



Introduction to computers and programming

Md. Mahadi Hasan Shaon
Lecturer, UGV

Course Logistics

Marking Distribution

Quiz 10%

Attendance 10%

Assignment & performance 10%

Lab Exam 20%

Total 50%

Final Grade/ Grand Total

Midterm: 20%

Final Term: 30%

Grand Total 50%

Content

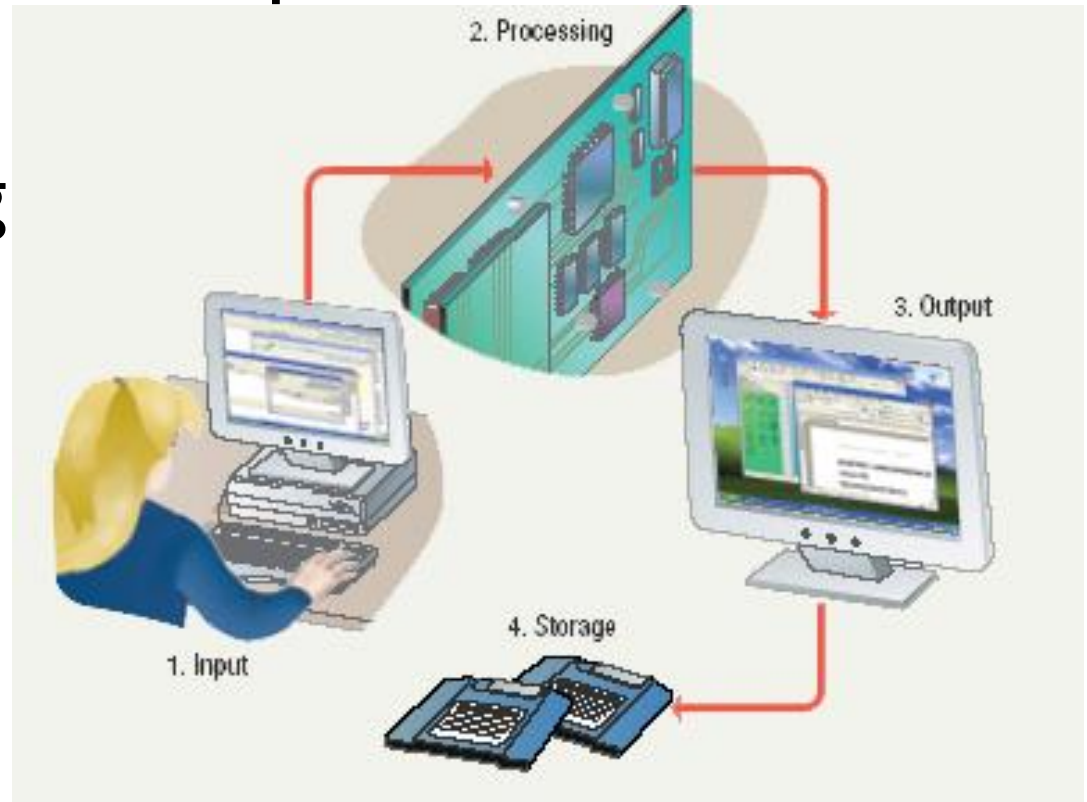
- Hardware
- Information storage
 - RAM, ROM
 - HD, DVD
- Display
 - Images
 - Characters (fonts)
- File system/type
 - Encryption
 - Compression
- Network
 - Protocols
 - Packets
- Programming
 - Algorithm
 - Pseudocode
 - Flowchart
- Languages
- Source code
 - Example

