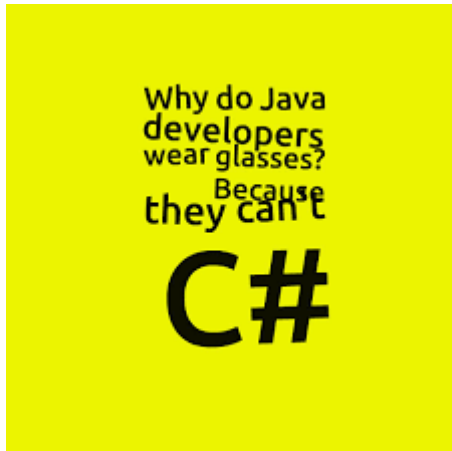
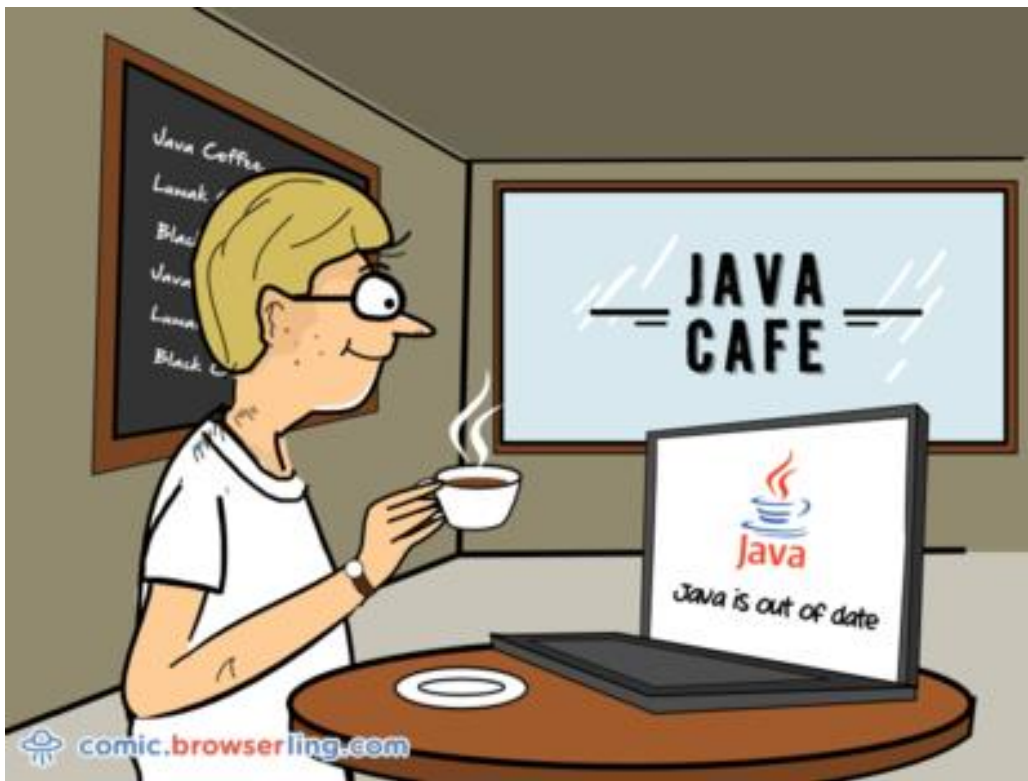




# CSS 142 A

## Lecture 2

Arkady Retik      aretik@uw.edu




## The difference between coding academic Java and enterprise Java

**ACADEMIA:**  
MY GOD ... THIS WILL MEAN A HALF-DOZEN PAPERS, A THESIS OR TWO, AND A PARAGRAPH IN EVERY TEXTBOOK ON QUEUING THEORY!

**BUSINESS:**  
YOU GOT THE PROGRAM TO STOP JAMMING UP? GREAT. WHILE YOU'RE FIXING STUFF, CAN YOU GET OUTLOOK TO SYNC WITH OUR NEW PHONES?

I JUST WROTE THE MOST BEAUTIFUL CODE OF MY LIFE.

THEY CASUALLY HANDED ME AN IMPOSSIBLE PROBLEM. IN 48 HOURS AND 200 LINES, I SOLVED IT.

 **Java**AssignmentHelp.net

## Top stories



Oracle Wins Latest Round vs. Google in Java Copyright Case | WIRED

WIRED

18 hours ago



Java-aaaargh! Google faces \$9bn copyright bill after Oracle scores 'fair use' court appeal...

The Register

19 hours ago



Oracle Wins Court Ruling Against Google in Multibillion-Dollar Copyright Case

Wall Street Journal

17 hours ago

# TODAY'S CONTENT

## 1. Recap L1 ; Introduction to Computer Organization

- ❖ Affinity Exercise results
- ❖ Java and Computer Organization
- ❖ Memory; Binary System



## 2. Programming in Java

## 3. Printing, assignments, statements

- ❖ Continue....



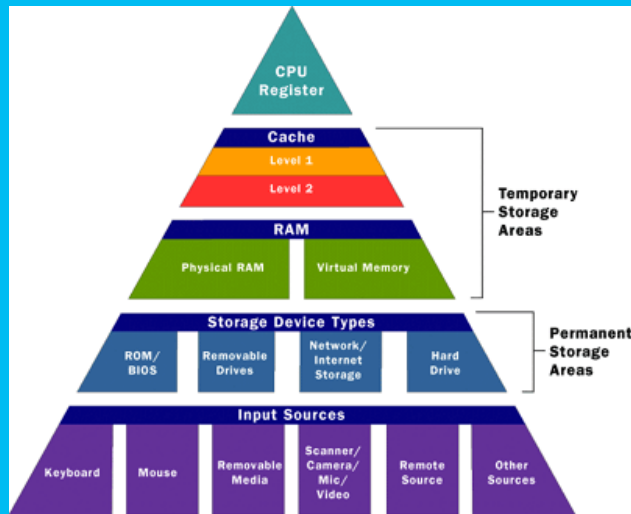
Will start today time permitting...

## 4. Next lecture: Reading 1.1, 1.2, **1.3**, 1.4; **2.1**

- ❖ **Class Activity (pairs) => bring paper and pencil/s**

# 1. Recap

continue..





# What do you **expect** / **want** from this course?

## Affinity exercise:

**5 minutes:** Write down your input (1 per post-it)

= < 3 *per person*

**1 minutes:** Place post-its on board

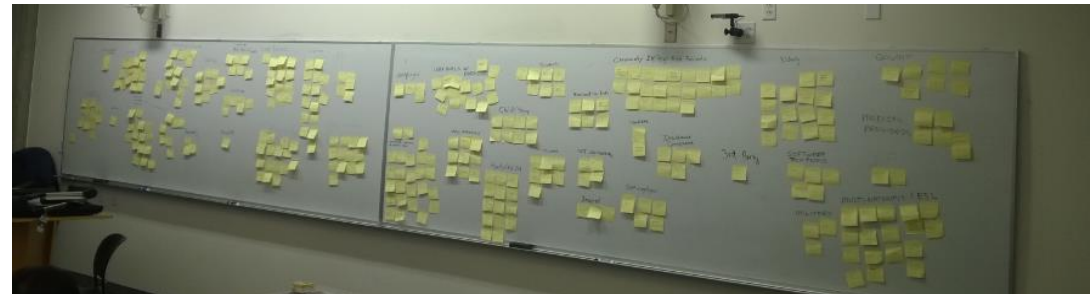
**3 minutes:** Group post-its to

**2 minutes:** Name groups (regroup as needed)

**1 minutes:** Add more post-its (regroup as needed)

**3 minutes:** Capture results

**RESULTS**



I would like to understand what it takes to go from programming in an IDE to making something like a phone app or a game  
 I want to know how to program communication devices  
 I want a 4.0 and to get the necessary base of knowledge to succeed in CSSE program  
 I want 4.0  
 Understand code well enough to move forward successfully (learn skills for long term success)  
 I would like to have a better understanding of computer programming

