

# PIC 10A: Week 1b

Section 1C, Spring 2016

Prof. Michael Lindstrom (TA: Eric Kim)

v1.0

# Announcements

- Schedule of homeworks/quizzes are on PIC 10A course website:
  - <http://www.math.ucla.edu/~mikel/teaching/pic10a/>
  - Course Calendar: "Rough schedule" pdf
- Office hours in effect starting week 2 (next week)
  - See course website for office hour times



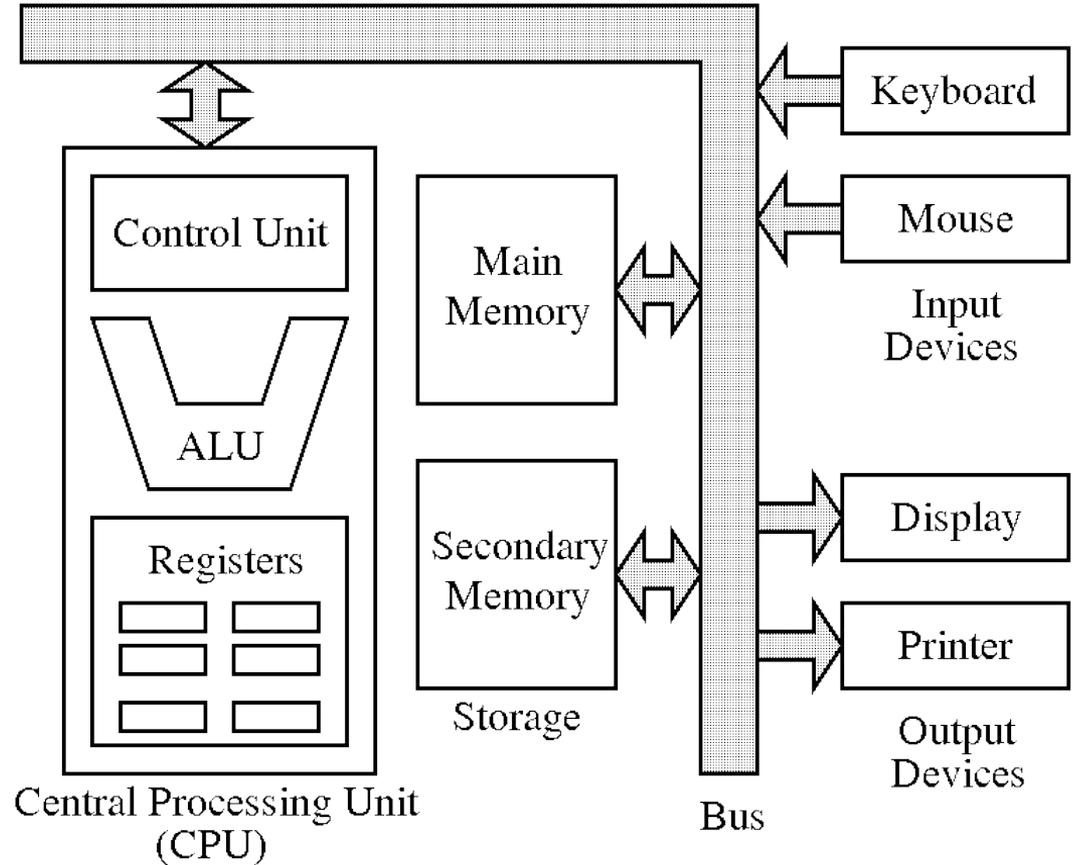
# Today

- Tour of Computer Architecture
- What is a Compiler?
- Visual Studio 2013 Demo
  - Compiling your first program

# Computer Organization

- CPU
- Wires, Transistors
- Memory
  - Random Access Memory (RAM)
  - Read Only Memory (ROM)
- Registers
- Bus
- Hard Disk

# (Simplified) View of a Computer



# (Simplified) View of a Computer

**CPU:** Central Processing Unit.

Purpose: Executes instructions (ie code), such as addition, multiplication, saving pictures to hard drive, etc.