The computer

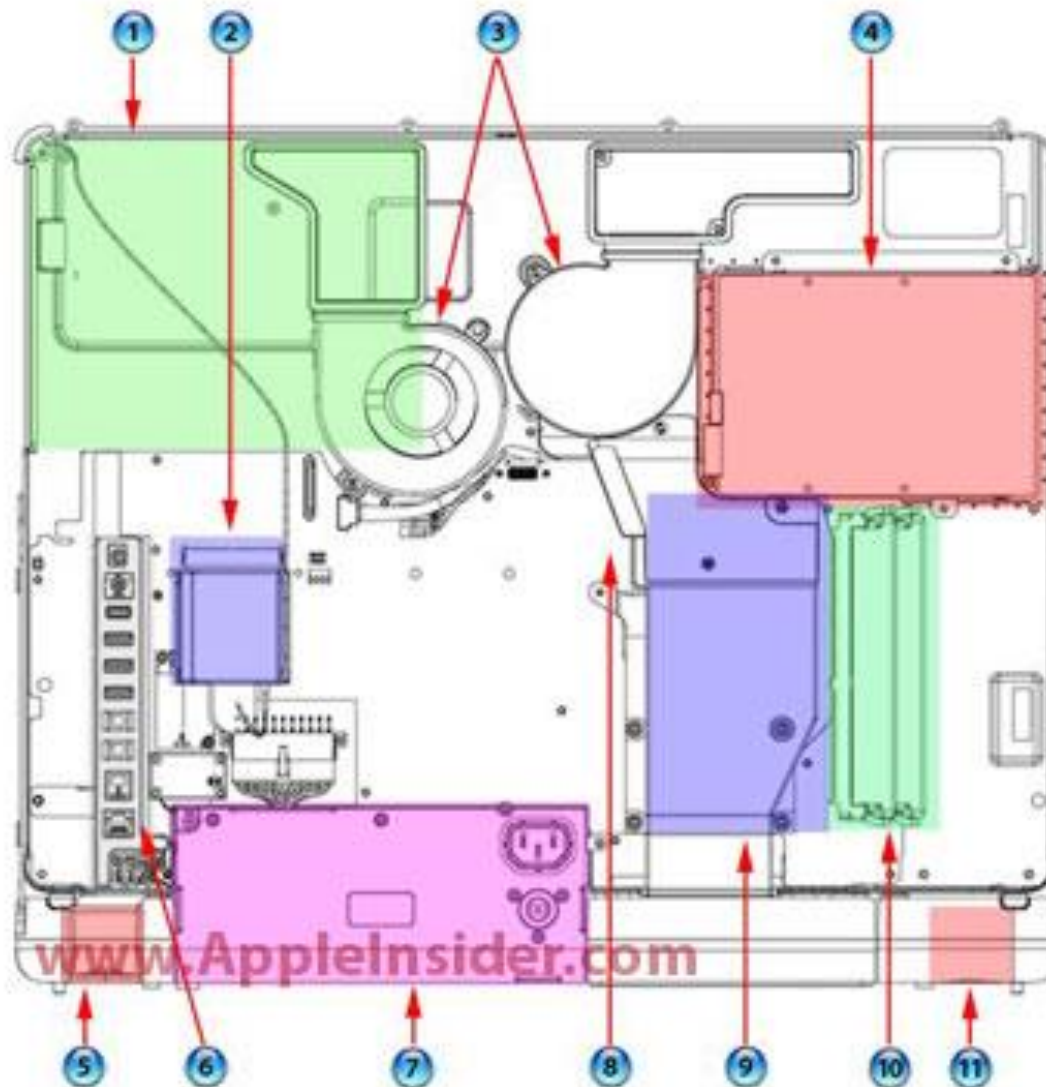


Information Processing Cycle

- Steps followed to process data
- Input
- Processing
- Output
- Storage

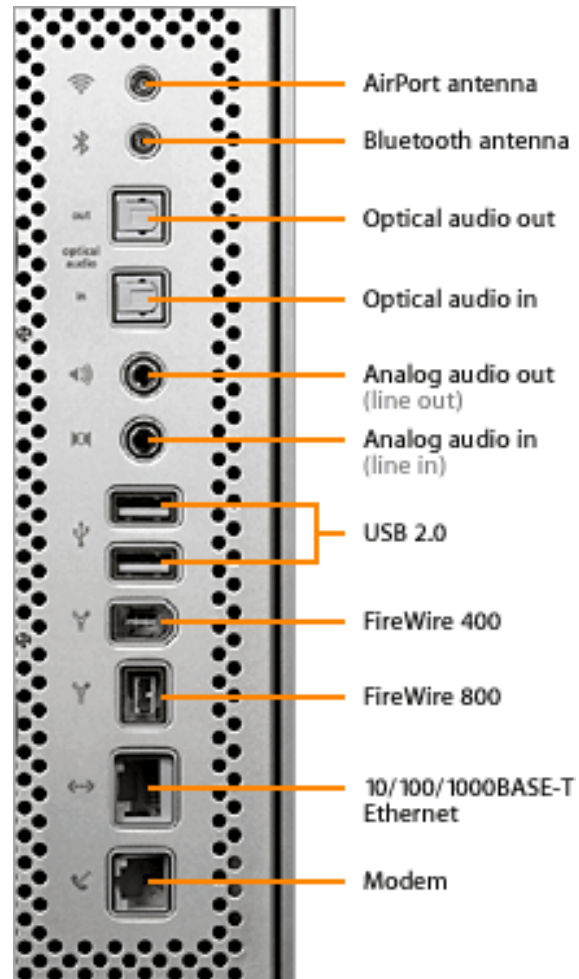


Internally



1. Optical drive
2. AirPort Extreme Card slot
3. System blowers
4. Hard drive
5. Right speaker
6. I/O ports
7. Power supply
8. Diagnostic LEDs (approximate location)
9. Power PC G5 processor
10. DIMM slots
11. Left speaker

The connections





RAM/ROM

- Memory devices
 - Stores data or programs
 - Random Access Memory (RAM)
 - Volatile
 - Stores current data and programs
 - More RAM results in a faster system
 - Read Only Memory (ROM)
 - Permanent storage of programs
 - Holds the computer boot directions

RAM/ROM

Memory mapping

Address

Values

0000

0001

0002

..