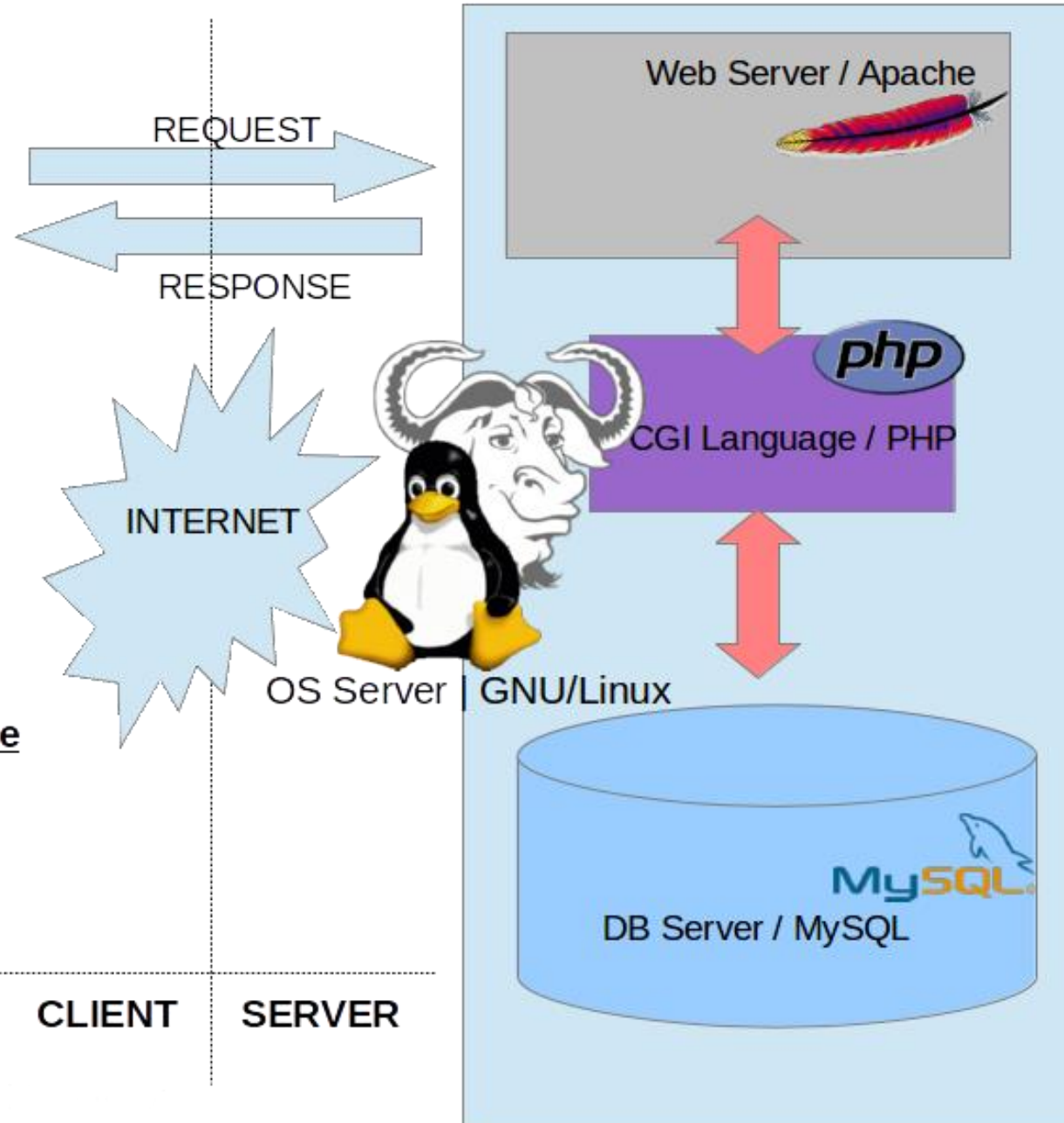
1. Master the language Java
2. Get at least 3.9
3. ~~Ite??~~ Foundation for programming

Be able to use Java well enough to list it on my resume  
 To have a better understanding of how computer software works  
 I want to learn how to efficiently code in Java and develop my Github Account  
 I want to learn the basics of programming  
 Set up for syntax success for web development/games  
 I want to be able to retain Java for the long term + apply it in my future career.  
 A good understanding of how to program with Java  
 Understand how computers and programs are built and organized  
 I want to study Artificial Intelligence  
 I want to learn everything about Classes

- > Dev in Linux
- > Bot dev. For games
- > pipeline for software dev., personal experiences working with computers, advice for being more efficient
- > ~3.8

1. Can skilled use Java
2. Can using the program to write a web pages
3. Mini program

Best way to study for exams?  
 I want to have the skills to be able to make a webpage  
 I expect to learn fundamentals thoroughly, so I am able to solve all the problems on the exam  
 A foundation in general programming  
 Write basic programs/applets  
 To learn basic operations on the computer & be able to fix bugs if possible  
 4.0  
 understanding what I'm doing  
 I want to also learn about the basis of sequel  
 I can learn basic things about computer programing and apply it  
 Want to more about programming  
 I want to learn more about complex functions  
 Programming a simple program  
 More info on software ethics  
 I wanna learn about basic knowledge of computer programming  
 I want 3.5-4.0 GPA  
 Understanding universal terminology used through software dev  
 Programming foundation (able to program on my own)  
 Through studying & practice I want to understand everything (or most) that will be taught this quarter  
 To learn Java more efficiently and to be able to be sufficient in coding  
 Achieve and acknowledge methods of constructing interfaces to function mechanics  
 Understand coding enough to make a game in Unity  
 Understand study routines  
 To code fast & clearly, like it's a second language



### LAMP Architecture

- Linux - OS
- Apache - Web
- MySQL - DB
- PHP - Script







Home > Quantitative Skills Center

## University of Washington Bothell Quantitative Skills Center

### Quantitative Skills Center

Tutoring

Hours & Schedules

QSC Team

Join Our Team!

Faculty & Staff Support

Student Resources and  
Equipment Checkout

Frequently Asked Questions

Questions, Comments or  
Suggestions

**The QSC promotes quantitative reasoning at UW  
Bothell through peer tutoring and faculty  
development.**

### Contact Info and Hours

#### Contact Info

**QSC Location:** UW2-030 between Common  
Grounds Café and Discovery Hall

**CSS Labs Location:** UW1-310 & 320

**Phone:** (425) 352-3170

**Email:** [uwbqsc@uw.edu](mailto:uwbqsc@uw.edu)

**Manager:** Tammy Wright

**Email:** [tswright@uw.edu](mailto:tswright@uw.edu)

#### Spring 2018 Hours (March 26 - June 1)

Mon - Thurs	9am - 8pm
Fri	10am - 4pm
Sun	12pm - 4pm

#### \*Finals Week Hours\* (June 4 - June 7)

Mon - Wed	9am - 6pm
Thurs	9am - 2pm
Fri - Sun	Closed

<http://www.uwb.edu/qsc>



<http://www.uwb.edu/qsc/schedule>

- » Biology Schedule
- » Business Schedule
- » Chemistry Schedule
- » **CSS Schedule**
- » EE Schedule
- » Math Schedule
- » Mech E Schedule
- » Physics Schedule
- » Environmental Science Schedule
- » Statistics & Probability Schedule
- » Software Schedule

QSC Team

Join Our Team!

Monday-Thursday 9am-8pm  
Friday 10am-4pm  
Sunday 12pm-4pm

\*The QSC will be closed for Martin Luther King Jr. Day

\*The QSC will be closed for Presidents Day



Biology »



Business »



CSS »



EE »



Physics »



Statistics & Probability »

## Students

We aim to help UW Bothell students develop skills and confidence with quantitative reasoning. We do this by offering FREE, **drop-in tutoring** and **online tutoring**. Our peer tutoring model involves asking a lot of questions, working through examples, and modeling the problem-solving process.

<http://www.uwb.edu/qsc/schedule/css>

Below is a list of the most common CSS courses with which course has a list of dates and times of coverage, and a list of works different hours within the overall coverage hours. To shift, please see the [time schedule](#), or the [staff page](#).

## CSS 142 Computer Programming I

Day	Times
Monday	None
Tuesday	11am - 6pm
Wednesday	12pm - 3pm
Thursday	11am - 5:30pm
Friday	10am - 4pm
Sunday	12pm - 4pm

Possible tutors:

- Alexander
- Ben
- Jason G
- Pengkun
- Polina

- **Java and Computer organization**
- **printing, assignment, simple statements.**

→ **Savitch 1.1; 1.2; 1.4**



We stopped here last time

15.6" DISPLAY  
16GB MEMORY  
1.0TB HARD DRIVE



## Computers



Do you understand all these terms?

