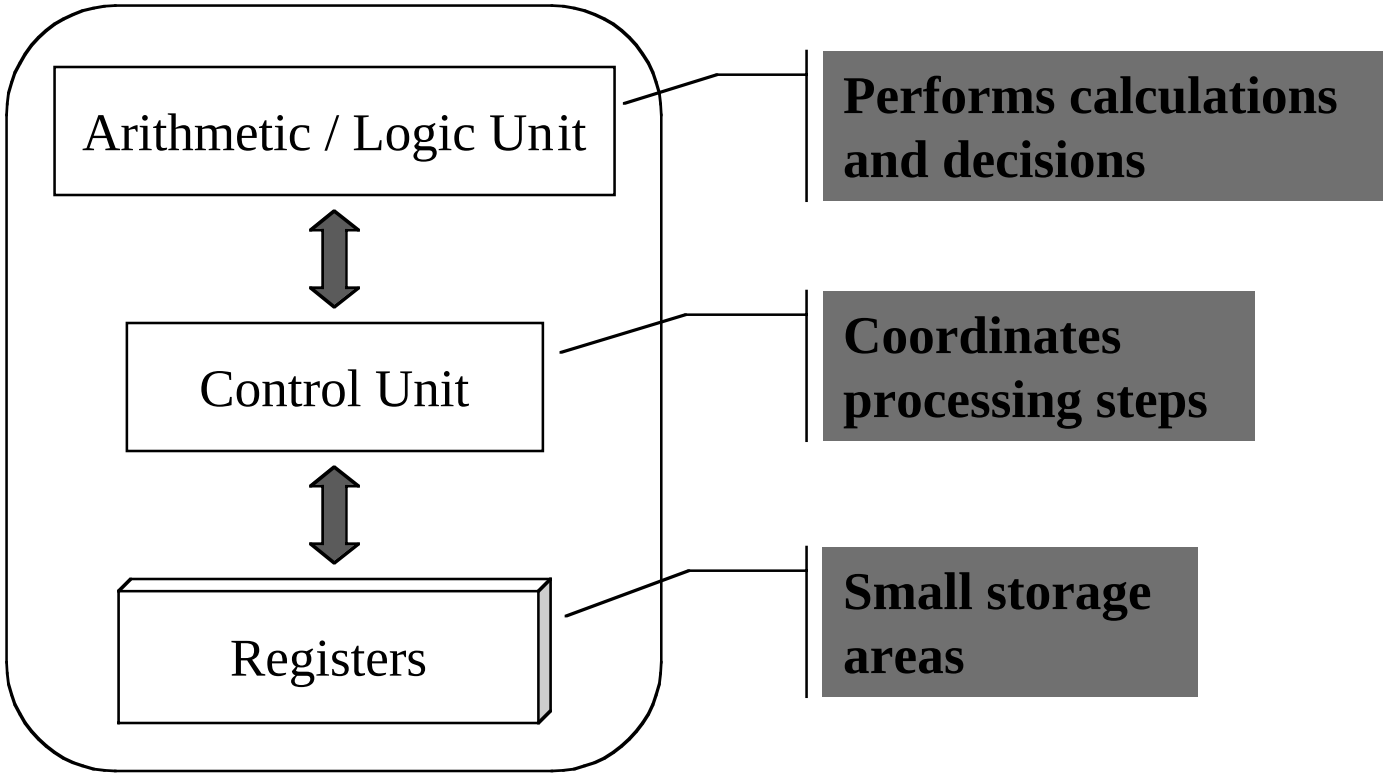


Carry out the instruction

Determine what the instruction is

The Central Processing Unit (CPU)

- **The CPU contains:**



The Central Processing Unit

- **The speed of a CPU is controlled by the *system clock***
- **The system clock generates an electronic pulse at regular intervals**
- **The pulses coordinate the activities of the CPU**
- **The speed is measured in *megahertz* (MHz)**

Monitor

- **The size of a monitor (17") is measured diagonally, like a television screen**
- **Most monitors these days have *multimedia* capabilities: text, graphics, video, etc.**
- **A monitor has a certain maximum *resolution* , indicating the number of picture elements, called *pixels*, that it can display (such as 1280 by 1024)**
- **High resolution (more pixels) produces sharper pictures**

Modem

- ***Data transfer devices*** **allow information to be sent and received between computers**
- **Many computers include a *modem*, which allows information to be moved across a telephone line**
- **A data transfer device has a maximum *data transfer rate***
- **A modem, for instance, may have a data transfer rate of 56,000 *bits per second* (bps)**