..

..

..

FFFF

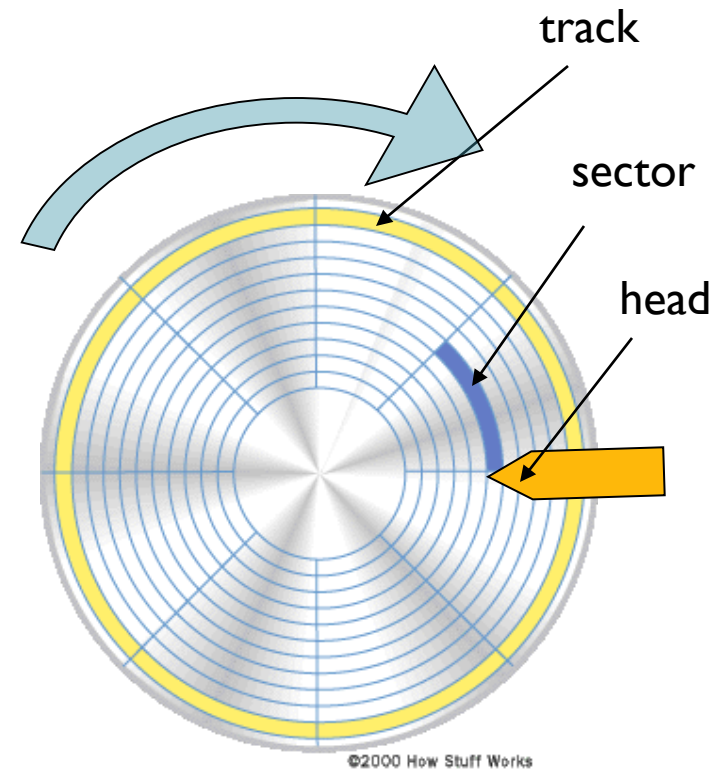
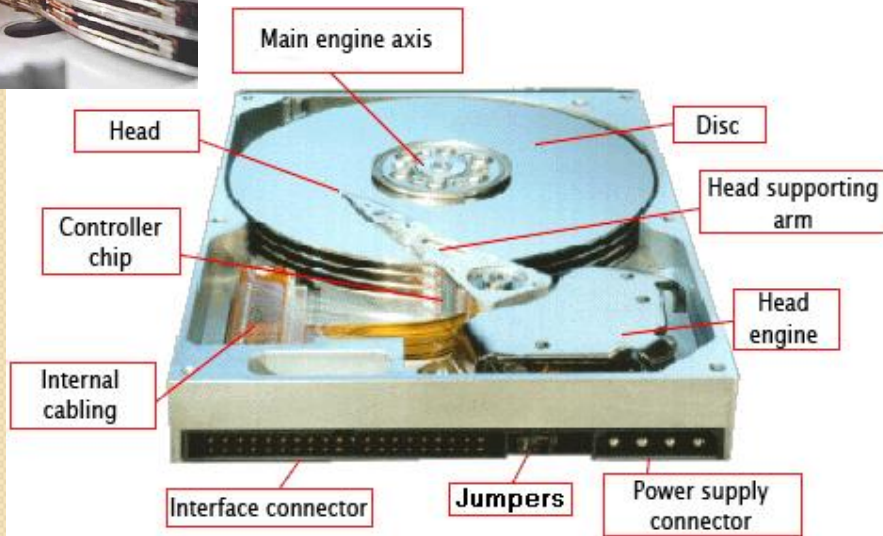
1	0	1	1	1	0	0	1
0	0	0	0	1	1	1	1
1	1	1	0	0	0	1	1
0	0	1	0	0	1	0	0

- Size reminder:

- Kilobyte Kb $2^{10} \sim 10^3$
- Megabyte Mb $2^{20} \sim 10^6$
- Gigabyte Gb $2^{30} \sim 10^9$
- Terabyte Tb $2^{40} \sim 10^{12}$
- Petabyte Pb $2^{50} \sim 10^{15}$
- ...