This product is no longer available. Continue shopping at [Costco](#).

### Toshiba Satellite S55 Laptop • Intel Core i7 • Backlit Keyboard

Purchase the computer and a FREE 1 Year Kaspersky Internet Security (1-User) download will be automatically added at checkout. The software download will be emailed to you. State law may require sales tax to be charged on the pre-discounted price if the product is subject to sales tax. Processor & Memory: Intel® Core™ i7-4700MQ Processor 2.4 GHz (up to 3.40 GHz with Intel® Turbo Boost Technology) 16GB DDR3 1600MHz (max 16GB) Drives: 1TB (5400 RPM) Serial ATA hard disk drive DVD-RW (Writes to DVD/CD) Operating System: Microsoft® Windows 8 (64-bit) Graphics & Video: 15.6" diagonal widescreen TruBrite® TFT display at 1366 x 768 native resolution (HD) Native support for 720p content 16:9 aspect ratio Mobile Intel® HD graphics with shared graphics memory Communications: HD Webcam and microphone 10/100/1000 Ethernet Wi-Fi® Wireless networking (802.11b/g/n) Audio: Built-in Harmon/Kardon® stereo speakers DTS Studio Sound™ Keyboard: Premium Backlit Keyboard with 10-key (black) Touch pad pointing device with multi-touch control One Touch function keys I/O Ports & Slots: 3 USB ports (2 USB 3.0 + 1 USB 2.0 port with USB Sleep and Charge) RJ-45 LAN port Memory Card Reader Microphone input port Headphone output port HDMI® RGB output port Slot for Security Lock Power Supply: 4cell/43Wh Lithium Ion battery pack 90W (19V 4.74A) 100-240V/50-60Hz AC Adapter Additional Information: Dimensions: 15.2" W x 9.9" D x 1.1" H Approximate Weight: 5.3 lbs. Model: S55-A5236 / PSKK2U-006005





