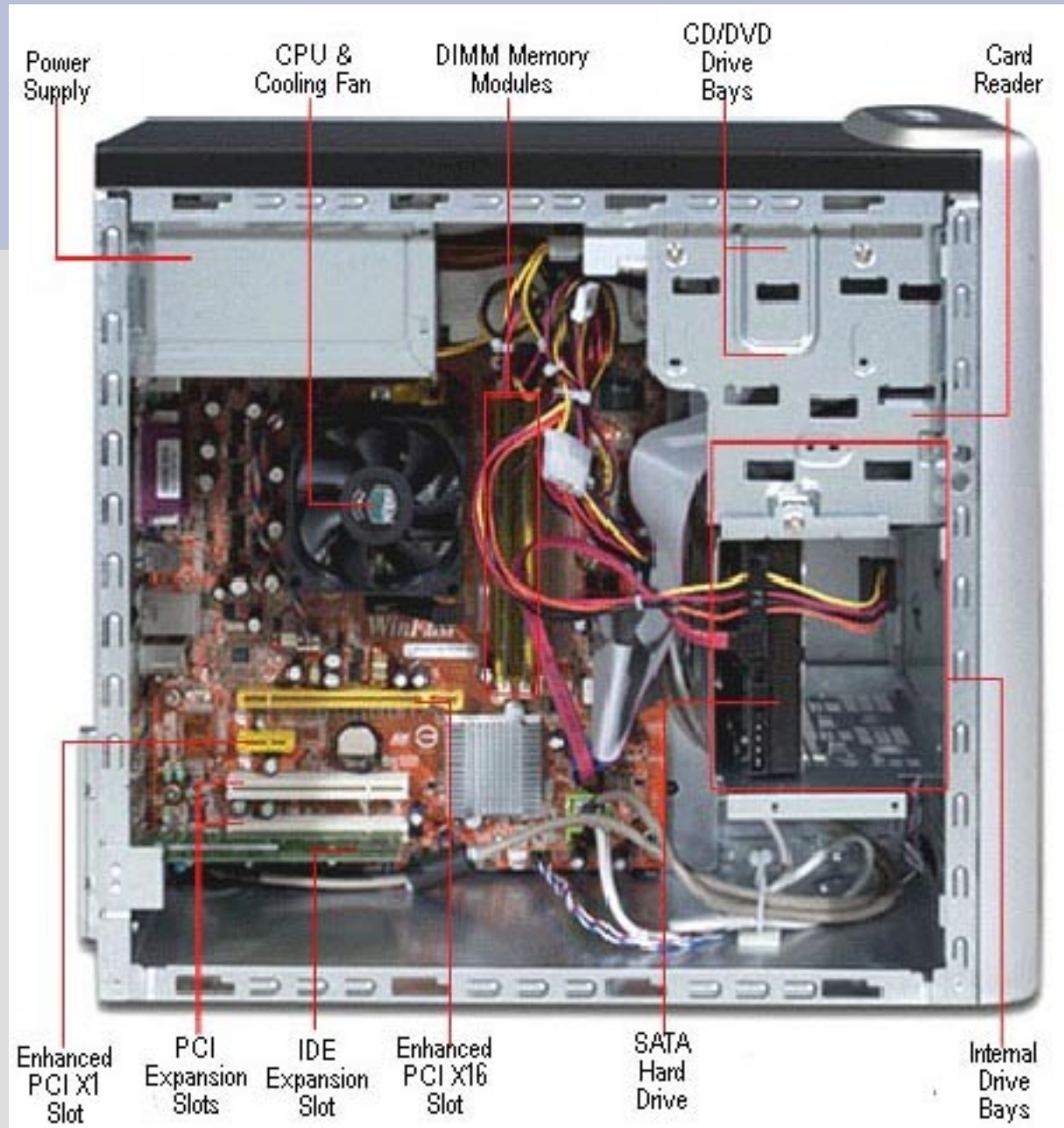
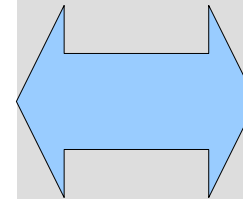