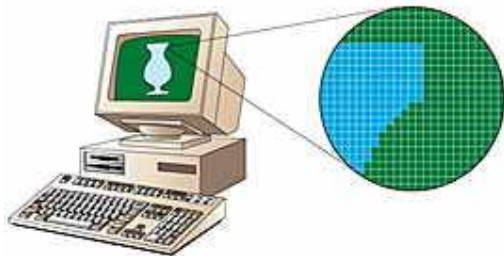


HD/DVD



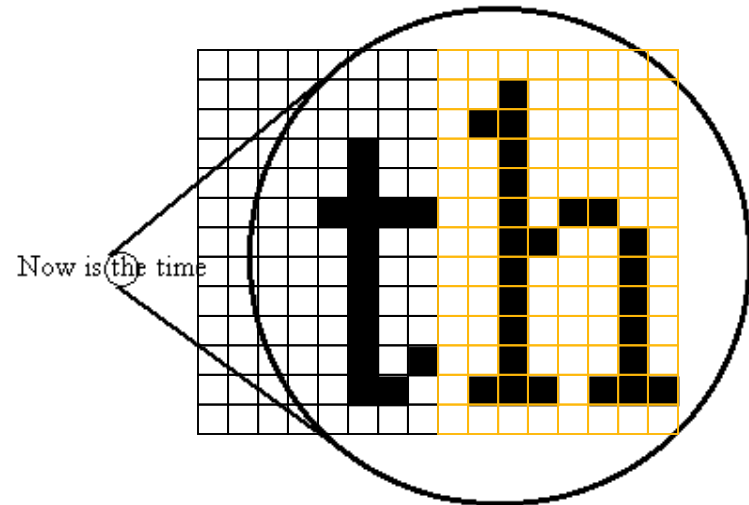
Display

- Screen pixel or dots
- Color coding: 32 bits
- 1 pixel = 3 bytes of color (RGB)+1 byte alpha channel (transparency)
- 1600x1200x4bytes=7,5 Mb!



• Fonts

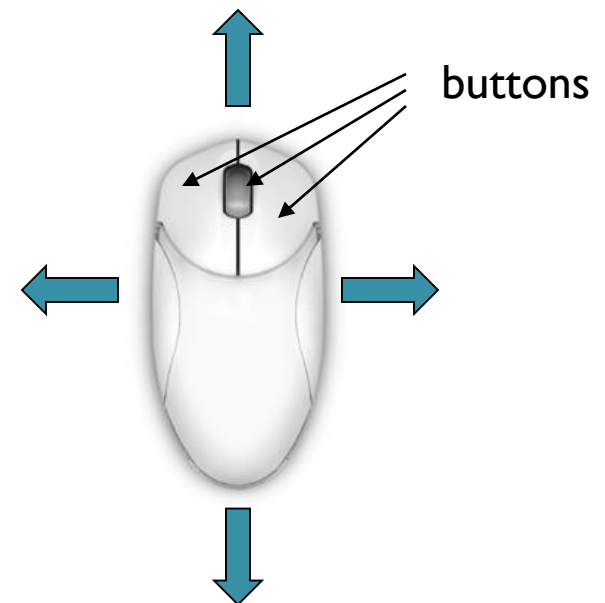
- Bitmap
- Vectors
 - TrueType, OpenType, PostScript





Keyboard/Mouse

- Each key (or combination of keys) of the keyboard sends a code to the computer.
- The code is interpreted and converted to the corresponding ASCII or Unicode number.
- The mouse movements
 - 1 to 4 bytes (vertical & horizontal)
- Buttons
 - Clicked, pressed, rolled



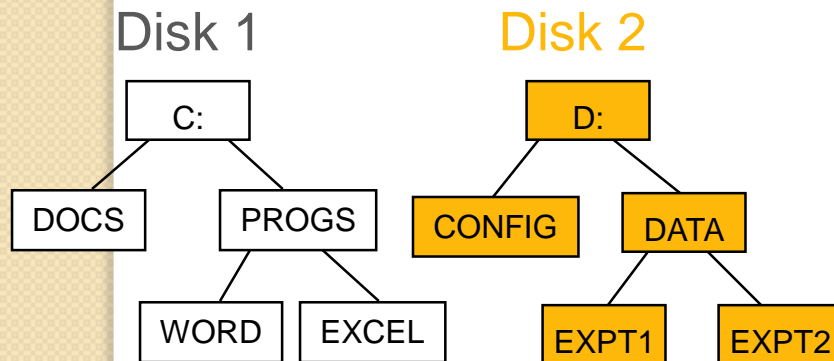
Software layers

Files	Data (.doc, .mp3)		
Software	Word, scripts, mail, web browser...		
User Interface	Quartz	CLI, X-Windows	Windows
Operating System	MacOSX	Linux	Windows
ROM	Firmware	BIOS	
Hardware	CPU, RAM, HD, DVD		