## Hardware



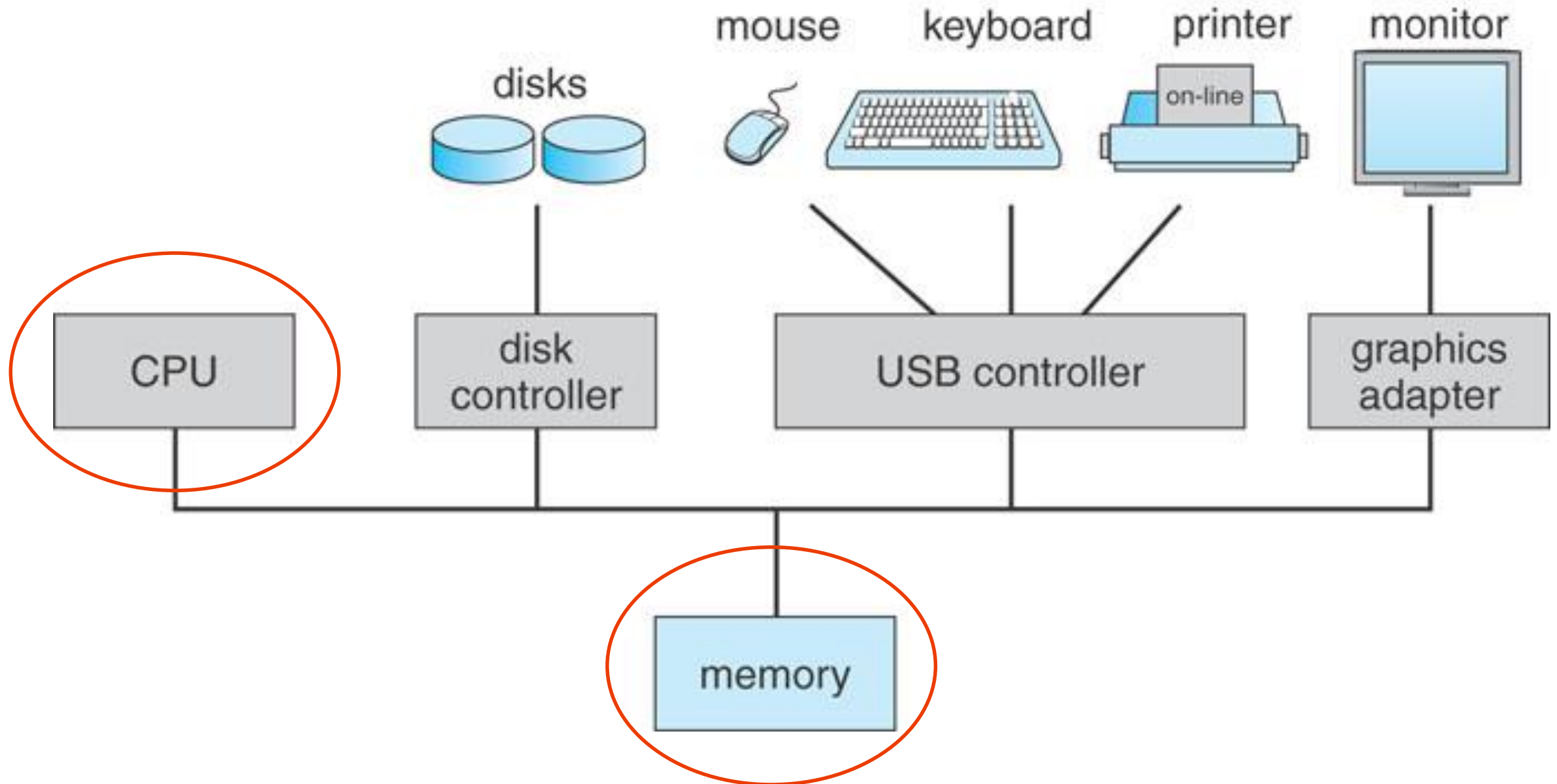
## Software



# Computer Hardware



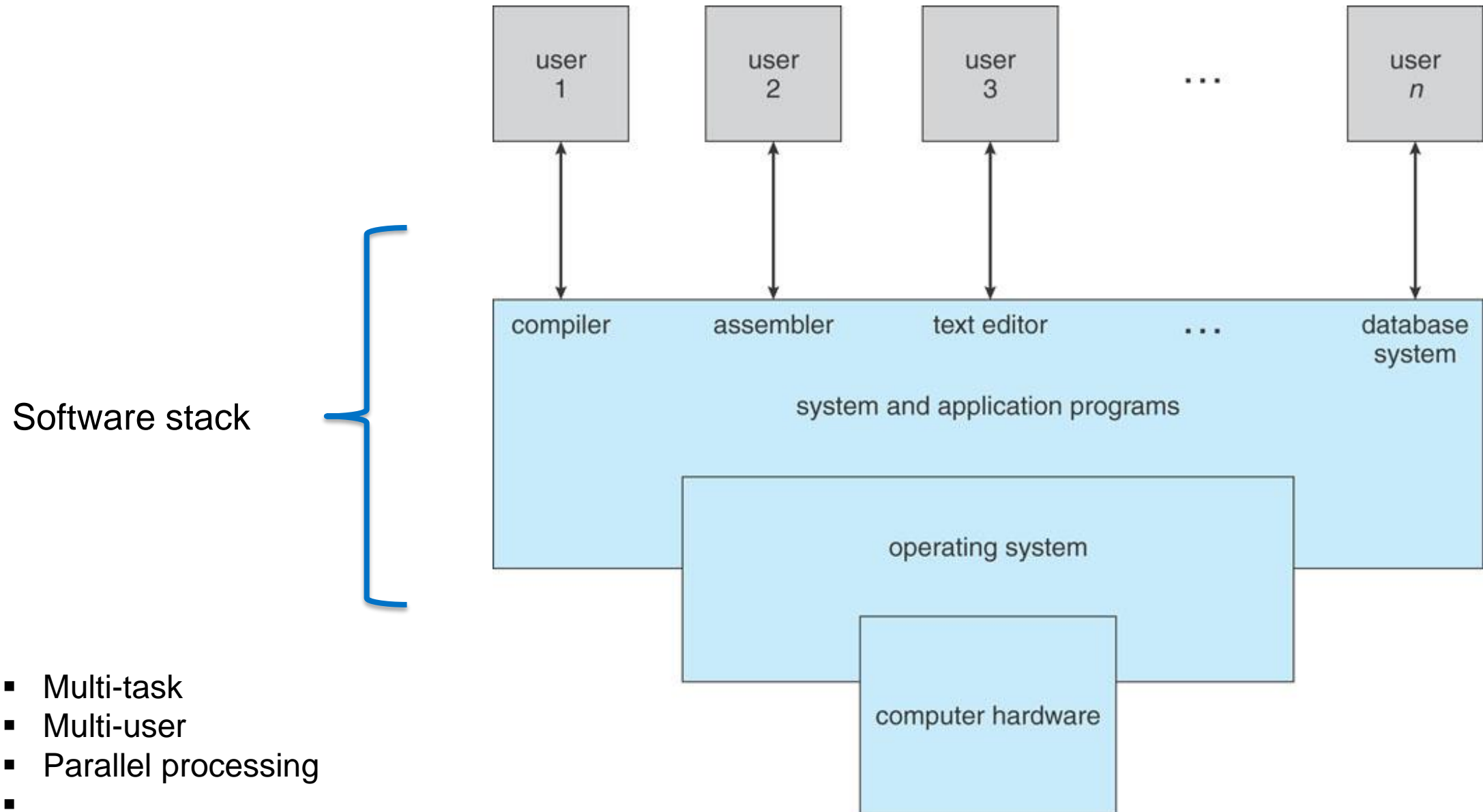
# Computer Hardware



# Computer Software



# Computer Software





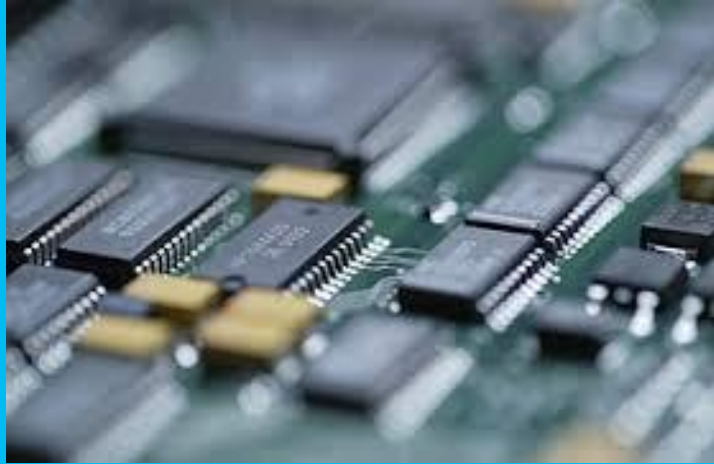
# Computer Software

Software stack



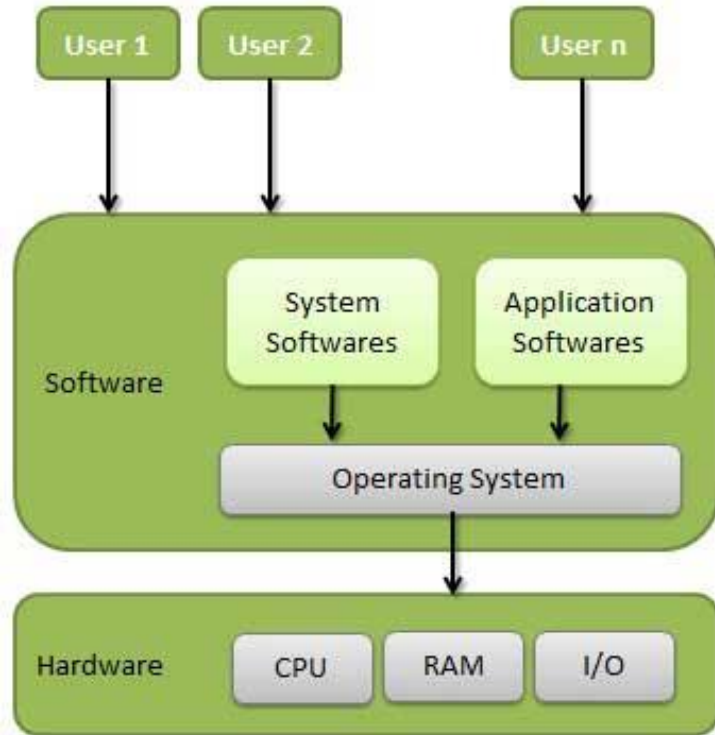
- Multi-task
- Multi-user
- Parallel processing
- .....