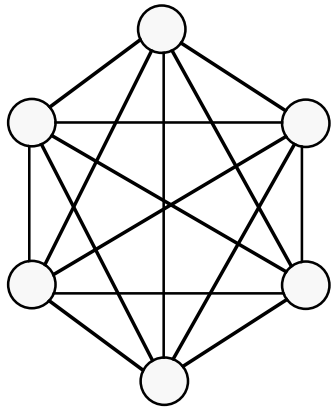
Networks

- **A *network* is two or more computers that are connected so that data and resources can be shared**
- **Most computers are connected to some kind of network**
- **Each computer has its own *network address*, which uniquely identifies it among the others**
- **A *file server* is a network computer dedicated to storing programs and data that are shared among network users**

Network Connections

- **Each computer in a network could be directly connected to each other computer in the network**
- **These are called *point-to-point* connections**

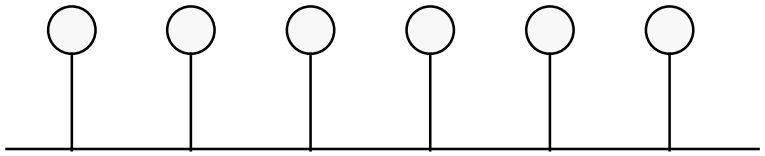
Adding a computer requires a new communication line for each computer already in the network



This technique is not feasible for more than a few close machines

Network Connections

- **Most modern networks share a single communication line**
- **Adding a new computer to the network is relatively easy**

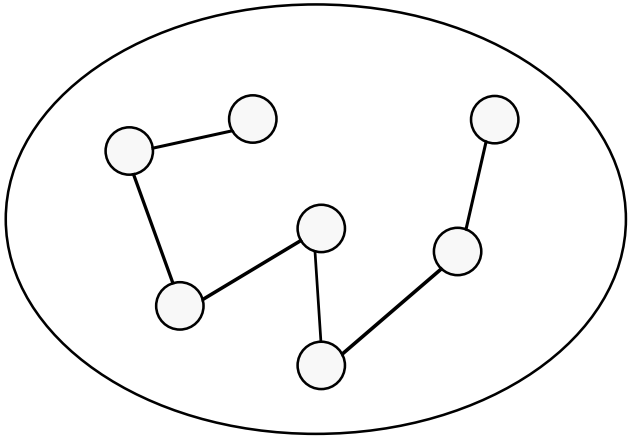


Network traffic must take turns using the line, which introduces delays

Often information is broken down in parts, called *packets*, which are sent to the receiving machine then reassembled

Local-Area Networks

A *Local-Area Network* (LAN) covers a small distance and a small number of computers