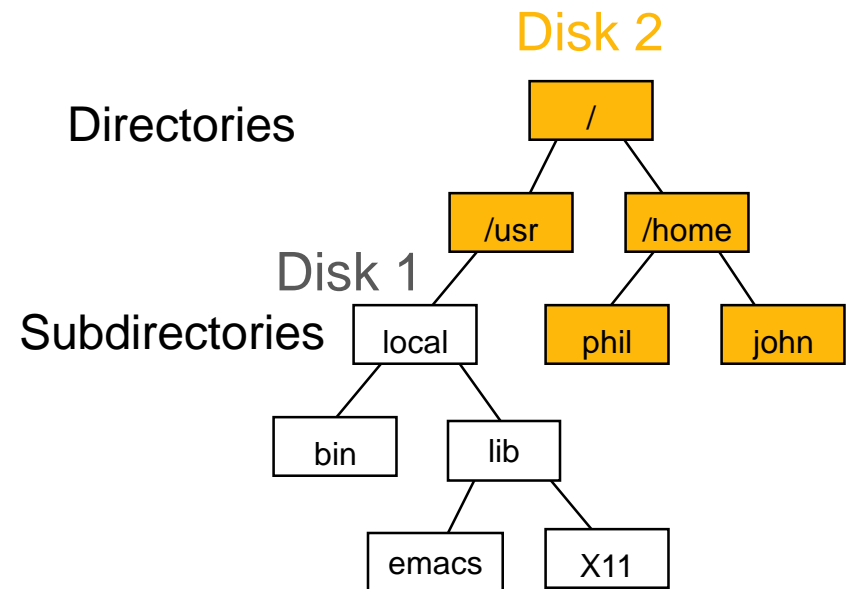
File systems

- Method the OS uses to store information
 - Storage unit, directories, subdirectories (Windows, VMS)
 - Single arborescence (Linux, MacOSX, all Unix)
- What exactly is a file?
 - a piece of information (text, graphic, data, music program, script)
 - it is identified by a name and an logical address (or path)
 - other informations: date, size, type, creator, ownership, physical address...

File system organisation



Windows



Unix

Path

- The path is the logical address used by the system or the user to locate a file.
- Example:

/bd_du_Palais/35/etage/4/appart/12/Dupont_Jean.txt

↑
path

↑ filename ↑ suffix

File types

- Executable
 - .exe
 - .app
 - Unix requires « x »
- Data
 - Text (.txt)
 - Music (.mp3)
 - Image (.jpg, .gif)
 - Movie (.mpg, .mov)
 - Binary (.bin)
- Special cases in Unix
 - STDIN
 - STDOUT
 - STDERR

Encryption / compression

- Compression
 - Reducing the size of files
 - E.g., .mp3, .gz, .jpg, .zip
- Encryption
 - Protecting your privacy
 - E.g., .pgp
- Packing
 - Grouping the files
 - E.g., .tar