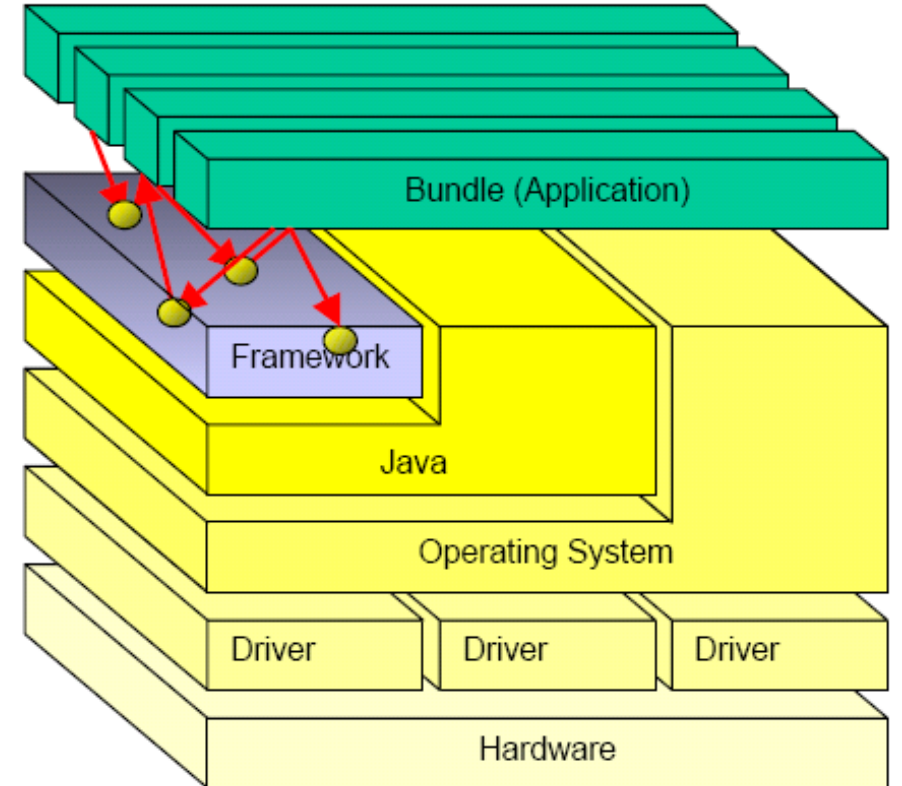


# Introduction to Memory

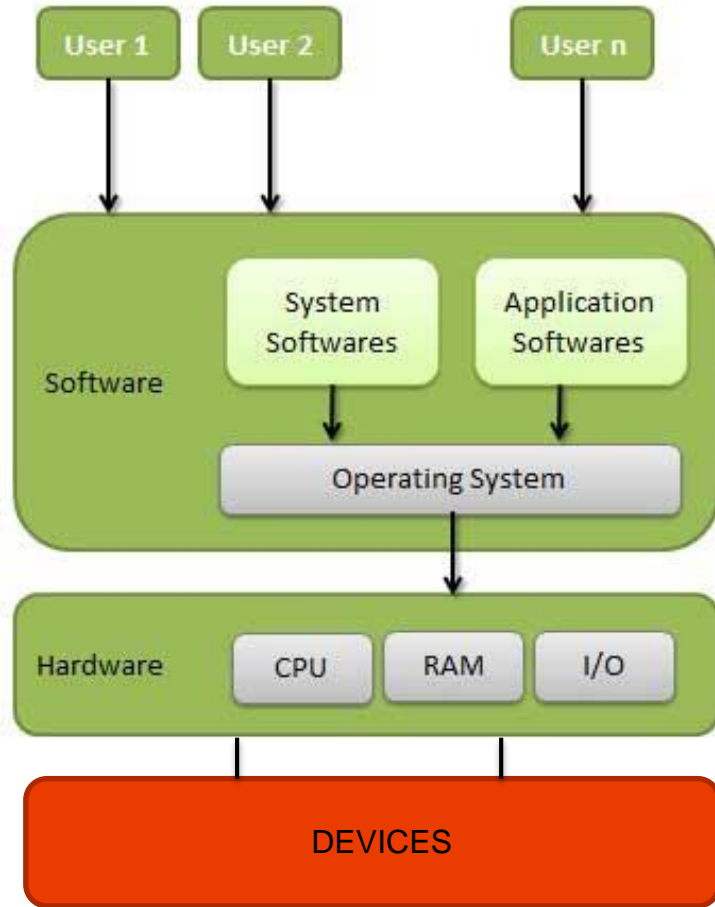
# Background: OS, Java and Memory



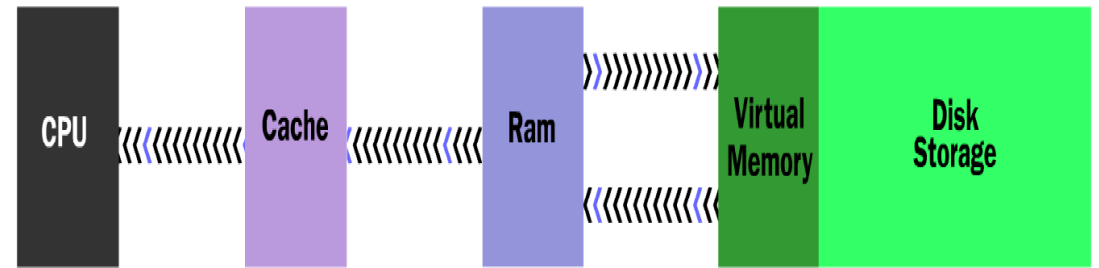
● = service interface  
exported and  
imported



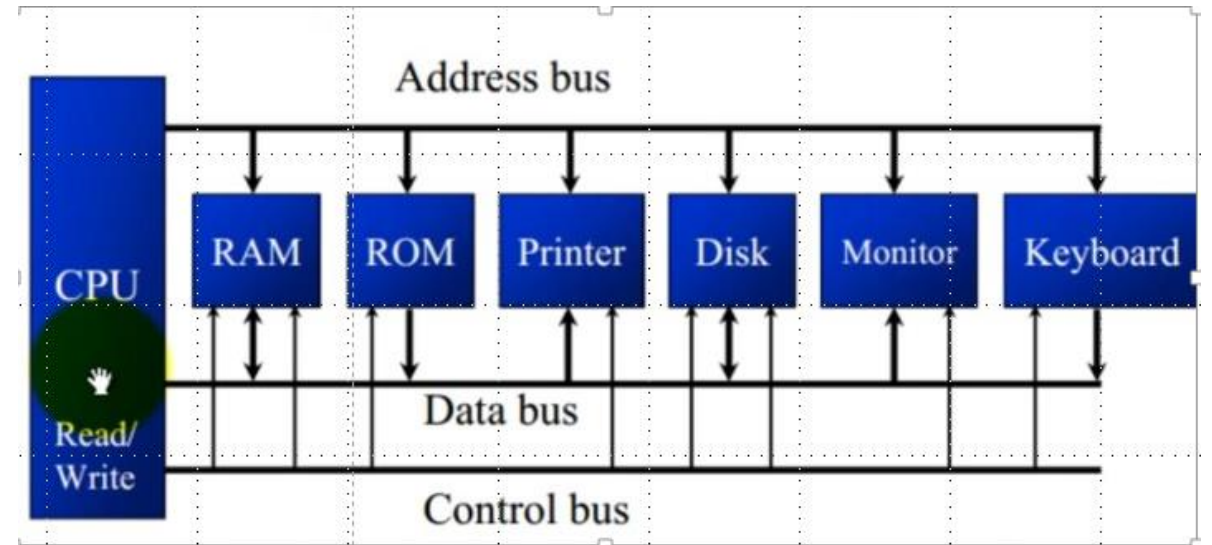
# Memory : Hardware and Software



## Memory Management

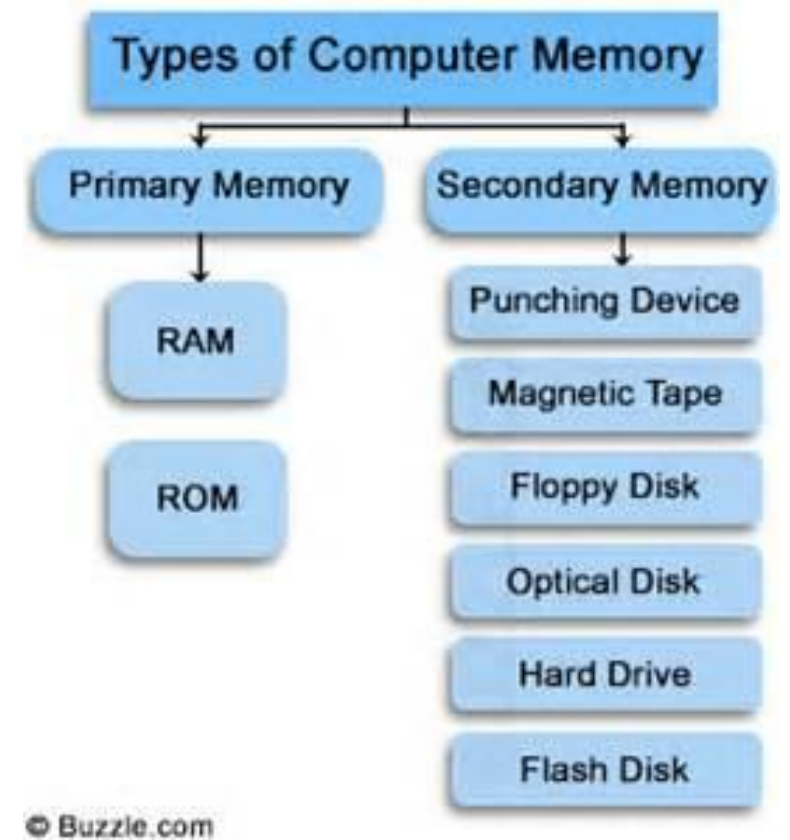


© 2000 How Stuff Works, Inc



# Variables and Memory

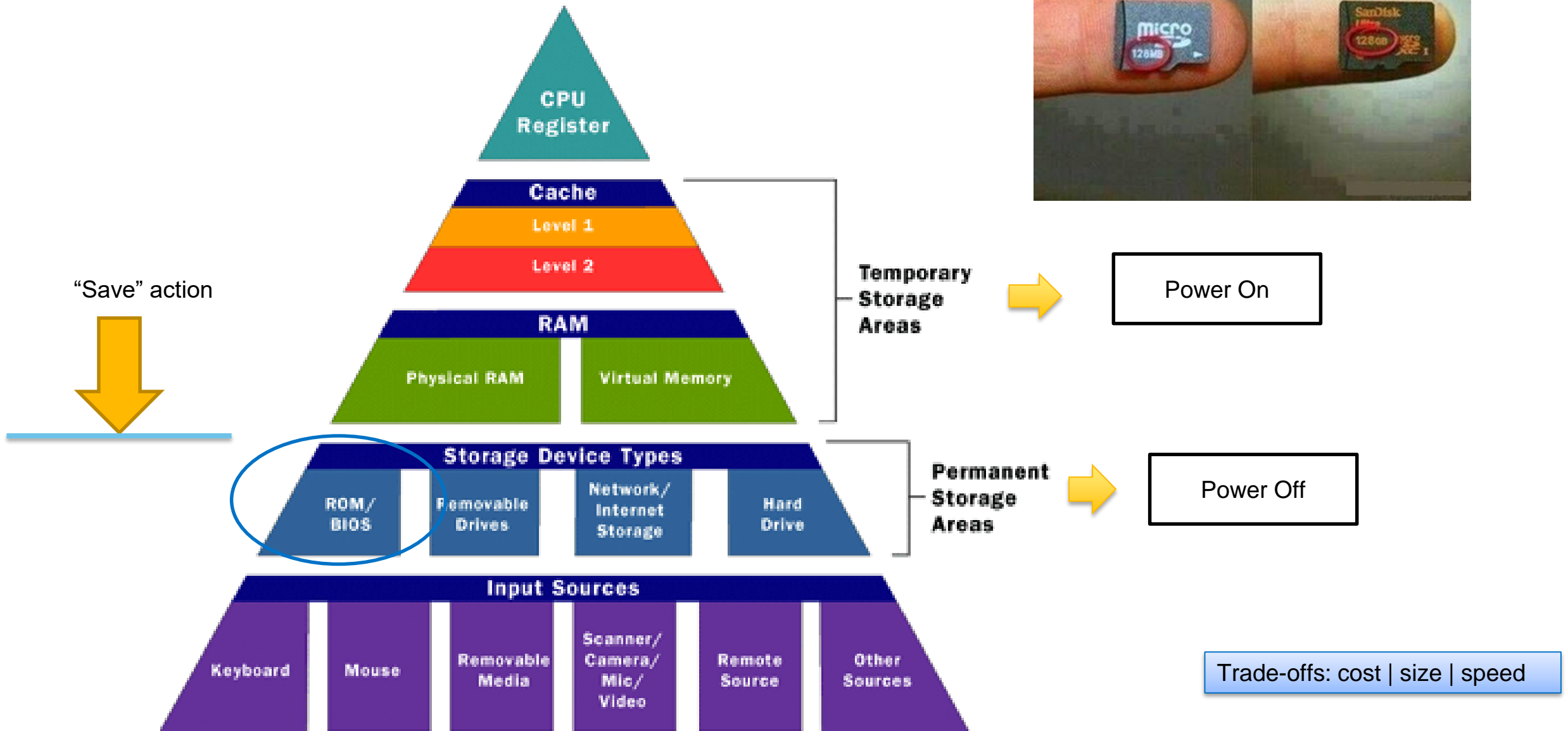
- A computer has two forms of memory
- **Primary** or *Main memory* is used by a computer when it is running a program
  - Values stored in a program's variables are kept in main memory
- **Secondary** memory is used to hold files for "permanent" storage