# Computer Basics





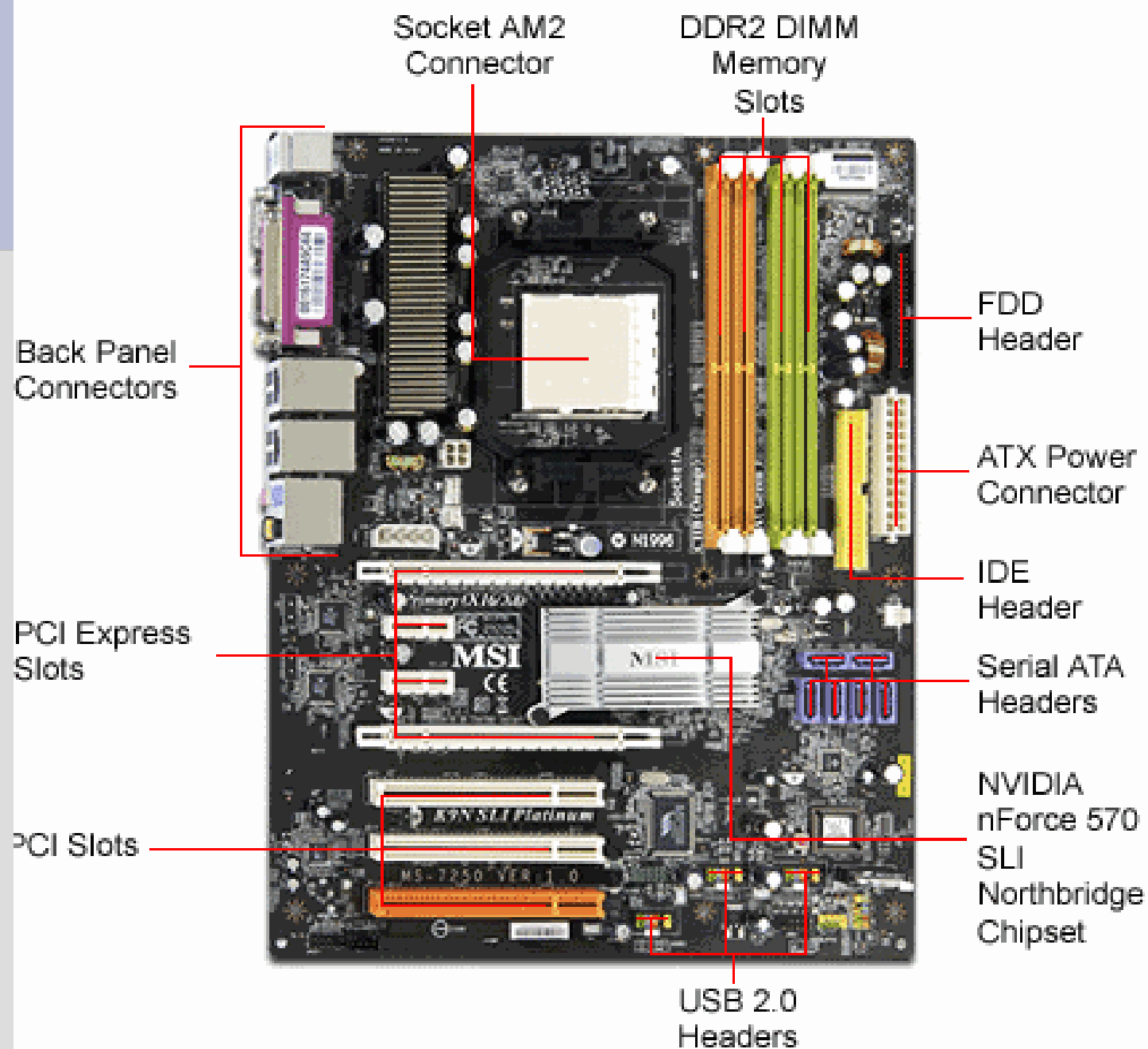
Mother Board



System Drive

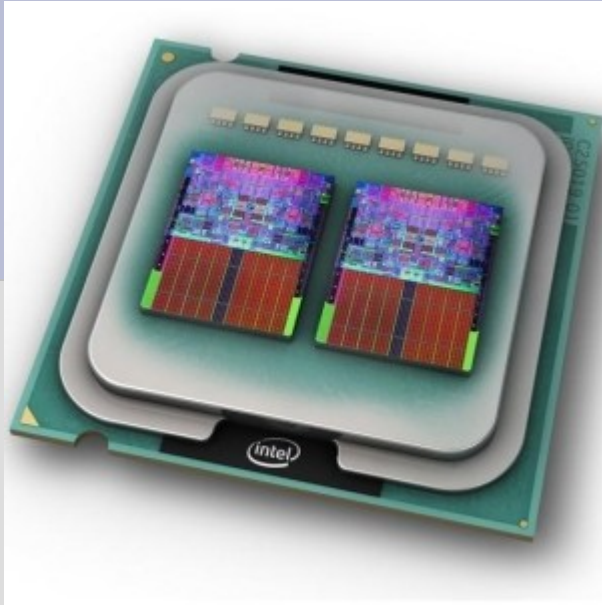


CD Rom Drive



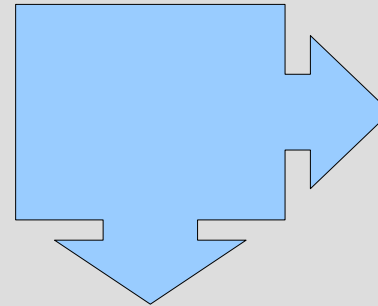


CPU

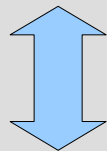


Video Card

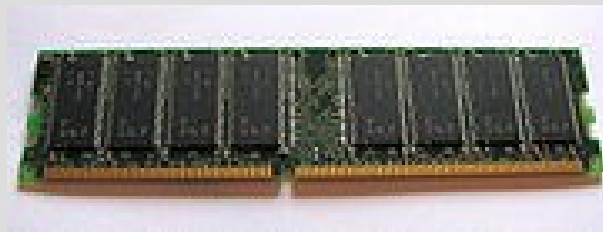
System bus



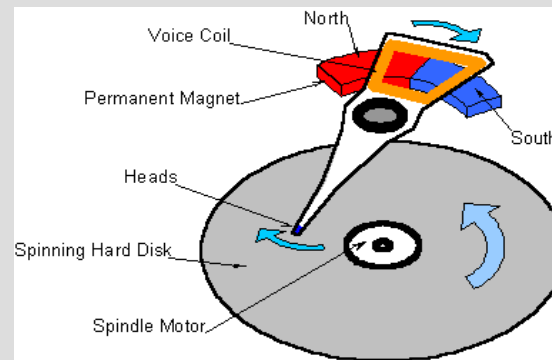
Sound Card



System bus

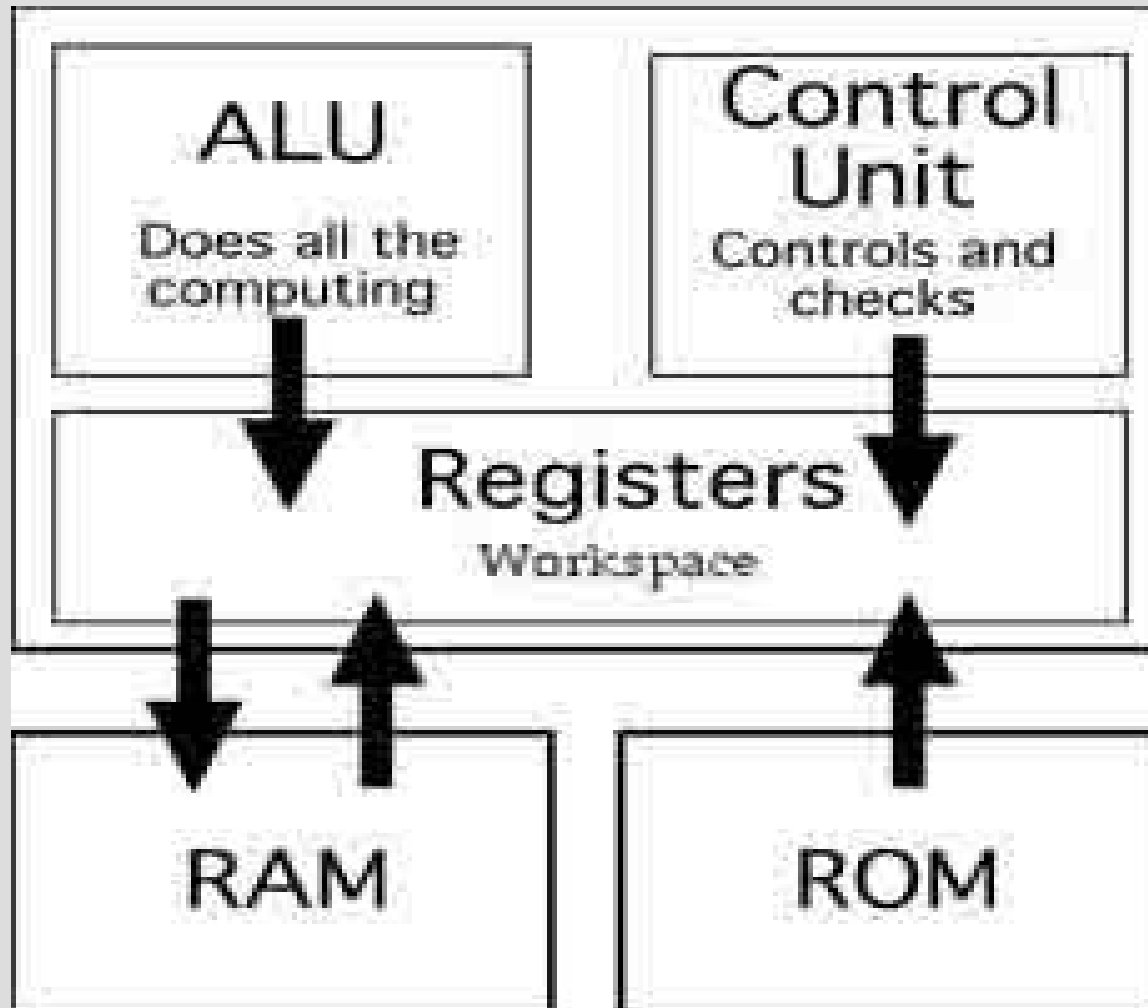


Computer Memory



Disk drivers

# The CPU



cache

# What Do These Parts Do?

- CPU
  - Stores numbers in special (very fast) memory locations called **registers**.
  - Adds, subtracts, multiplies and divides numbers in registers. Stores results in registers.
  - Get numbers from specific RAM locations via the bus. Some numbers are data, some are instructions. Each data item or instruction has a location in RAM where it is stored, ie a memory address
  - Test value (positive, negative, zero) of register and change instruction number (memory address of next instruction) based on result.
  - Write numbers to bus.

# What Do These Parts Do?

- **Bus**
  - Moves numbers from one device to another. Devices are described by numbers (bus addresses). Each device signals bus to send some amount of data (numbers) to another device. Bus signal other device to read incoming data.
- **RAM and Disk Drives**
  - Receives memory address and data and stores data in specified address or sends data from a specified address.
- **Video Card and Sound Card**
  - Receives data and instructions from bus and controls video or sound device based on instructions received. Writes data to bus in response to instructions received.