Networks

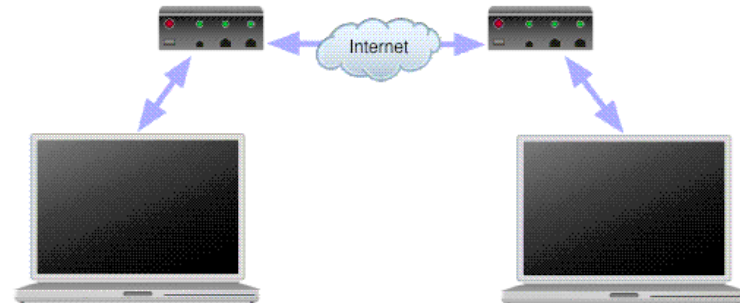
- Direct

- USB 11Mb-480Mb
- Ethernet 10Mb-1Gb



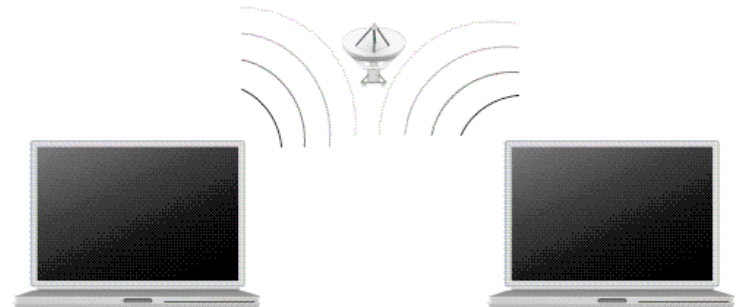
- Wired

- Modem 56Kb
- ADSL 600Kb-8Mb
- LAN 10Mb-10Gb



- Wireless

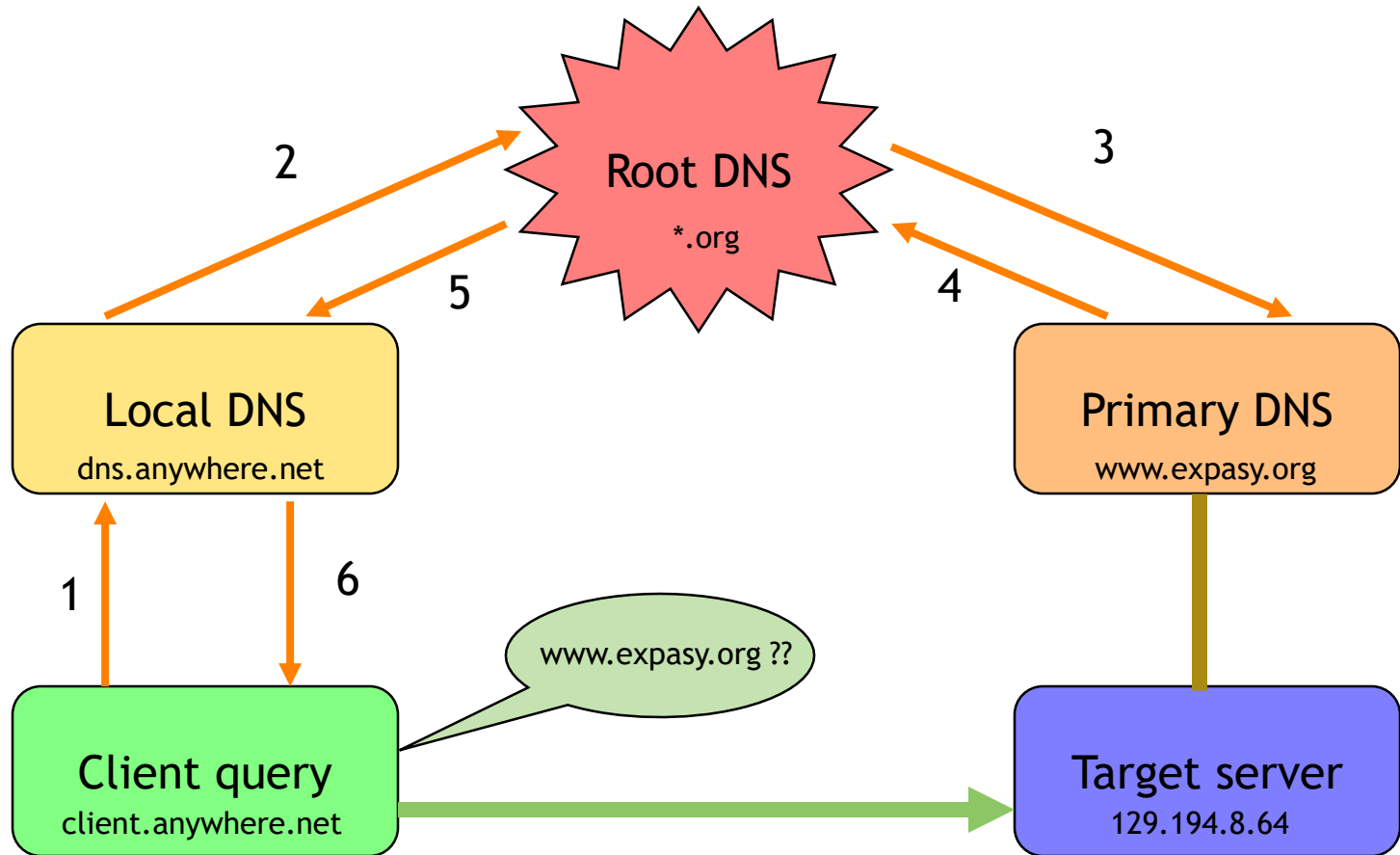
- Bluetooth 1Mb-20Mb
- WIFI (AirPort) 11Mb-54Mb



Network (ethernet or wireless)

- Computer talk to each other via network protocols
 - ip, tcp, http, ftp, ...
- TCP/IP
 - transmission control protocol/internet protocol
- DNS
 - ♦ Domain Name Server
- URL
 - ♦ Universal Resource Locator
- IP address
 - ♦ 192.42.197.51