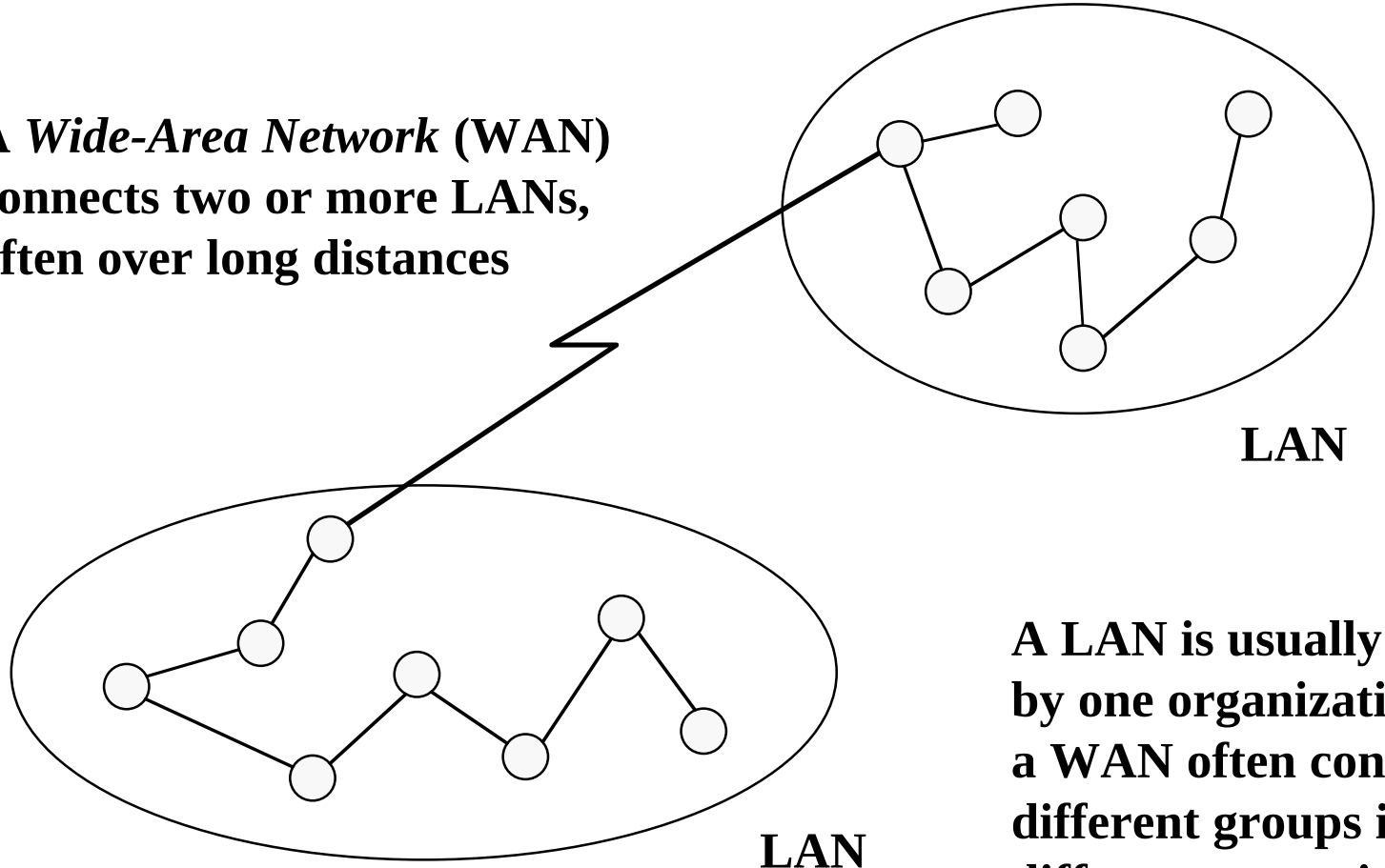


LAN

A LAN often connects the machines in a single room or building

Wide-Area Networks

A *Wide-Area Network* (WAN) connects two or more LANs, often over long distances



A LAN is usually owned by one organization, but a WAN often connects different groups in different countries

The Internet

- **The *Internet* is a WAN which spans the entire planet**
- **The word Internet comes from the term *internetworking*, which implies communication among networks**
- **It started as a United States government project, sponsored by the Advanced Research Projects Agency (ARPA), and was originally called the ARPANET**
- **The Internet grew quickly throughout the 1980s and 90s**
- **Less than 600 computers were connected to the Internet in 1983; now there are over 10 million**

TCP/IP

- **A protocol is a set of rules that determine how things communicate with each other**
- **The software which manages Internet communication follows a suite of protocols called *TCP/IP***
- **The *Internet Protocol* (IP) determines the format of the information as it is transferred**
- **The *Transmission Control Protocol* (TCP) dictates how messages are reassembled and handles lost information**

IP and Internet Addresses

- **Each computer on the Internet has a unique *IP address*, such as:**

  `204.192.116.2`

- **Most computers also have a unique Internet name, which is also referred to as an *Internet address*:**

  `renoir.villanova.edu`
  `kant.breakaway.com`

- **The first part indicates a particular computer (`renoir`)**
- **The rest is the *domain name*, indicating the organization (`villanova.edu`)**

Domain Names

- **The last section (the suffix) of each domain name usually indicates the type of organization:**

`edu`	-	**educational institution**
`com`	-	**commercial business**
`org`	-	**non-profit organization**
`net`	-	**network-based organization**

Sometimes the suffix indicates the country:

`uk`	-	**United Kingdom**
`au`	-	**Australia**
`ca`	-	**Canada**
`se`	-	**Sweden**

New suffix categories are being considered

Domain Names

- **A domain name can have several parts**
- **Unique domain names mean that multiple sites can have individual computers with the same local name**
- **When used, an Internet address is translated to an IP address by software called the *Domain Name System* (DNS)**
- **There is no one-to-one correspondence between the sections of an IP address and the sections of an Internet address**

The World-Wide Web

- **The *World-Wide Web* allows many different types of information to be accessed using a common interface**
- **A *browser* is a program which accesses and presents information**
  - **text, graphics, sound, audio, video, executable programs**
- **A Web document usually contains *links* to other Web documents, creating a *hypermedia* environment**
- **The term Web comes from the fact that information is not organized in a linear fashion**

