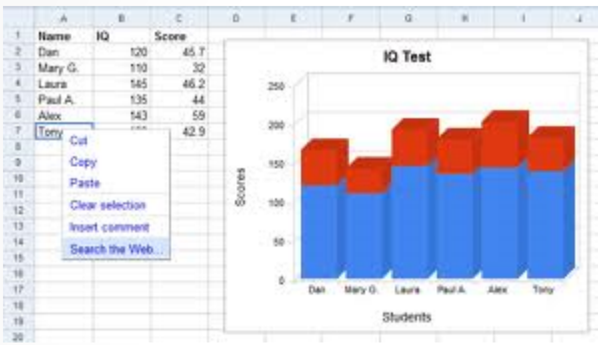
Welcome to CSC220 (CSI) Computational Problem Solving

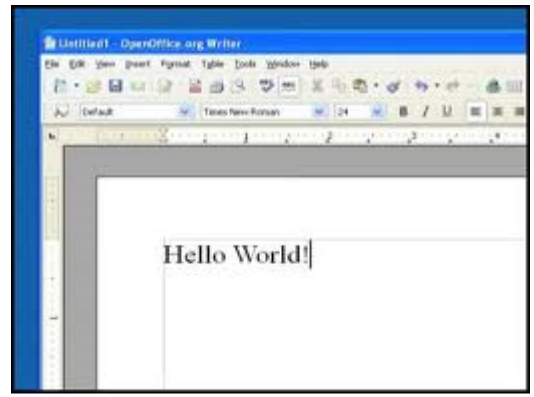
The College of New Jersey

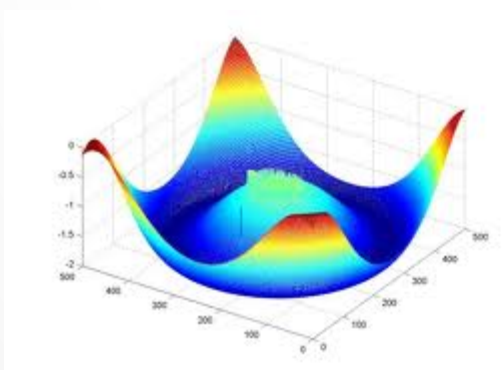
Please turn off your cell phone!

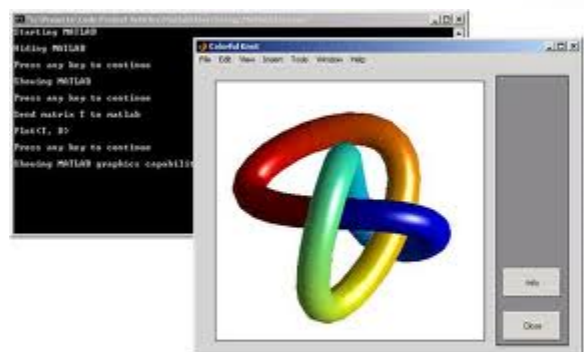
What is Computing?

Computing: Your Parent's View









Computing: internet, e-mail, network…

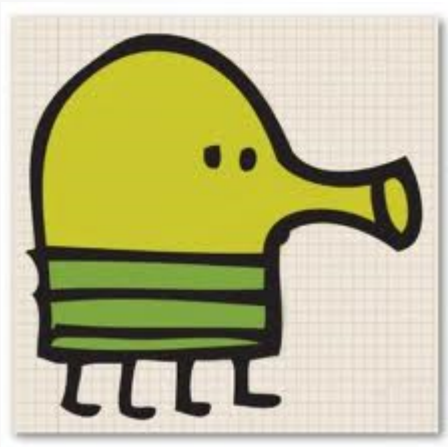

Computing: Digital Photography
http://www.alanzeyes.com/2009/02/hdr-photography.html