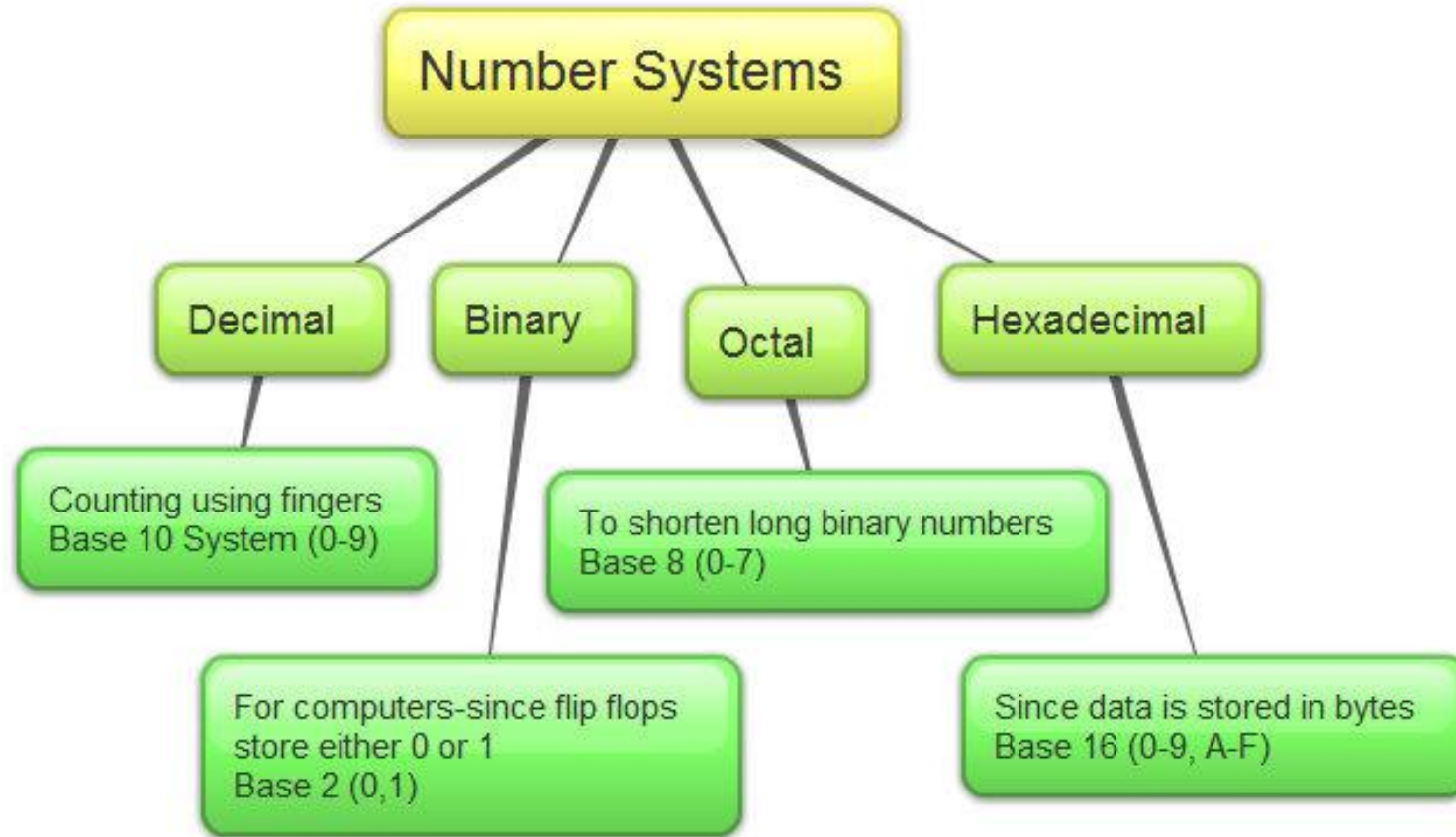
Examples of Primary and Secondary Memory



# Memory: permanent and temporary storage



# Memory



Binary	Decimal	Hexadecimal
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	10	A
1011	11	B
1100	12	C
1101	13	D
1110	14	E
1111	15	F

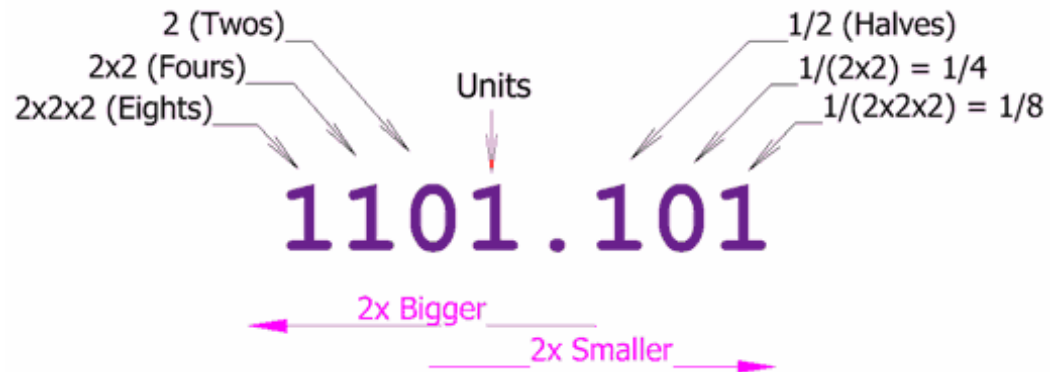
$16^4$	$16^3$	$16^2$	$16^1$	$16^0$
2	4	A	B	F
				$15 \times 1 = 15$
				$11 \times 16 = 176$
			$10 \times 256 = 2560$	
		$4 \times 4096 = 16384$		
$2 \times 65536 = 131072$				
$131,072 + 16,384 + 2,560 + 176 + 15 = 150,207$				

# Variables and Memory

- Main memory consists of a long list of numbered locations called ***bytes***
  - Each byte contains **eight *bits***: eight 0 or 1 digits
- The number that identifies a byte is called its ***address***
  - A data item can be stored in one (or more) of these bytes
  - The address of the byte is used to find the data item when needed

# Binary System

```
010110111001010
101000101000000
0100101110111010
100001000111101
```



100111

→	$2^0 \times 1 = 1$
→	$2^1 \times 1 = 2$
→	$2^2 \times 1 = 4$
→	$2^3 \times 0 = 0$
→	$2^4 \times 0 = 0$
→	$2^5 \times 1 = 32$
<hr/>	
	Decimal ← 39