The World-Wide Web

- **Web documents are often defined using the *HyperText Markup Language* (HTML)**

- **Information on the Web is found using a *Uniform Resource Locator* (URL):**

```
http://www.lycos.com
http://www.villanova.edu/webinfo/domains.html
ftp://java.sun.com/applets/animation.zip
```

- **A URL indicates a protocol (http), a domain, and possibly specific documents**

Problem Solving

- **The purpose of writing a program is to solve a problem**

- **The general steps in problem solving are:**

  - **Understand the problem**
  - **Dissect the problem into manageable pieces**
  - **Design a solution**
  - **Consider alternatives to the solution and refine it**
  - **Implement the solution**
  - **Test the solution and fix any problems that exist**

Problem Solving

- **Many software projects fail because the developer didn't really understand the problem to be solved**
- **We must avoid assumptions and clarify ambiguities**

- **As problems and their solutions become larger, we must organize our development into manageable pieces**
- **This technique is fundamental to software development**
- **We will dissect our solutions into pieces called classes and objects, taking an *object-oriented approach***

The Java Programming Language

- A *programming language* specifies the words and symbols that we can use to write a program
- A programming language employs a set of rules that dictate how the words and symbols can be put together to form valid *program statements*
- Java was created by Sun Microsystems, Inc.
- It was introduced in 1995 and has become quite popular
- It is an object-oriented language

Java Program Structure

- **In the Java programming language:**
  - **A program is made up of one or more *classes***
  - **A class contains one or more *methods***
  - **A method contains program *statements***

- **These terms will be explored in detail throughout the course**

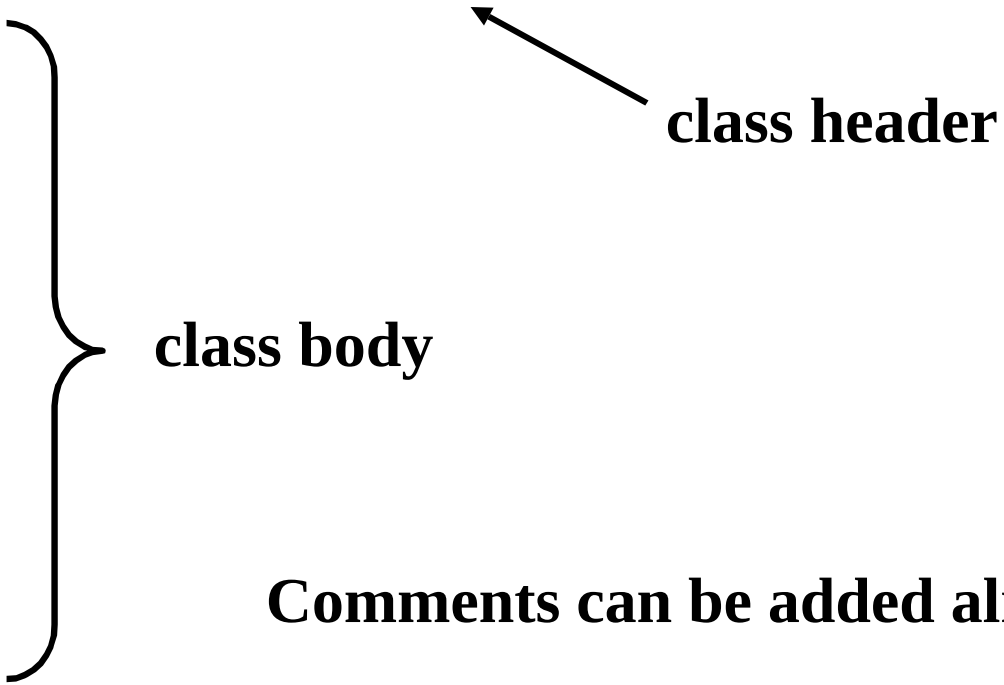
- **A Java application always contains a method called `main`**

- **See `Lincoln.java` (page 26)**

Java Program Structure

```
//  comments about the class
public class MyProgram
{


}
```