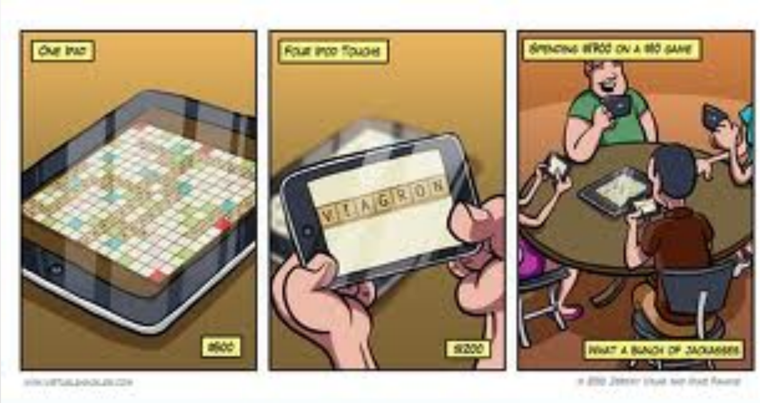
Computing: Entertainment…

YouTube

NETFLIX®

Computing: Entertainment…

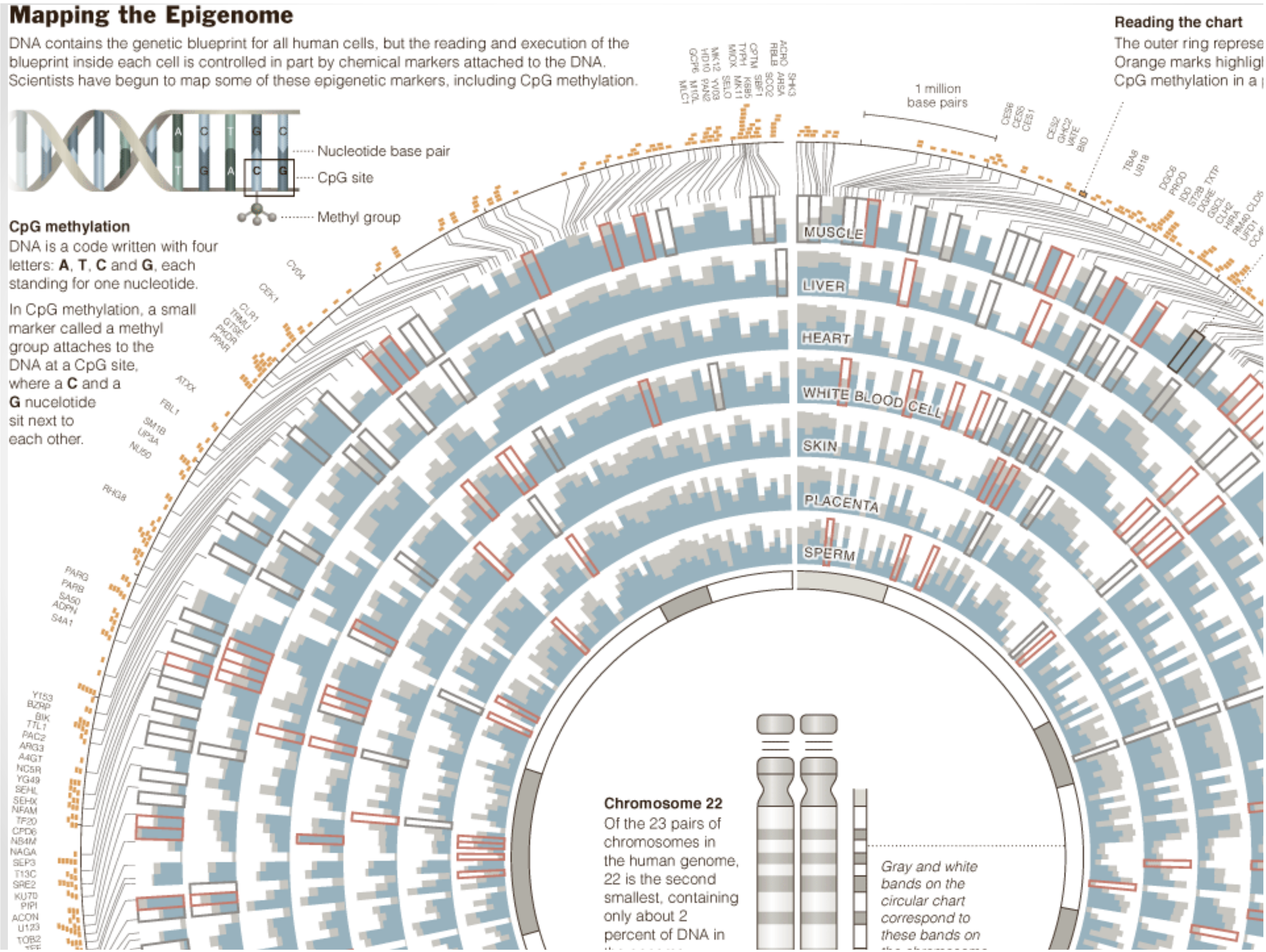




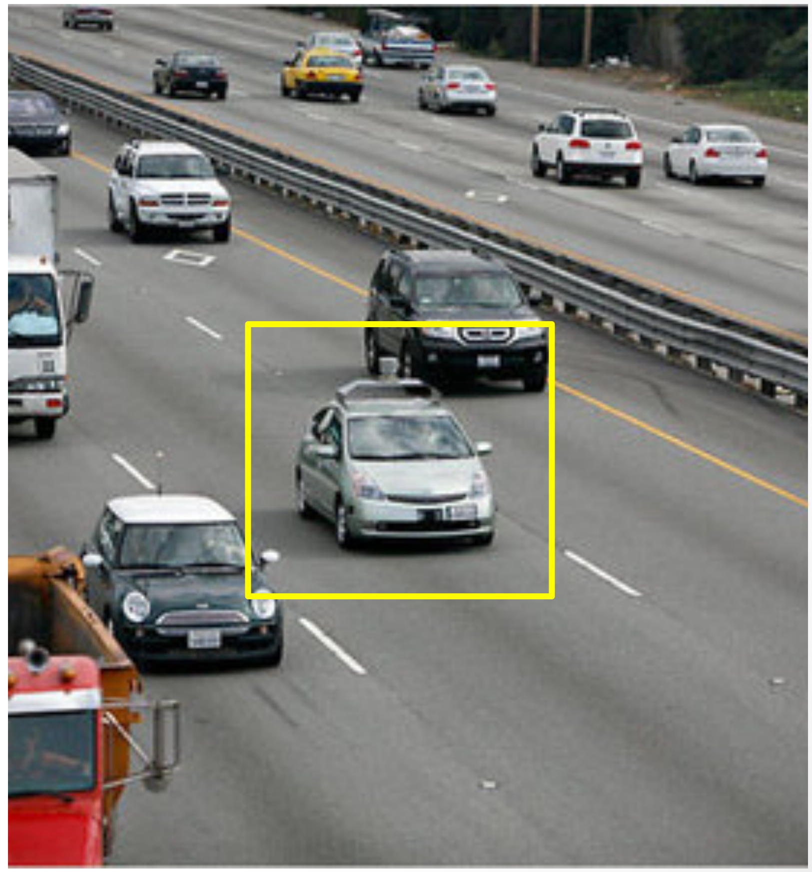
"Computer science is no more about computers than astronomy is about telescopes"

- Edsger Dijkstra

Cutting Edge Computer Science


Mapping the Epigenome
DNA contains the genetic blueprint for all human cells, but the reading and execution of the blueprint inside each cell is controlled in part by chemical markers attached to the DNA. Scientists have begun to map some of these epigenetic markers, including CpG methylation.
Nucleotide base pair
CpG site
Methyl group
A C T G C
T G A C G
CpG methylation
DNA is a code written with four letters: A, T, C and G, each standing for one nucleotide.
In CpG methylation, a small marker called a methyl group attaches to the DNA at a CpG site, where a C and a G nucelotide sit next to each other.
Reading the chart
The outer ring represe
Orange marks highligl
CpG methylation in a
1 million base pairs
MUSCLE
LIVER
HEART
WHITE BLOOD CELL
SKIN
PLACENTA
SPERM
Chromosome 22
Of the 23 pairs of chromosomes in the human genome, 22 is the second smallest, containing only about 2 percent of DNA in
Gray and white bands on the circular chart correspond to these bands on

Google's Autonomous Car



- Nevada made it legal for autonomous cars to drive on roads in June 2011
- California introduced a similar bill in Aug 2012

2011 Jeopardy!

- In February 2011, IBM Watson bested Brad Rutter (biggest all-time money winner) and Ken Jennings (longest winning streak)
- IBM is currently applying Watson's technology to medical diagnosis and legal research

ART

Protobytes
By Ira Greenberg

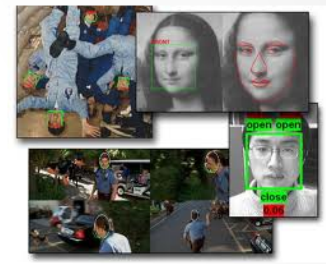
Areas in Computer Science

Artificial Intelligence

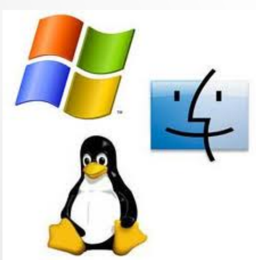
Robotics

Human-Computer Interaction

Computer Graphics



Computer Vision



Operating Systems

Computer Networking

Databases

Computer Security

Ubiquitous Computing

What is Computer Science?

Computer science is the study of solving problems using computation

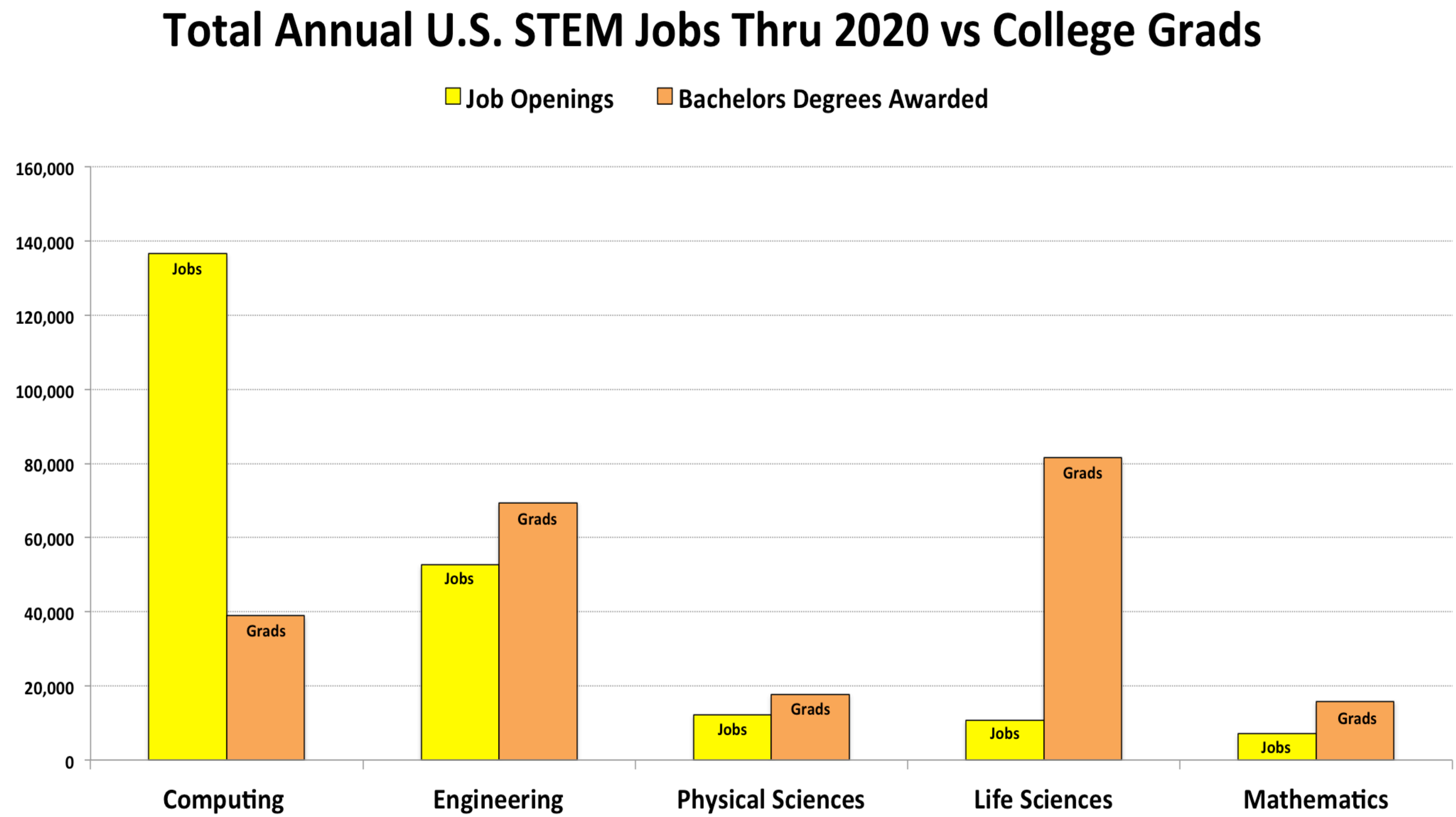
- Computers are part of it, but the emphasis is on the problem solving aspect