# That Does Not Seem to Compute

- How these pieces go together to do what we know a computer can do (play music, do spreadsheets, play videos, etc) is not easy to see.
- Understanding what a computer program (the instructions the CPU receives) does is the first step toward understanding how to make a computer do fancy things.

# Computer Software(program) Components

- Operating System: Software that controls the interaction of the user, software applications, and the hardware. Examples: Microsoft WindowsXP, Microsoft Vista, Linux, SunOS, HPUX.
  - When the computer starts it gets its instructions from a section of memory called the boot ROM (Read Only Memory). The initial program stored in the boot ROM reads into memory another program from the disk called the boot loader. After the boot loader is in memory, the startup boot program instructs the CPU to get its next instruction at the start of the memory it just loaded.
  - The boot loader then loads the next part of the operating system and then cause the CPU to execute the newly loaded instructions. This boot strapping procedure is repeated until the full operating system is loaded and executing. The process is known as booting the computer.

# Computer Software Components

- Drivers: programs that the operating system uses to control the hardware.
  - Drivers are hardware specific. When you install a new hardware component, it often requires a driver written specifically for it.
- Applications programs: By clicking on an icon or by typing in a command the user selects a program, the operating system loads the program into memory and instructs the CPU to execute its next instruction at the start of the newly loaded instructions.

# Computer Memory

- Computer memory stores numbers as a sequence of 0's and 1's.
- A single 0 or 1 is called a ***bit***.
- PC's and most other computers organize memory into collections of 8 ***bits***, called a ***byte***.
- Modern computers have from 500 million bytes to 10's of billion bytes.
- One byte can hold 256 numbers from 0 to 255.

# Computer Memory

- Zero is 00000000 and 255 is 11111111.
- Other numbers are shown below.
- 

128	64	32	16	8	4	2	1	
0	0	1	1	1	1	0	0	60
0	1	0	0	0	0	1	0	66
1	0	1	0	0	1	0	1	165
1	0	0	0	0	0	0	1	129
1	0	1	0	0	1	0	1	165
1	0	0	1	1	0	0	1	153
0	1	0	0	0	0	1	0	66
0	0	1	1	1	1	0	0	60

# Computer Instructions

- A program to add the value of the contents of register A to the contents of register B and store the results in register B
- In computer memory this might look like

Memory Addr	Instruction	mnemonics
10110110111101101110111100011101	10000000000000000000000000000000	CLA
10110110111101101110111100011110	1000000000000000000000000000010101	ADD B
10110110111101101110111100011111	1000000000000000000000000000010110	ADD A
10110110111101101110111100100000	10000000000000000000000000000110101	STR B



# Computer Languages

- Originally, programmers (in the before 1950's) wrote programs using the binary code for the instructions ie the 0's and 1's—Machine language programming
- As that was exceedingly difficult to get correct, programmers soon switched to writing in the mnemonics and having the computer fill in the binary numbers—Assembly language programming.

# Computer Languages

- But assembly language was still hard to program in and was different for each type of computer.
- It is easier to think of the operations mathematically, say as  $b=b+a$ . This led to the development of ***high level languages*** that express the operation one needs to perform on the numbers, rather than the machine instructions to do it.

# Computer Languages

- The first high level language Fortran became available in 1957.
- Other well-known high level languages are COBOL, C, C++, and Java.
- C/C++ code:

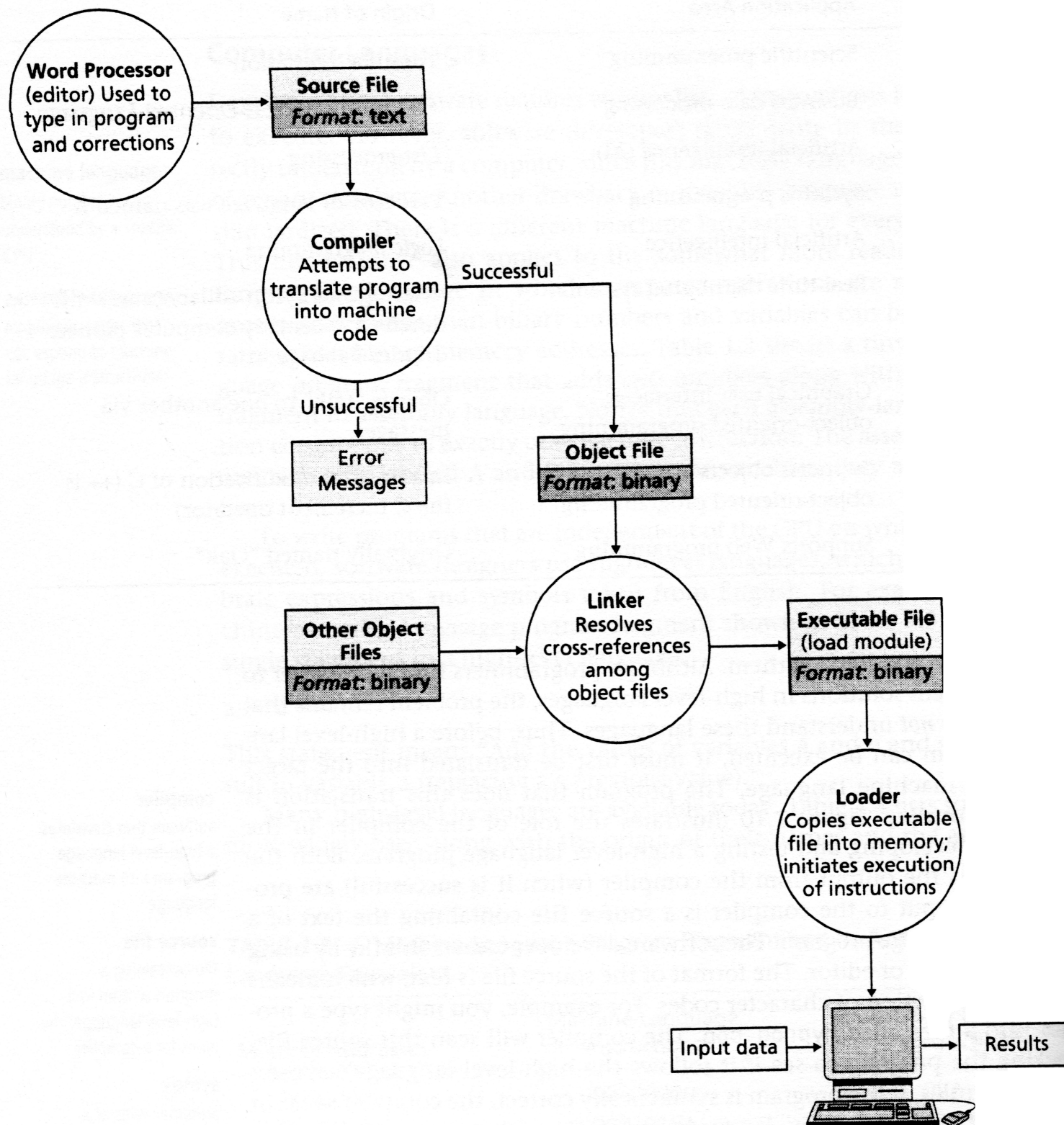
```
a=4;  
b=3.1;  
b=a+b;  
printf("b=%f\n",b);
```

# Compilers

- A high level language needs another program, called a ***compiler***, to translate the operations it describes into machine code (ie binary numbers that are machine instructions).
- Actually, modern compilers first perform an intermediate translation into what is know as an object file.

# Linker

- Object files usually contain reference to other programs that must be combined together before a working program is produced.
- A program called a ***linker*** resolves cross references between object files creating a complete program. Often the compiler invokes the linker when it is done translating the ***source file***.





# Object Oriented Languages

- Object oriented languages allow a programmer to group variables and computer code together to define a conceptual object such as a window on a computer screen.
- Once created an object can easily be specialized to an individual programmer's needs.
- This feature makes it easier to develop programs.
- C++ is an Object Oriented Language.