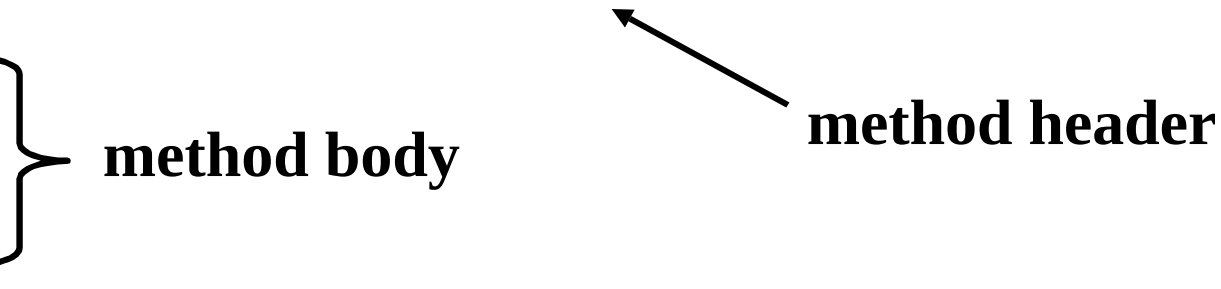


Java Program Structure

```
//  comments about the class
public class MyProgram
{

   //  comments about the method
   public static void main (String[] args)
   {
   }

}
```



Comments

- **Comments in a program are also called *inline documentation***
- **They should be included to explain the purpose of the program and describe processing steps**
- **They do not affect how a program works**
- **Java comments can take two forms:**

```
// this comment runs to the end of the line


/*  this comment runs to the terminating
    symbol, even across line breaks       */
```

Identifiers

- ***Identifiers*** **are the words a programmer uses in a program**
- **An identifier can be made up of letters, digits, the underscore character (_), and the dollar sign**
- **They cannot begin with a digit**
- **Java is *case sensitive*, therefore `Total` and `total` are different identifiers**

Identifiers

- **Sometimes we choose identifiers ourselves when writing a program (such as `Lincoln`)**
- **Sometimes we are using another programmer's code, so we use the identifiers that they chose (such as `println`)**
- **Often we use special identifiers called *reserved words* that already have a predefined meaning in the language**
- **A reserved word cannot be used in any other way**

Reserved Words

- **The Java reserved words:**

abstract	default	goto	operator	synchronized
boolean	do	if	outer	this
break	double	implements	package	throw
byte	else	import	private	throws
byvalue	extends	inner	protected	transient
case	false	instanceof	public	true
cast	final	int	rest	try
catch	finally	interface	return	var
char	float	long	short	void
class	for	native	static	volatile
const	future	new	super	while
continue	generic	null	switch	

White Space

- Spaces, blank lines, and tabs are collectively called *white space*
- White space is used to separate words and symbols in a program
- Extra white space is ignored
- A valid Java program can be formatted many different ways
- Programs should be formatted to enhance readability, using consistent indentation
- See `Lincoln2.java` and `Lincoln3.java`

Programming Language Levels

- **There are four programming language levels:**
  - **machine language**
  - **assembly language**
  - **high-level language**
  - **fourth-generation language**

- **Each type of CPU has its own specific *machine language***

- **The other levels were created to make it easier for a human being to write programs**