Computer scientists work across disciplines:

Mathematics	Geoscience	Medicine/Surgery
Biology (bioinformatics)	Archeology	Engineering
Chemistry	Psychology	Linguistics
Physics	Sociology	Art
Geology	Cognitive Science	…

Computing is important!

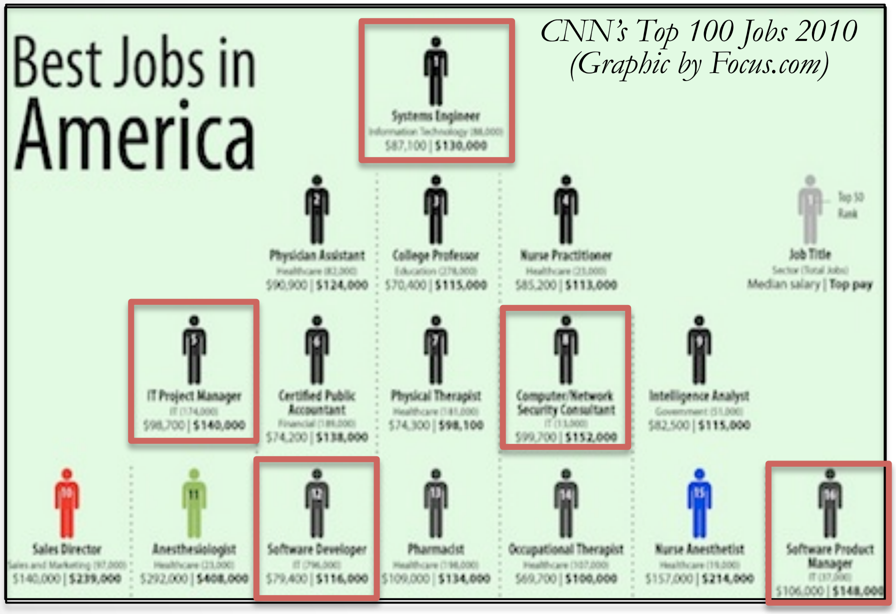
Huge Growth in Computing-Related Jobs



Data Sources: *US-BLS Employment Projections, 2010-2020* (http://www.bls.gov/emp/ep_table_102.htm), *National Science Foundation Division of Science Resource Statistics* (http://www.nsf.gov/statistitcs/nsf11316/pdf/tab28.pdf, tab33,pdf, tab34.pdf, tab35.pdf, tab46.pdf)

Computing is Consistently Ranked Among the Best Occupations

CS-Related Jobs Highlighted in Red



CNN's Top 100 Jobs 2010
(Graphic by Focus.com)

The 25 Best Jobs of 2012

# 1	Registered Nurse	#6	Web Developer
# 2	Software Developer	#7	Computer Systems Analyst
# 3	Pharmacist	#8	Physical Therapist
# 4	Medical Assistant	#9	Computer Programmer
# 5	Database Administrator	#10	Occupational Therapist

CS Careers Rank Highly In:

- Job satisfaction
- Salary
- Work/life balance
- Growth potential
- Employment rate
- Work environment

Computer science tops list of best major for jobs

BY RACHEL GOTTFRIED

Computer science graduates now get more offers of employment than any other major. This is the first time since 2008 that computer science has topped the list: previously, accounting majors had the highest offer rate.

In 2011, 56.2% of computer science majors received job offers, compared to only 53.8% of accounting majors. The offer rate for computer science majors increased 13.8% this year from the previous year.

Computer science and accounting majors are in high demand because both are needed in a wide range of industries.

"There are many different companies that need to hire computer scientists," said Mimi Collins, director of communications at the National Association of Colleges and Employers.

"They aren't tied to one particular industry—majors like nursing do not enjoy that benefit."

Although this is good news for computer science grads, it might not be for the computer industry. According to Collins, "One computer science graduate may have 10 offers as opposed to one accounting graduate that's getting five offers." So, computer science majors may be getting more offers, but this is only because there is a shortage of people who graduate with such a degree.

According to Collins, companies like to hire recent graduates because they have the latest skills.

"Things change very quickly, especially in computer science," said Collins. "Many organizations have a formal track where they want to bring in new college graduates and train them the way they want them to be trained."

Annabelle Evans graduated as a computer science major from the University of Southern California in 2008. "When I picked my major, I knew there wouldn't be a lack of jobs as a computer scientist, and that was part of the appeal," she said. Evans now works at Google.

…many different companies … need to hire computer scientists. They aren't tied to one particular industry.

Computational Problem Solving

Algorithms

An **algorithm** is an effective method for solving a problem expressed as a finite sequence of instructions. For example,

Put on shoes

left sock

right sock

left shoe

right shoe

Programming = Writing Apps

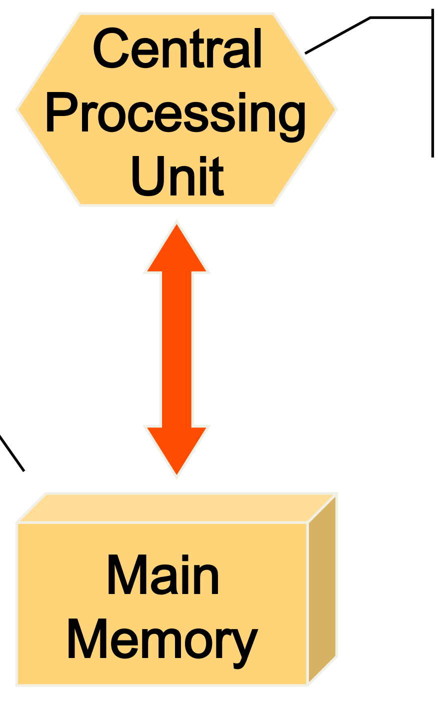
Programming is the process of designing, writing, testing, debugging / troubleshooting, and maintaining the source code of computer programs. This source code is written in a programming language.