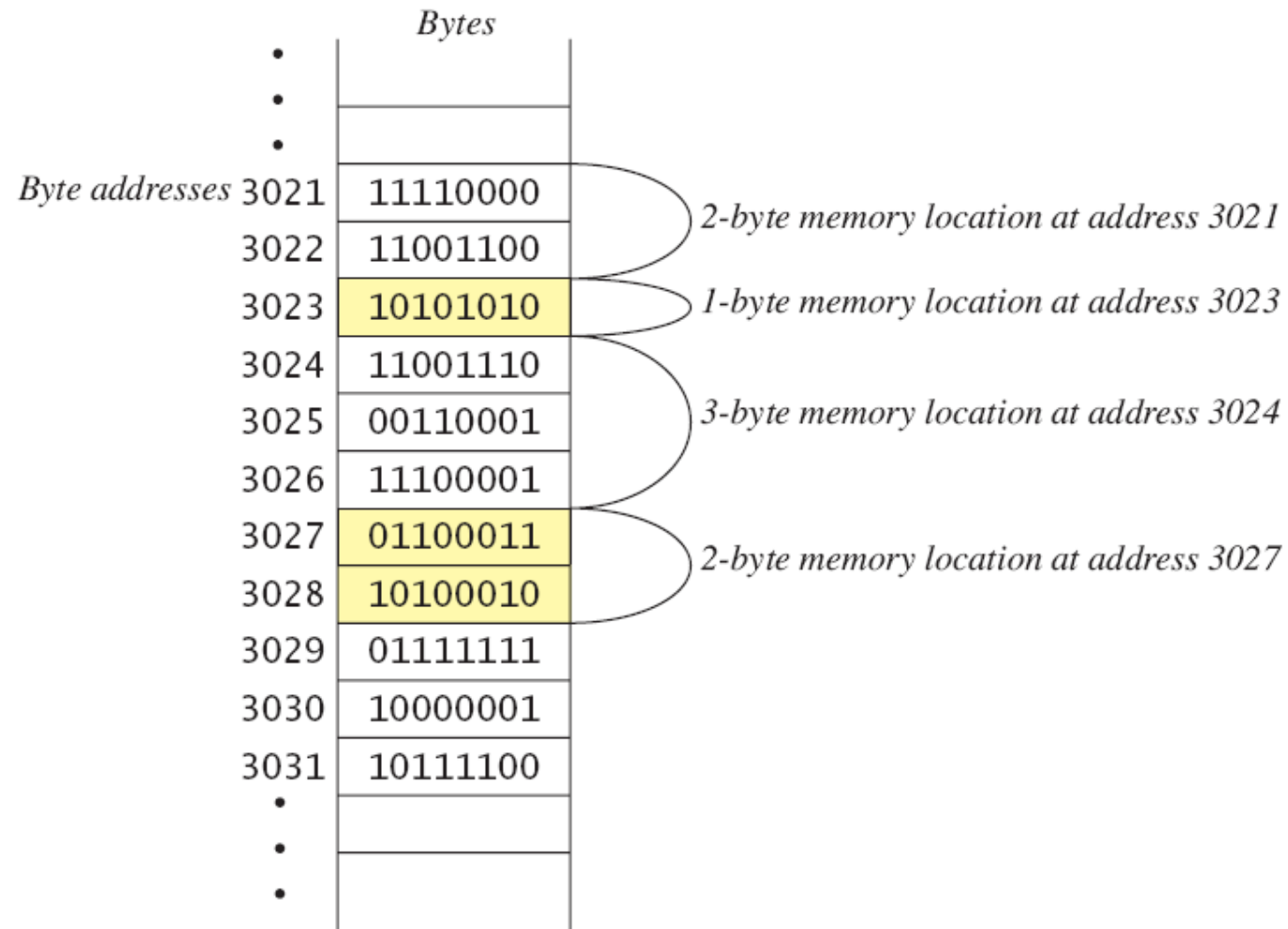
# Memory: whiteboard



Binary system



# Computer Memory

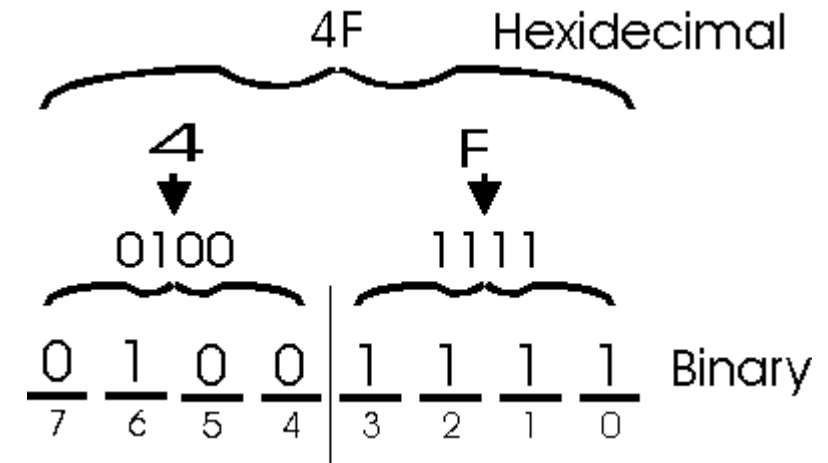


## Systems

- Binary
- Decimal
- Hexa-decimal

# Hexadecimal System

Decimal (Base 10)	Binary (Base 2)	Hexadecimal (Base 16)
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F



# Memory naming conventions

SI System of Prefixes:	
S	
M	Giga (G) $10^9 = 1000000000$
A	Mega (M) $10^6 = 1000000$
L	Kilo (k) $10^3 = 1000$
<hr/>	
L	Base $10^0$ (g, m, L, s, mol)
L	Deci (d) $10^{-1} = 0.1$
L	Centi (c) $10^{-2} = 0.01$
E	Milli (m) $10^{-3} = 0.001$
E	Micro ( $\mu$ ) $10^{-6} = 0.000001$
R	Nano (n) $10^{-9} = 0.000000001$
R	Pico (p) $10^{-12} = 0.000000000001$

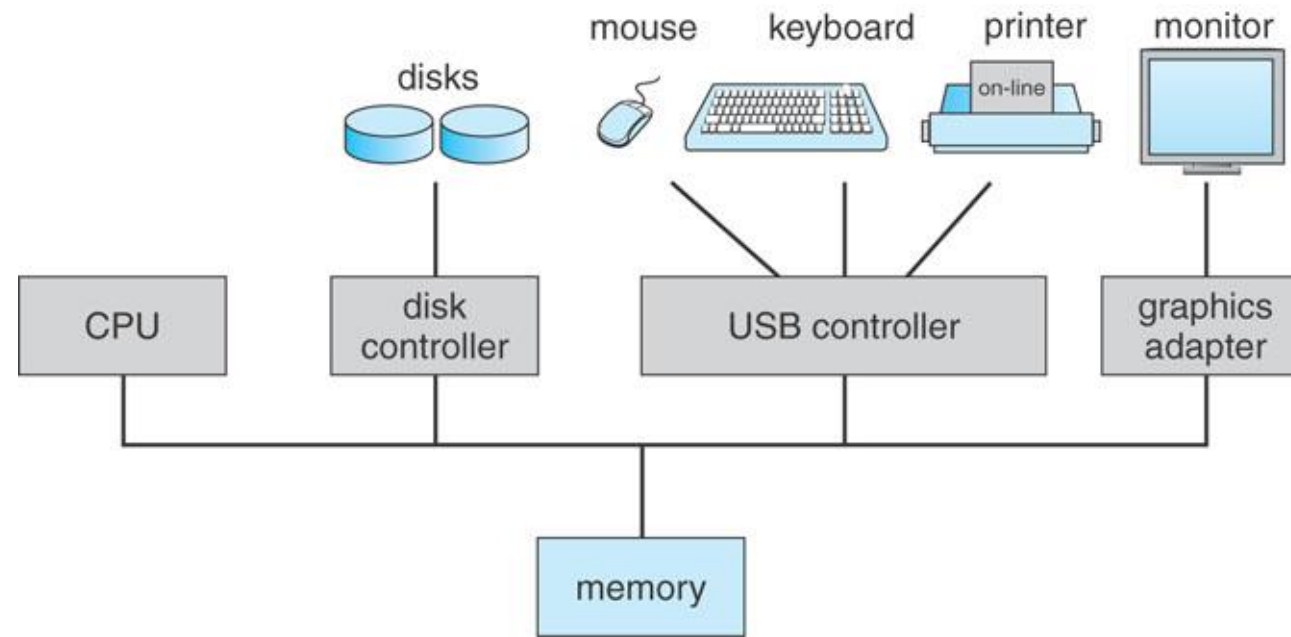
Prefixes for bit and byte multiples				
Decimal		Binary		
Value	SI	Value	IEC	JEDEC
1000 k	kilo	1024 Ki	kibi	K kilo
$1000^2$ M	mega	$1024^2$ Mi	mebi	M mega
$1000^3$ G	giga	$1024^3$ Gi	gibi	G giga
$1000^4$ T	tera	$1024^4$ Ti	tebi	
$1000^5$ P	peta	$1024^5$ Pi	pebi	
$1000^6$ E	exa	$1024^6$ Ei	exbi	
$1000^7$ Z	zetta	$1024^7$ Zi	zebi	
$1000^8$ Y	yotta	$1024^8$ Yi	yobi	