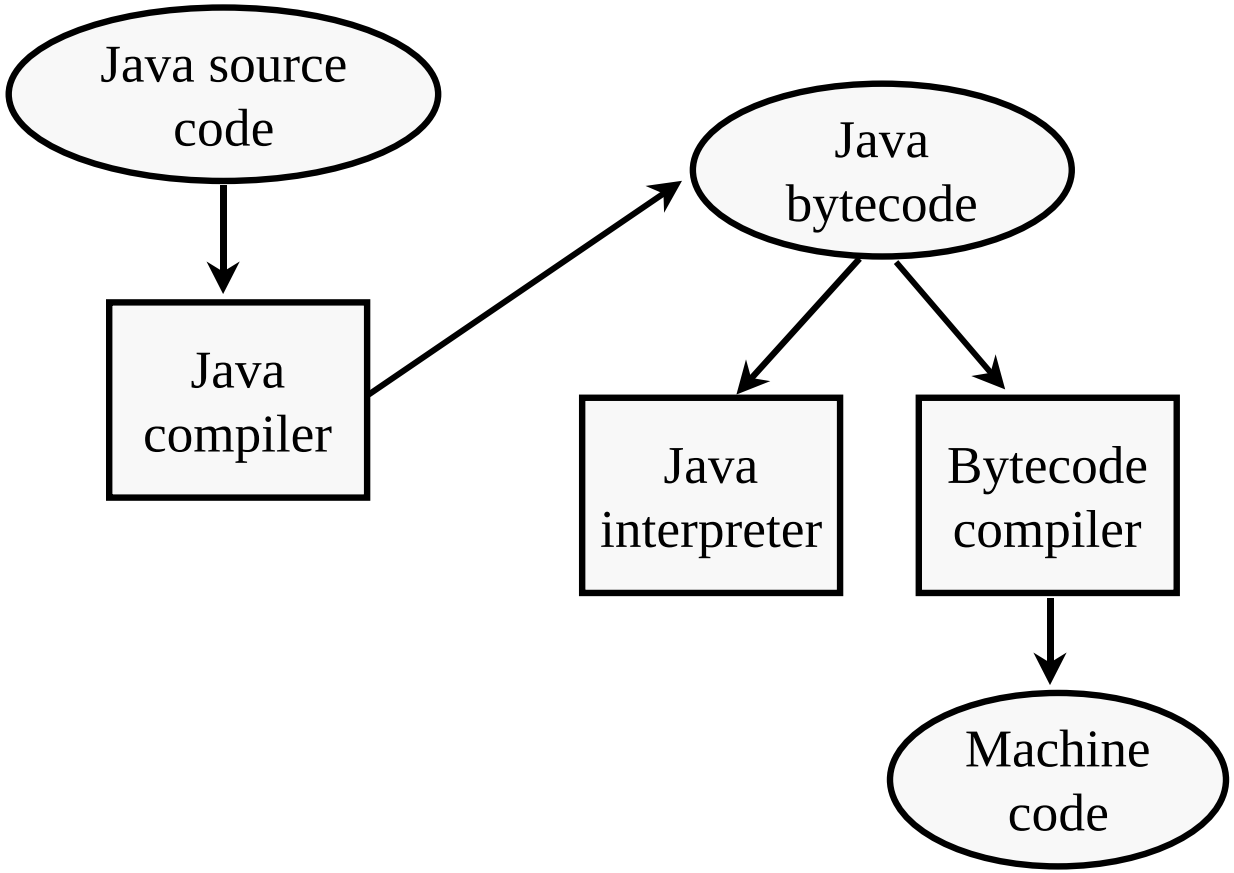
Programming Languages

- **A program must be translated into machine language before it can be executed on a particular type of CPU**
- **This can be accomplished in several ways**
- **A *compiler* is a software tool which translates *source code* into a specific target language**
- **Often, that target language is the machine language for a particular CPU type**
- **The Java approach is somewhat different**

Java Translation and Execution

- **The Java compiler translates Java source code into a special representation called *bytecode***
- **Java bytecode is not the machine language for any traditional CPU**
- **Another software tool, called an *interpreter*, translates bytecode into machine language and executes it**
- **Therefore the Java compiler is not tied to any particular machine**
- **Java is considered to be *architecture-neutral***

Java Translation and Execution



Development Environments

- **There are many development environments which develop Java software:**
  - **Sun Java Software Development Kit (SDK)**
  - **Borland JBuilder**
  - **MetroWork CodeWarrior**
  - **Microsoft Visual J++**
  - **Symantec Café**

- **Though the details of these environments differ, the basic compilation and execution process is essentially the same**

Syntax and Semantics

- **The *syntax rules* of a language define how we can put symbols, reserved words, and identifiers together to make a valid program**
- **The *semantics* of a program statement define what that statement means (its purpose or role in a program)**
- **A program that is syntactically correct is not necessarily logically (semantically) correct**
- **A program will always do what we tell it to do, not what we meant to tell it to do**

Errors

- **A program can have three types of errors**
- **The compiler will find problems with syntax and other basic issues (*compile-time errors*)**
  - **If compile-time errors exist, an executable version of the program is not created**
- **A problem can occur during program execution, such as trying to divide by zero, which causes a program to terminate abnormally (*run-time errors*)**
- **A program may run, but produce incorrect results (*logical errors*)**

Introduction to Graphics

- **The last one or two sections of each chapter of the textbook focus on graphical issues**
- **Most computer programs have graphical components**
- **A picture or drawing must be digitized for storage on a computer**
- **A picture is broken down into pixels, and each pixel is stored separately**

Representing Color

- **A black and white picture can be stored using one bit per pixel (0 = white and 1 = black)**
- **A color picture requires more information, and there are several techniques for representing a particular color**
- **For example, every color can be represented as a mixture of the three primary colors Red, Green, and Blue**
- **In Java, each color is represented by three numbers between 0 and 255 that are collectively called an *RGB value***

Coordinate Systems

- Each pixel can be identified using a two-dimensional coordinate system
- When referring to a pixel in a Java program, we use a coordinate system with the origin in the upper left corner