Computer Processing

Computer Processing

- Hardware
  - the physical, tangible parts of a computer
  - keyboard, monitor, disks, wires, chips, etc.
- 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

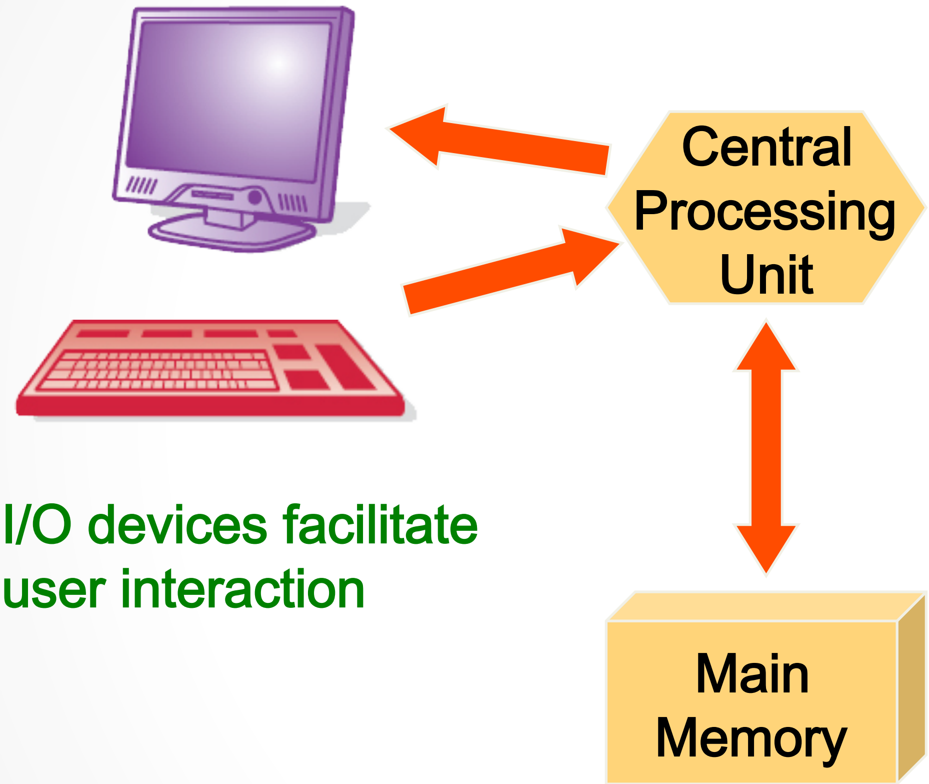


Chip that executes program commands

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

Synonymous with RAM

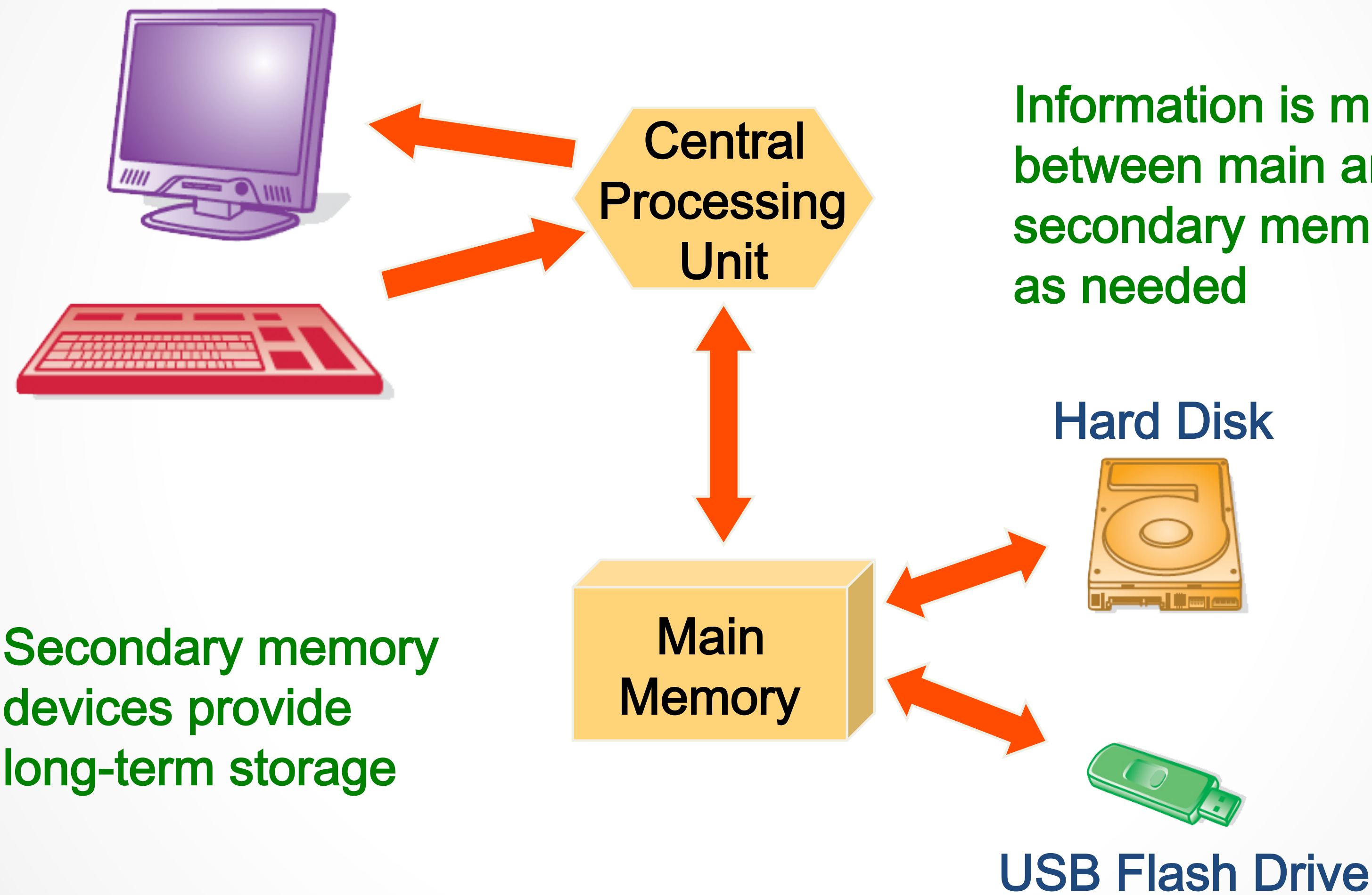
Input / Output Devices



Monitor screen
Keyboard
Mouse
Touch screen

I/O devices facilitate user interaction

Secondary Memory Devices



Information is moved between main and secondary memory as needed