DNS reminder

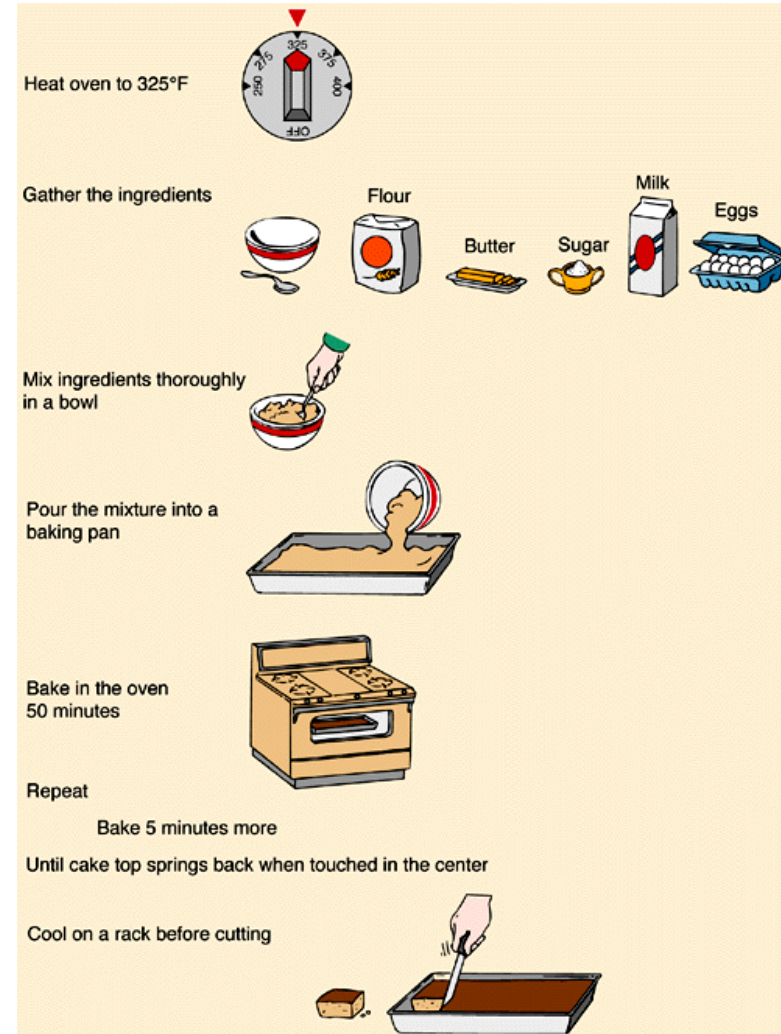




Introduction to programming

What is a program?

- How to cook?
- The algorithm
- Are you a programmer?



Pseudocode

- This is the pseudocode for a game of Monopoly

Main Procedure Monopoly_Game

Hand out each player's initial money.

Decide which player goes first.

Repeat

 Call Procedure Monopoly_Move for next player.

 Decide if this player must drop out.

Until all players except one have dropped out.

Declare the surviving player to be the winner.

Procedure Monopoly_Move

Begin one's move.

Throw the dice.

Move the number of spaces on the board shown on the dice.

If the token landed on "Go to Jail,"

 then go there immediately.

Else if the token landed on "Chance" or "Community Chest,"

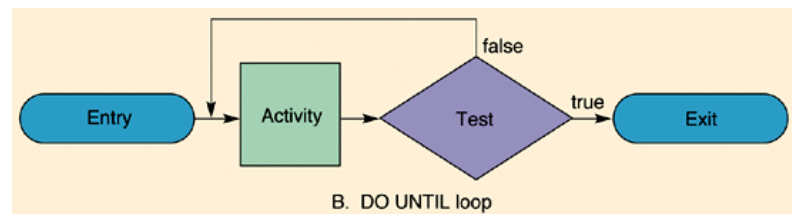
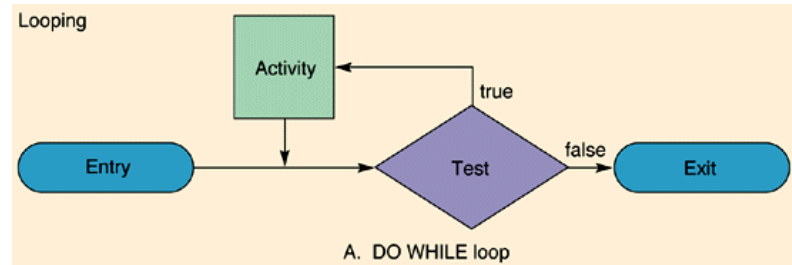
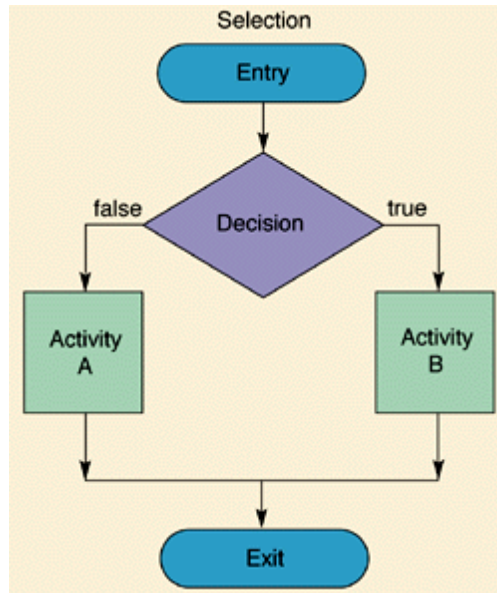
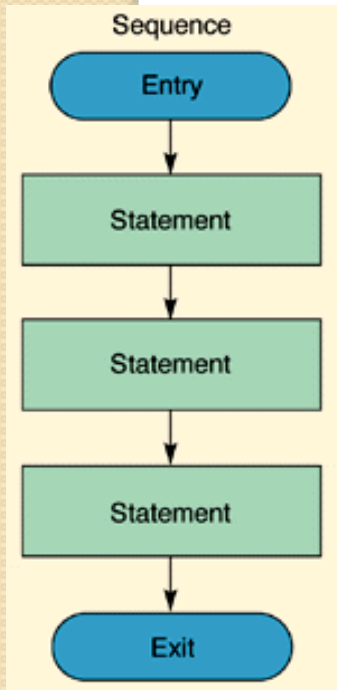
 then draw a card and follow its instructions.