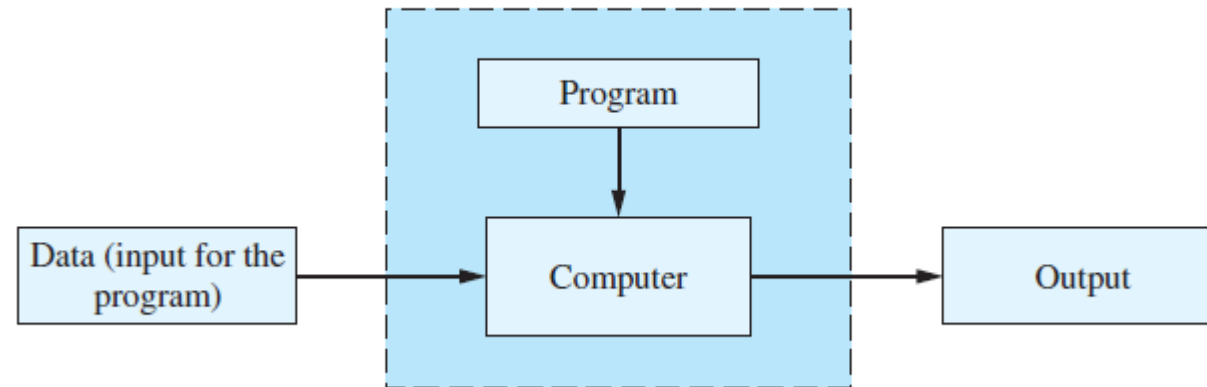
Examples?

Microprocessor; Nanotechnology,....

# Putting all together



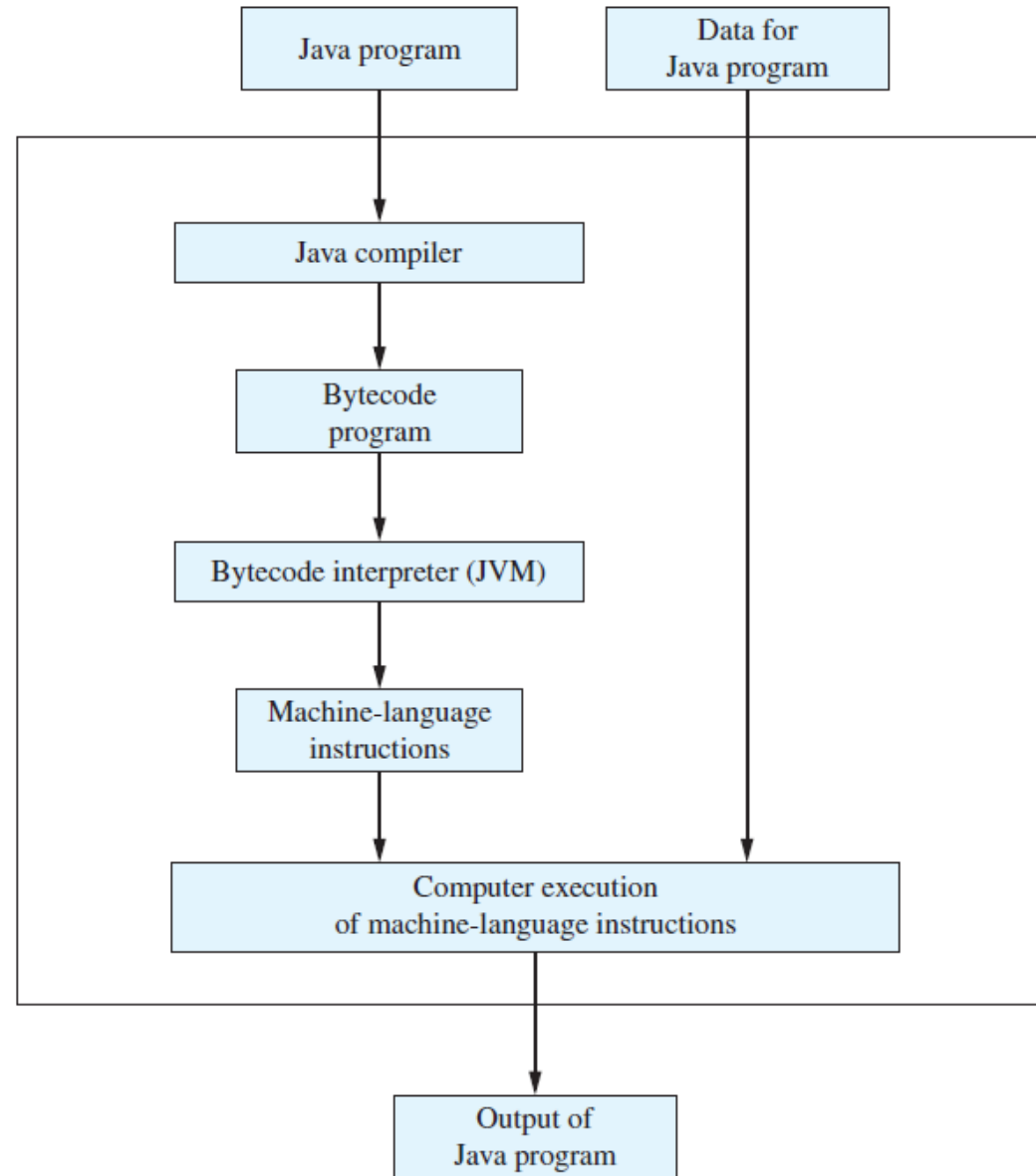
## Running a Program



# Putting all together

## Compiling and Running a Java Program

---





## Display 1.1 A Sample Java Program

---

```
1 public class FirstProgram
2 {
3     public static void main(String[] args)
4     {
5         System.out.println("Hello reader.");
6         System.out.println("Welcome to Java.");
7
8         System.out.println("Let's demonstrate a simple calculation.");
9         int answer;
10        answer = 2 + 2;
11        System.out.println("2 plus 2 is " + answer);
12    }
```

Annotations:

- ← Name of class (program) points to `FirstProgram`
- ← The main method points to `public static void main(String[] args)`

### SAMPLE DIALOGUE 1

```
Hello reader.
Welcome to Java.
Let's demonstrate a simple calculation.
2 plus 2 is 4
```

## LISTING 1.1 A Sample Java Program

```
import java.util.Scanner;

public class FirstProgram
{
    public static void main(String[] args)
    {
        System.out.println("Hello out there.");
        System.out.println("I will add two numbers for you.");
        System.out.println("Enter two whole numbers on a line:");

        int n1, n2;

        Scanner keyboard = new Scanner(System.in);

        n1 = keyboard.nextInt();
        n2 = keyboard.nextInt();

        System.out.println("The sum of those two numbers is");
        System.out.println(n1 + n2);
    }
}
```

*Gets the Scanner class from the package (library) java.util*

*Name of the class—your choice*

*Sends output to screen*

*Says that n1 and n2 are variables that hold Integers (whole numbers)*

*Reads the program for keyboard input*

*Reads one whole number from the keyboard*

Using external input

### Sample Screen Output

```
Hello out there.
I will add two numbers for you.
Enter two whole numbers on a line:
12 30
The sum of those two numbers is
42
```

Programming in Java

**To be continued**