Secondary memory devices provide long-term storage

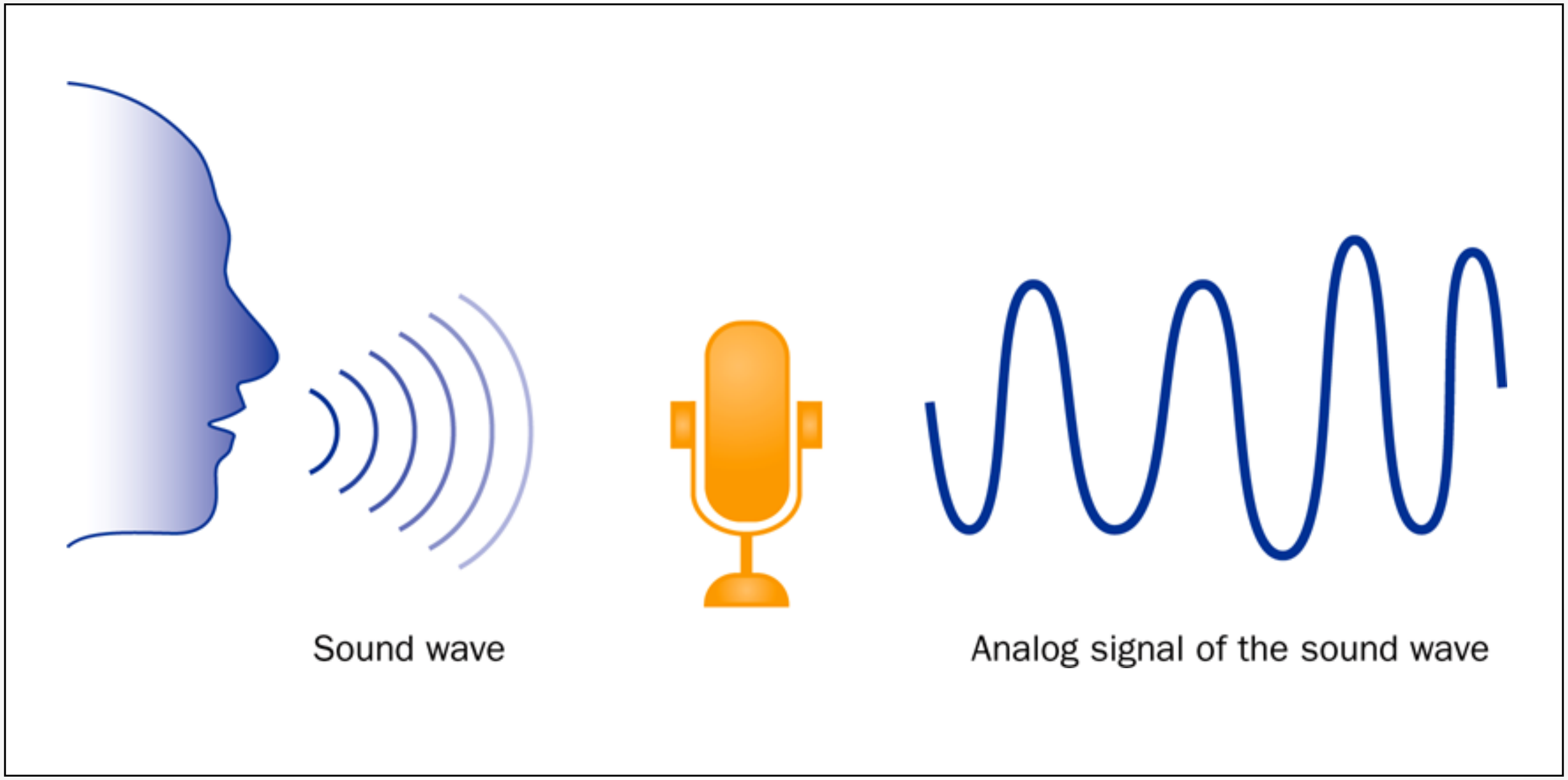
Software Categories

- 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)

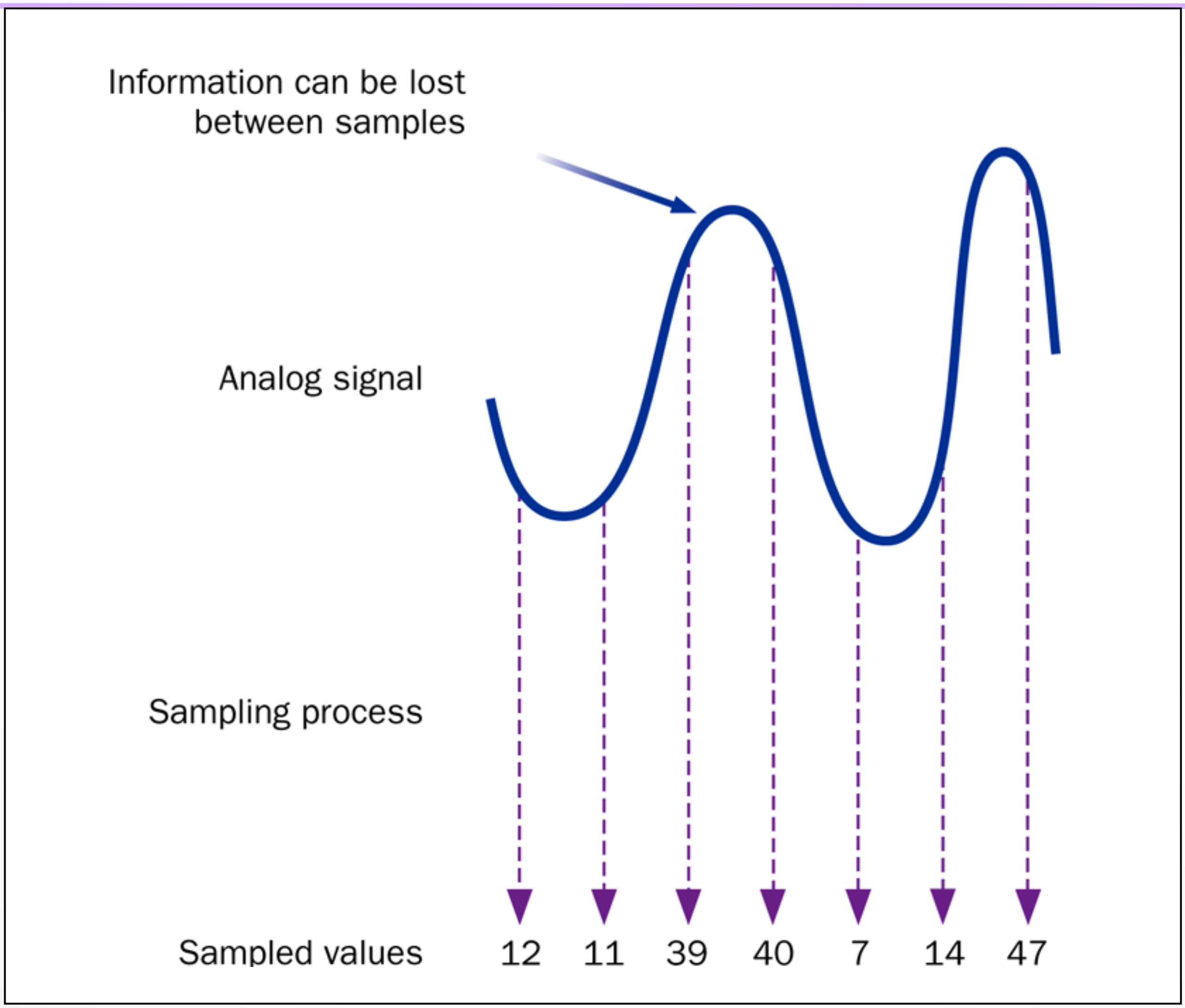
Analog vs. Digital

- 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

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

H i ,   H e a t h e r .

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*

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

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?

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

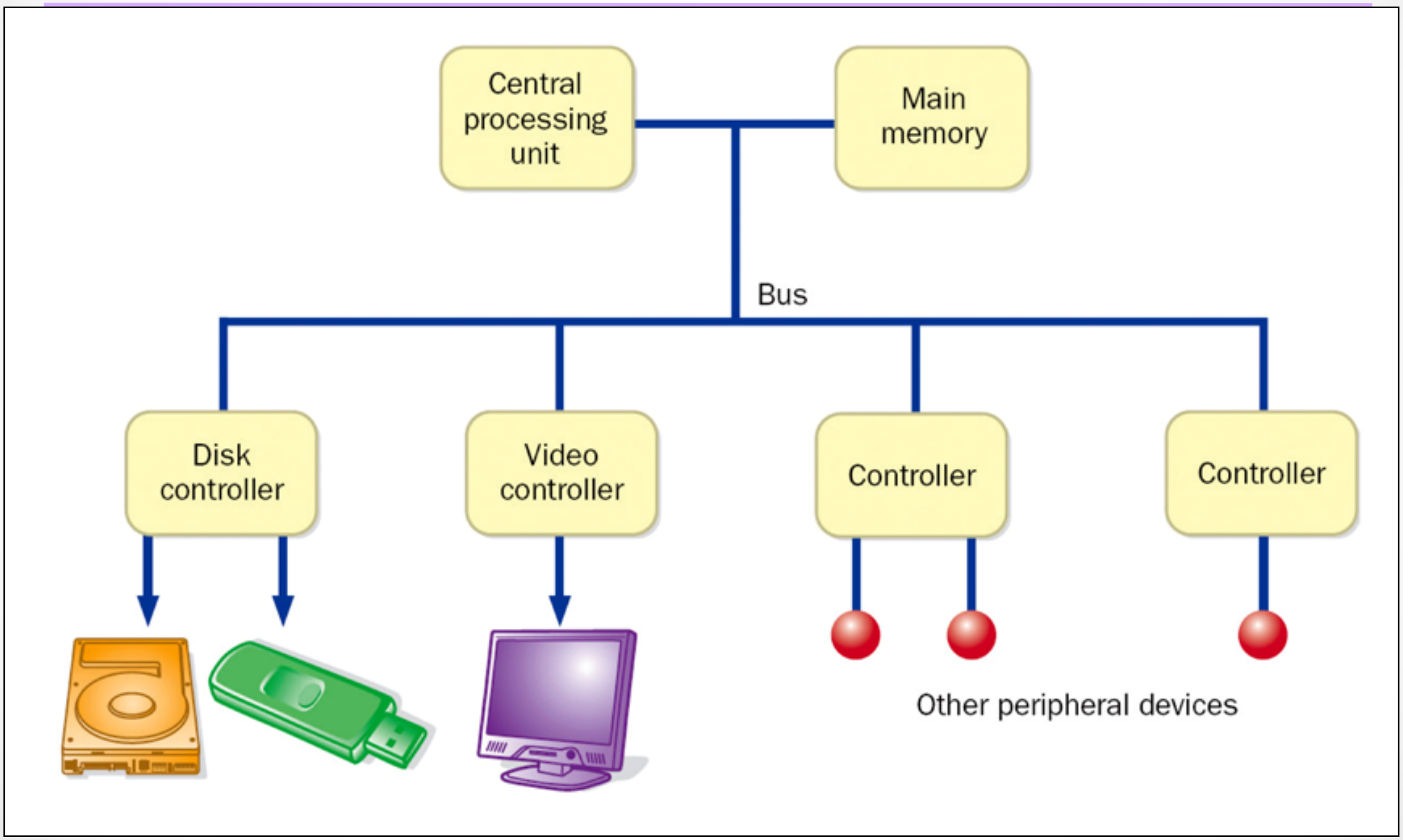
etc.