Else

 follow the usual rules for the square (buying property,
 paying rent, collecting \$200 for passing "Go", etc.).

End one's move.

Flowcharts details



Languages

- Low level (processor dependent)
 - Machine code, assembler
- High level: structured, procedural
 - Fortran, C, Pascal...
- High level: object oriented
 - C++, Java, C#, Perl, Objective-C...
- Virtual machines
 - Java, C#...
- Scripting
 - Perl, Python, JavaScript...

Source code -> Object code

- **Compiler+linker**
 - Fortran, C, Pascal, C++...
 - **Interpreter**
 - Basic, Perl...
 - **Intermediate**
 - Java
- Compiler+linker
 - ♦ Fast to execute, but slow to debug
 - Interpreter
 - ♦ Slow to execute, but fast to debug (no need to recompile)
 - Intermediate
 - ♦ Slow...

Source code

- Instructions
 - Statement, blocks
 - Affectation
 - Operators
 - Loops
 - Tests
 - Subroutines
 - Comments
- Data structures
 - Variable
 - List
 - Array
 - Hash
 - Pointers
 - Objects

Source code (2)

- Statement, blocks
 - One or more instructions for the processor
- Affectation
 - Change to a variable
- Operator
 - affect one or more variable
 - + * - / AND OR NOT...
- Variable
 - A region in memory that can be modified
 - Exists in different types
 - Scalar, char, numeric, boolean
 - List, array
 - Hash
 - Combination->data structure

Source code (3)

- Loops
 - Allow the computer to repeat blocks
- Tests
 - Decide what to do
- Subroutines
 - Programs frequently called (functions)
- Comments
 - The most important lines of the source code...
- Pointers
 - Reference to region in memory (address)
- Objects
 - Combination of data and code

Source code example

```
#!/usr/bin/perl -w # essential line of all perl scripts
```

```
$filename = "avoid.txt"; # affect « avoid.txt » to the variable $filename
```

```
# open the file, or exit
```

```
open(FILE, $filename) || die "Cannot open file '$filename'\n\n";
```

```
@text = <FILE>; # add each line of the file to an array
```

```
close FILE;
```

```
foreach $line (@text) { # read one line from the array into $line and repeat for each line
```

```
    @table = split(//, $line); # read each character of the line in an array
```

```
    while ($char = pop(@table)) { # read one character of the array 'table' and repeat for all
```

```
        $char =~ s/[^a-z]//; # keep only the alphabetical character a to z
```

```
        if ($char) { # check if the character exists and execute the block
```

```
            $count{$char}++; # if yes, increment by one the hash 'count'
```

```
        }
```

```
    }
```

```
}
```

```
# print each character and its number of occurrence one per line
```

```
foreach $c (keys %count) {
```

```
    print "$c=$count{$c}\n";
```

```
}
```

```
exit; # quit the program
```

Tips

- Think about your problem
- Create a flowchart
- Create the pseudocode
- Verify the memory used by your variables
- Write the code
- Test the code
 - For all the possible functions or cases (if possible)
 - Give it to users as a beta (if not possible)
 - Sell it (if you work for Microsoft®... ;-)
- Debug