Hardware Components

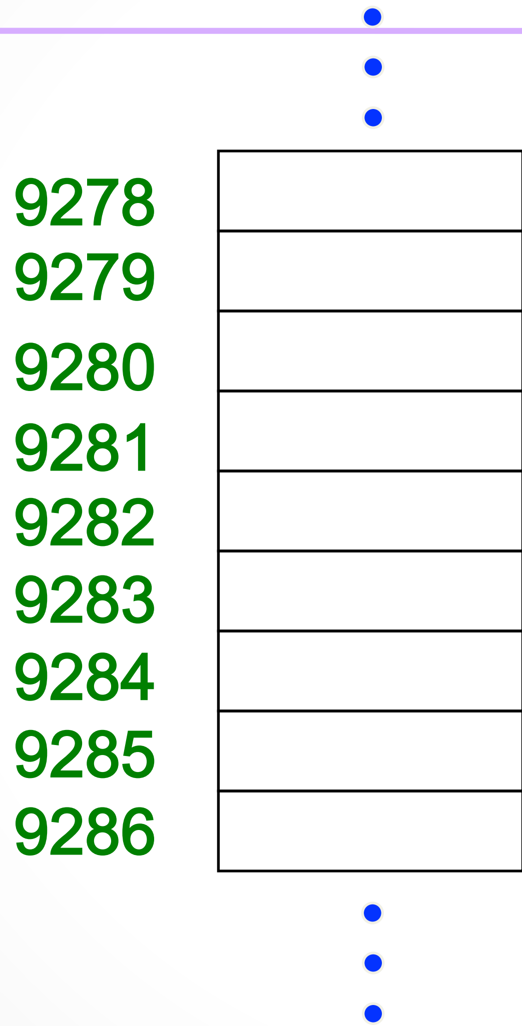
A Computer Specification

- Consider the following specification for a personal computer:
  - o 3.07 GHz Intel Core i7 processor
  - o 4 GB RAM
  - o 750 GB Hard Disk
  - o 16x Blu-ray / HD DVD-ROM & 16x DVD+R DVD Burner
  - o 17" Flat Screen Video Display with 1280 x 1024 resolution
  - o Network Card

Computer Architecture



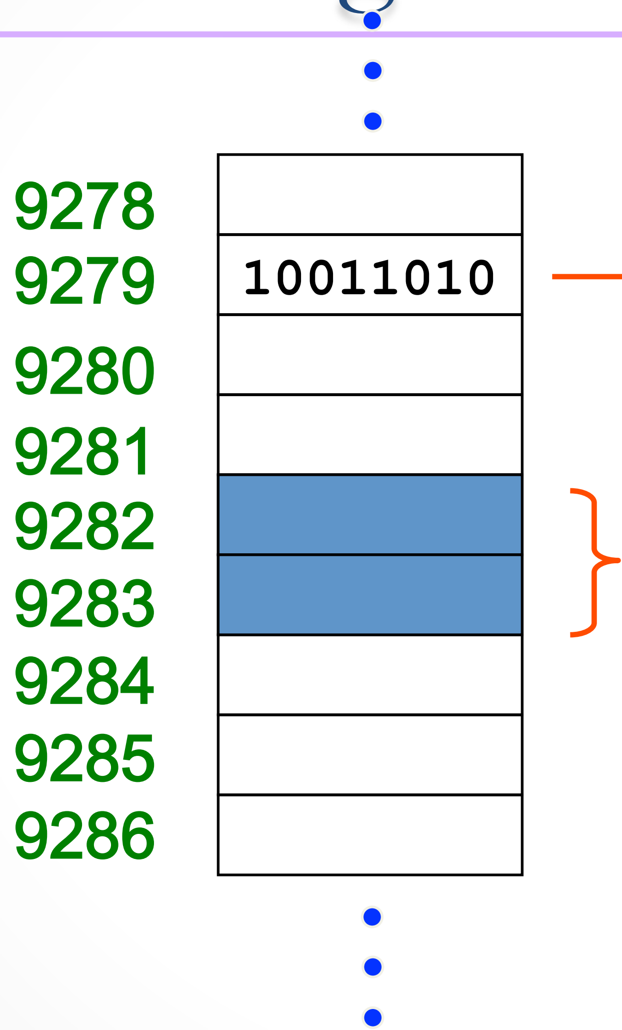
Memory



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

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

Storing Information



Each memory cell stores a set number of bits (usually 8 bits, or one *byte*)

Large values are stored in consecutive memory locations

Storage Capacity

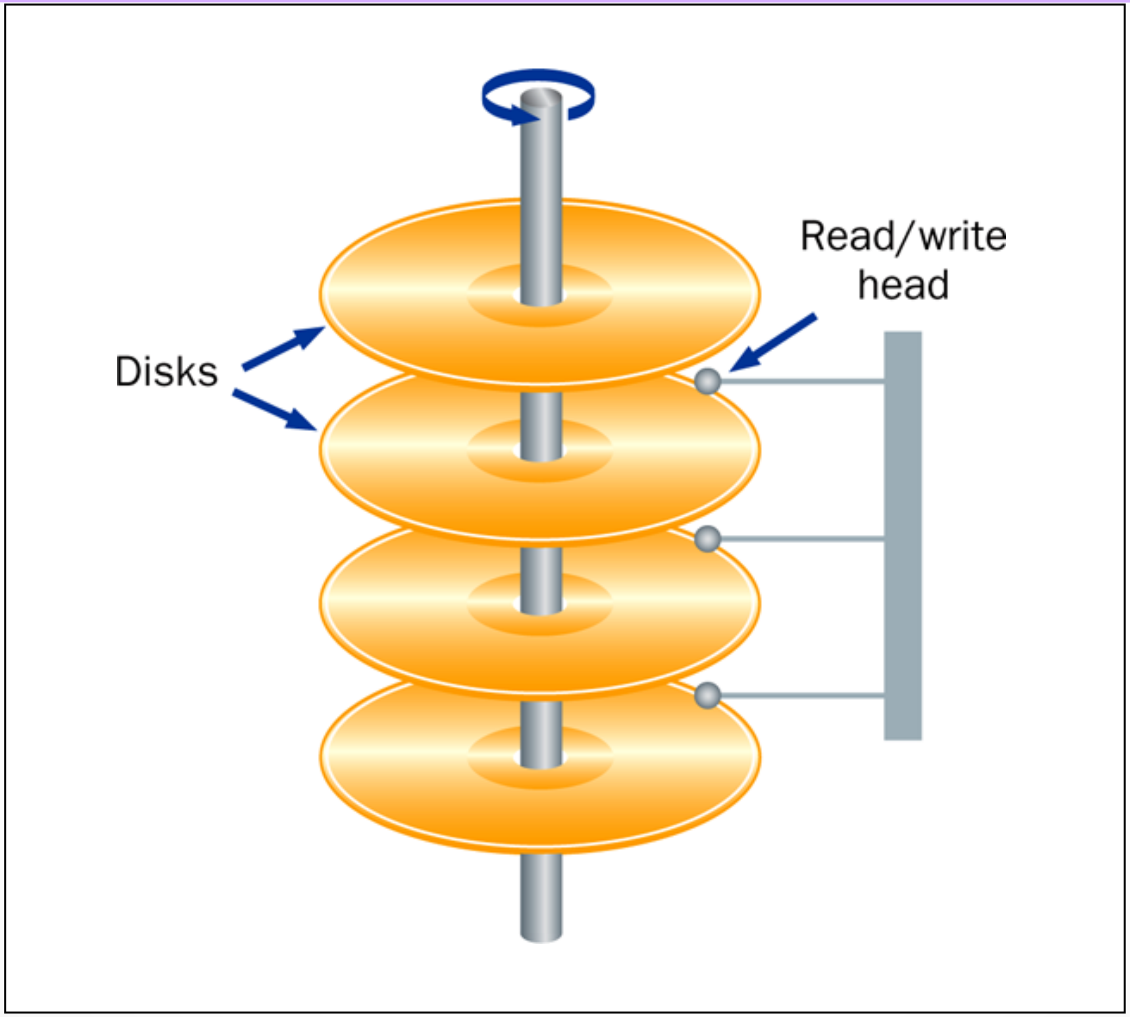
- Every memory device has a *storage capacity*, indicating the number of bytes it can hold
- Capacities are expressed in various units:

Unit	Symbol	Number of Bytes
kilobyte	KB	2^{10} = 1024
megabyte	MB	2^{20} (over one million)
gigabyte	GB	2^{30} (over one billion)
terabyte	TB	2^{40} (over one trillion)
petabyte	PB	2^{50} (a whole bunch)

Memory

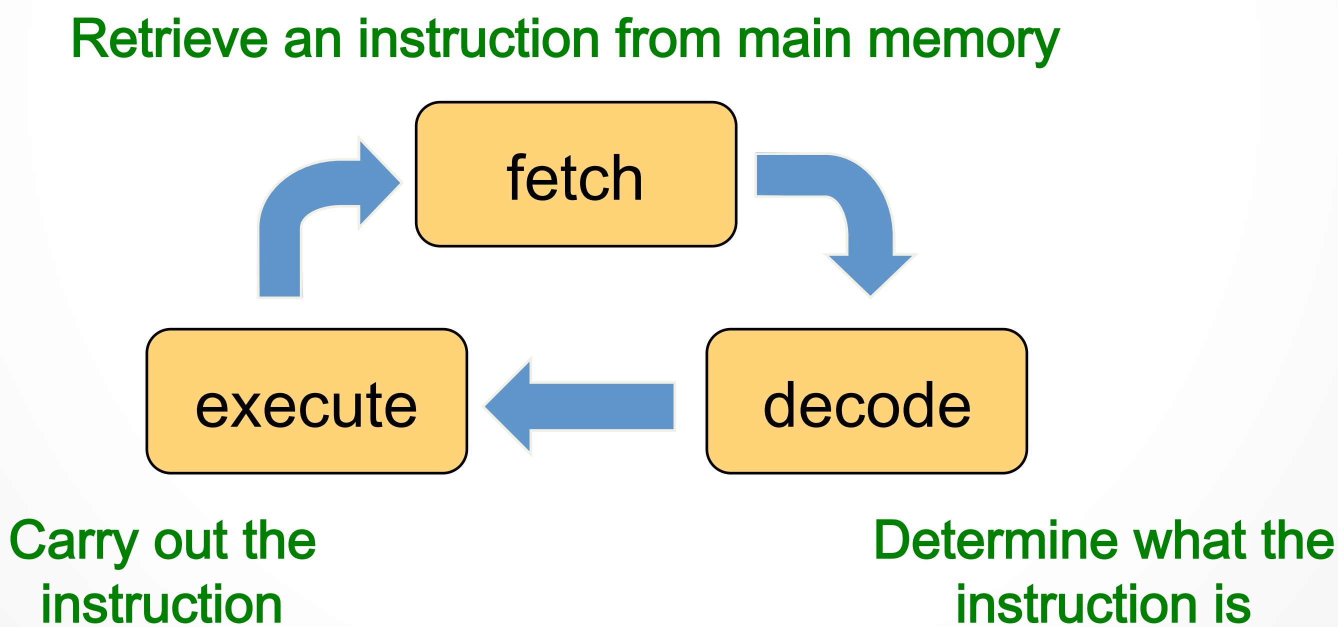
- Main memory is *volatile* - stored information is lost if the electric power is removed
- Secondary memory devices are *nonvolatile*
- Main memory and disks are *direct access* devices - information can be reached directly
- The terms *direct access* and *random access* often are used interchangeably
- 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

Hard Disk Drive

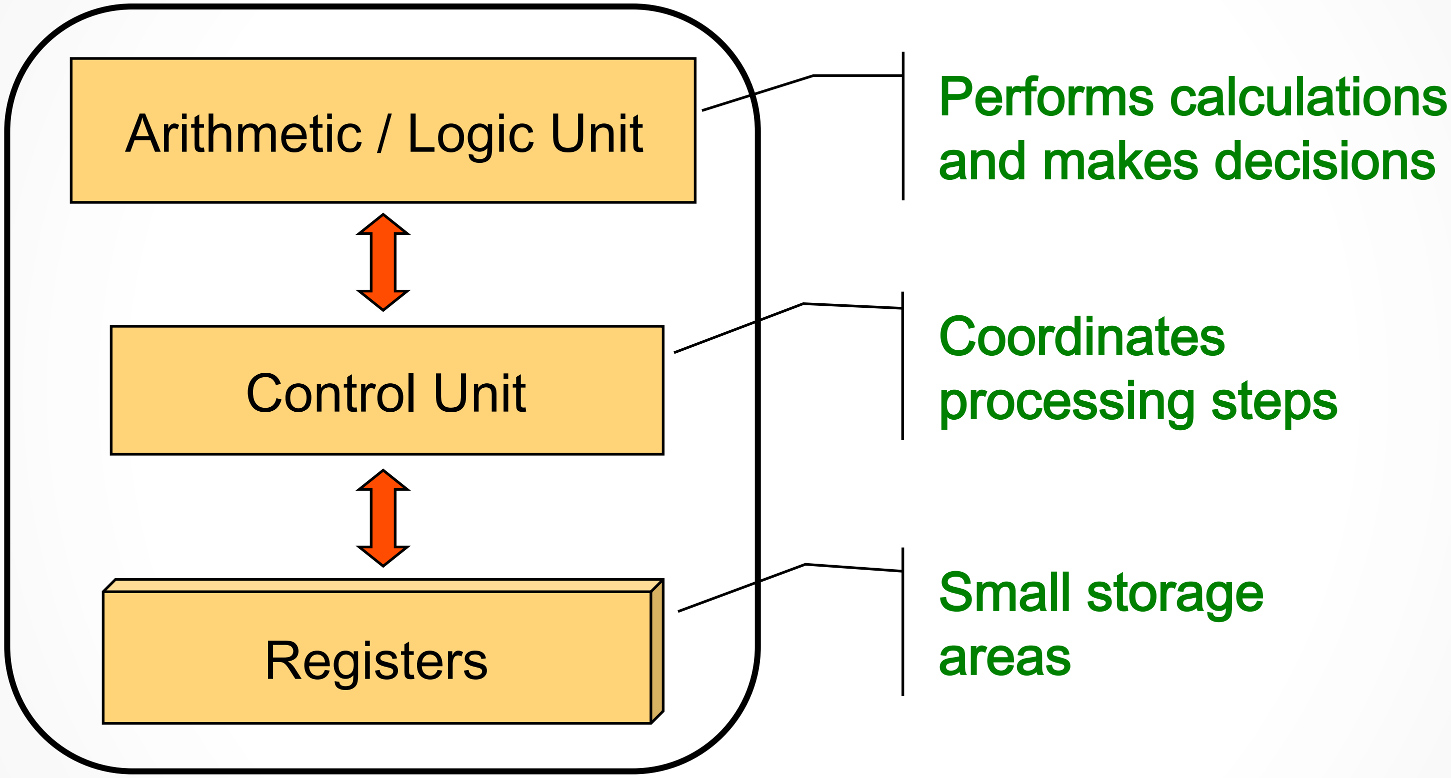


The Central Processing Unit

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



The Central Processing Unit



Monitor

- The size of a monitor (17") is measured diagonally, like a television screen
- 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

The Java Programming Language

Java

- The Java programming language was created by Sun Microsystems, Inc.
- It was introduced in 1995 and it's popularity has grown quickly since
- 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 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

```java
//********************************************************************
//  Lincoln.java       Author: Lewis/Loftus
//
//  Demonstrates the basic structure of a Java application.
//********************************************************************

public class Lincoln
{
   //-----------------------------------------------------------------
   //  Prints a presidential quote.
   //-----------------------------------------------------------------
   public static void main (String[] args)
   {
      System.out.println ("A quote by Abraham Lincoln:");

      System.out.println ("Whatever you are, be a good one.");
   }
}
```

Output

```
A quote by Abraham Lincoln:
Whatever you are, be a good one.
```

```java
//*******                                             ***********
//  Lincol
//
//  Demons
//********************************************************************

public class Lincoln
{
   //-----------------------------------------------------------------
   //  Prints a presidential quote.
   //-----------------------------------------------------------------
   public static void main (String[] args)
   {
      System.out.println ("A quote by Abraham Lincoln:");

      System.out.println ("Whatever you are, be a good one.");
   }
}
```

Java Program Structure

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

class header

class body

Comments can be placed almost anywhere

```
}
```

Java Program Structure

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

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


   }

}
```

method body

method header

Comments

- Comments 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 three forms:

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

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

/** this is a javadoc comment   */
```

Identifiers

- *Identifiers* are the "words" in a program
- A Java identifier can be made up of letters, digits, the underscore character (_), and the dollar sign
- Identifiers cannot begin with a digit
- Java is *case sensitive*: `Total`, `total`, and `TOTAL` are different identifiers
- By convention, programmers use different case styles for different types of identifiers, such as
  - *title case* for class names - `Lincoln`
  - *upper case* for constants - `MAXIMUM`

Identifiers

- Sometimes the programmer chooses the identifier (such as `Lincoln`)

- Sometimes we are using another programmer's code, so we use the identifiers that he or she 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`	`else`	`interface`	`switch`
`assert`	`enum`	`long`	`synchronized`
`boolean`	`extends`	`native`	`this`
`break`	`false`	`new`	`throw`
`byte`	`final`	`null`	`throws`
`case`	`finally`	`package`	`transient`
`catch`	`float`	`private`	`true`
`char`	`for`	`protected`	`try`
`class`	`goto`	`public`	`void`
`const`	`if`	`return`	`volatile`
`continue`	`implements`	`short`	`while`
`default`	`import`	`static`	
`do`	`instanceof`	`strictfp`	
`double`	`int`	`super`	

Quick Check

Which of the following are valid Java identifiers?

```
grade
quizGrade
NetworkConnection
frame2
3rdTestScore
MAXIMUM
MIN_CAPACITY
student#
Shelves1&2
```

Quick Check

Which of the following are valid Java identifiers?

`grade`	Valid
`quizGrade`	Valid
`NetworkConnection`	Valid
`frame2`	Valid
`3rdTestScore`	Invalid – cannot begin with a digit
`MAXIMUM`	Valid
`MIN_CAPACITY`	Valid
`student#`	Invalid – cannot contain the '#' character
`Shelves1&2`	Invalid – cannot contain the '&' character

Program Development

- The mechanics of developing a program include several activities:
  - writing the program in a specific programming language (such as Java)
  - translating the program into a form that the computer can execute
  - investigating and fixing various types of errors that can occur
- Software tools can be used to help with all parts of this process

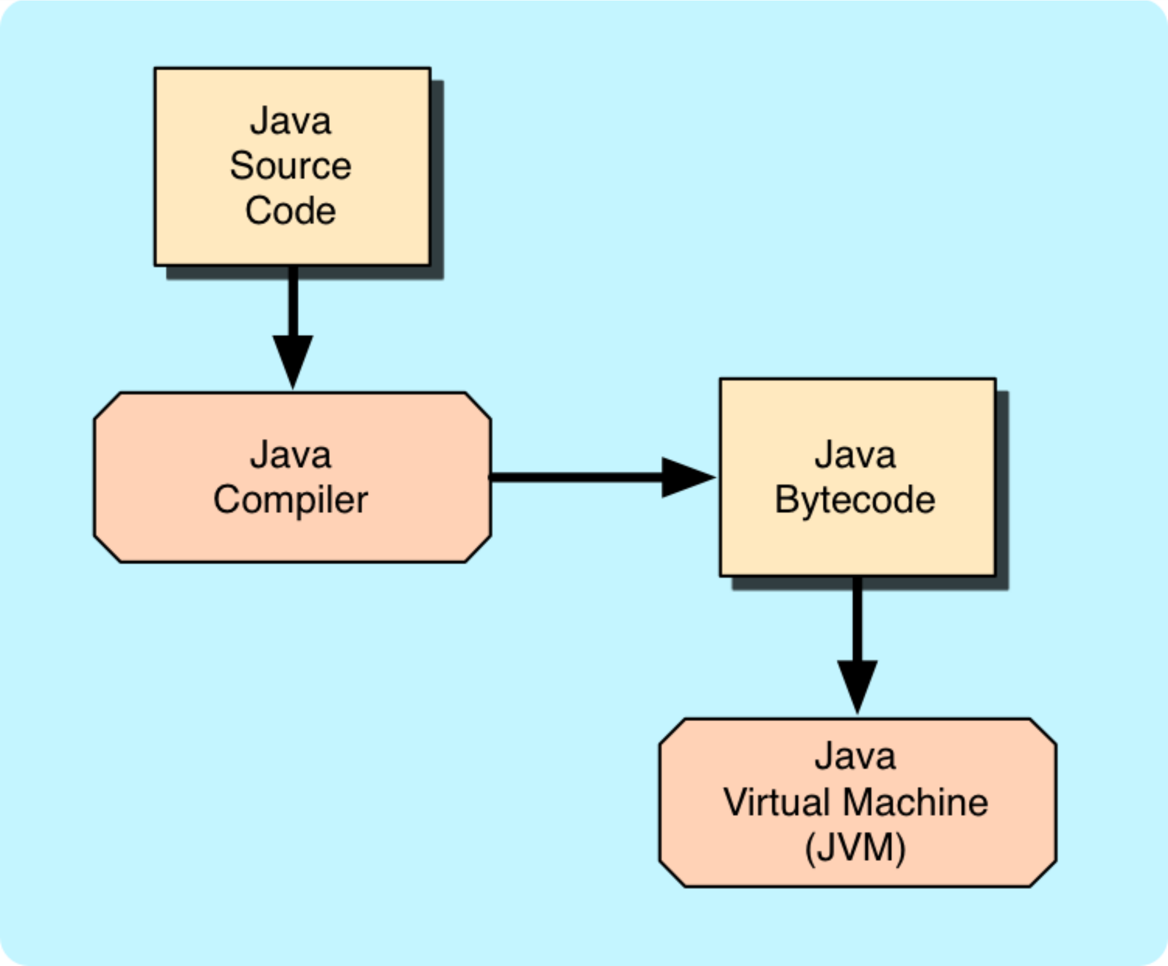
Language Levels

- 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*
- A program must be translated into machine language before it can be executed
- Sometimes, that target language is the machine language for a particular CPU type

Java Translation

- The Java compiler translates Java source code into a special representation called *bytecode*
- Java bytecode is not the machine language for any traditional CPU
- Bytecode is executed by the *Java Virtual Machine* (JVM)
- Therefore Java bytecode is not tied to any particular machine
- Java is considered to be *architecture-neutral*

Java Translation



Syntax and Semantics

- The *syntax rules* of a language define how we can put together symbols, reserved words, and identifiers 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 <u>meant</u> to tell it to do

Errors

- A program can have three types of errors
- The compiler will find syntax errors and other basic problems (*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, perhaps using an incorrect formula (*logical errors*)

Basic Program Development