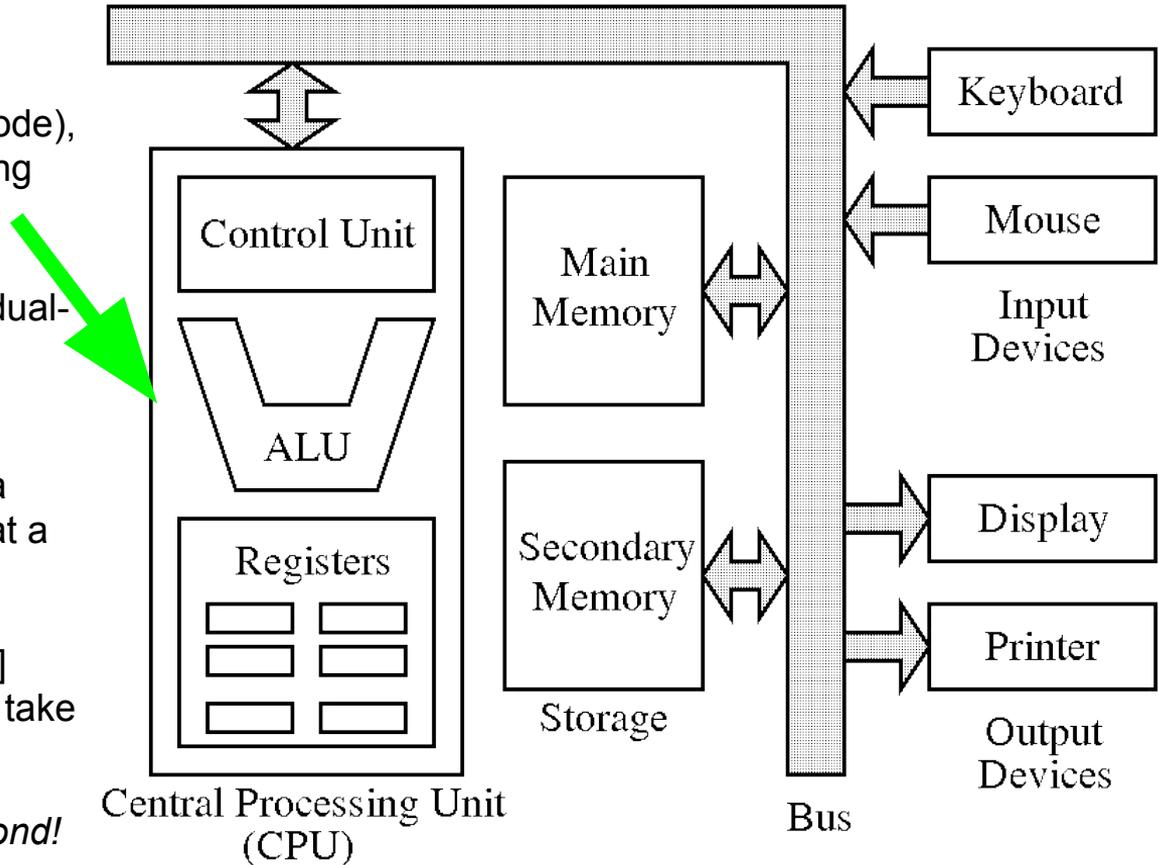
Ex: Macbook (2016) has a 1.2GHz dual-core Intel Core M processor.

1.2GHz: **Clock speed** of the CPU.

Similar to the pistons of an engine, a CPU is constantly performing work at a rate governed by its clock speed.

A clock speed of 1.2GHz [Gigahertz] means that (simple) operations only take **~1 nanosecond**.

*Light travels 30cm in 1 nanosecond!*



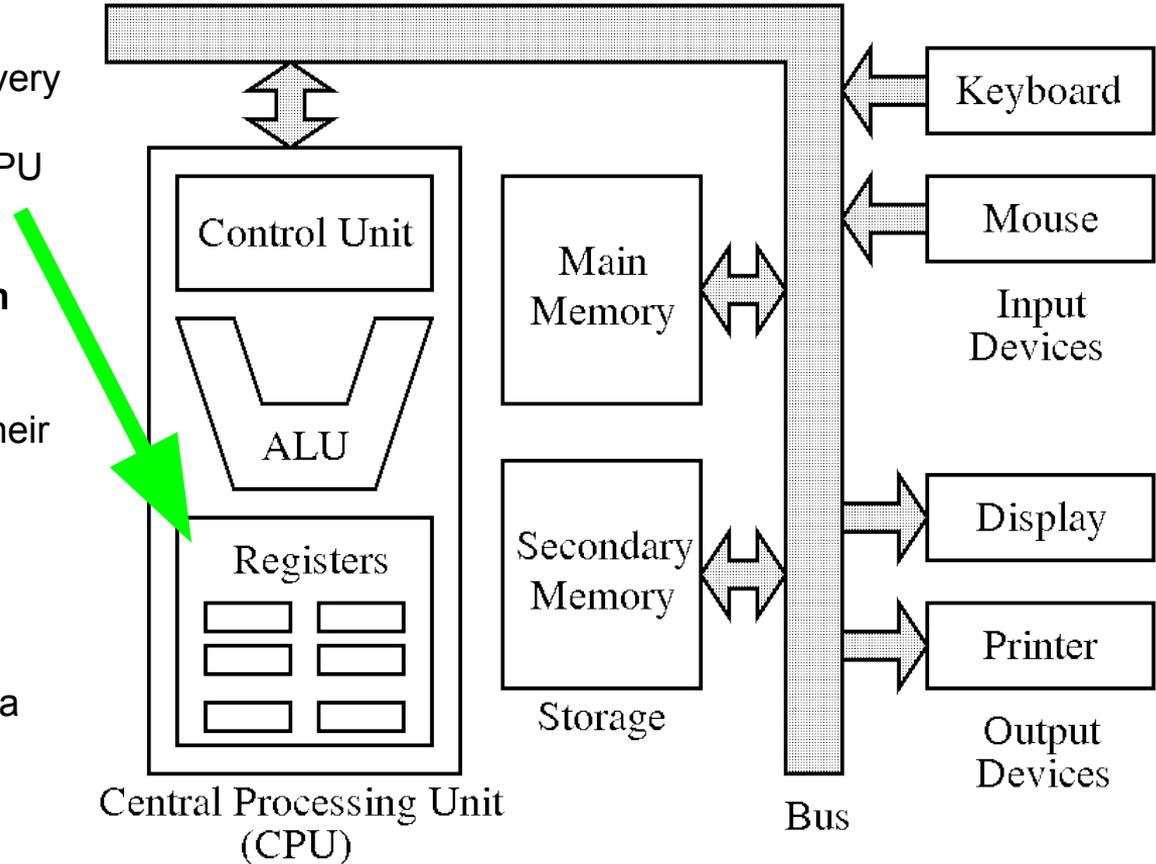
# (Simplified) View of a Computer

**Registers:** A device that stores a (very small) amount of data.

Purpose: "Scratch" space for the CPU to do its work.

A CPU can access a register **much faster** than accessing RAM or the hard disk. Thus, to speed up computation, computers try to do their work within registers as much as possible.

Ex: Most Intel chips (ie in most computers these days) have >32 registers, each capable of holding a single number.



# (Simplified) View of a Computer

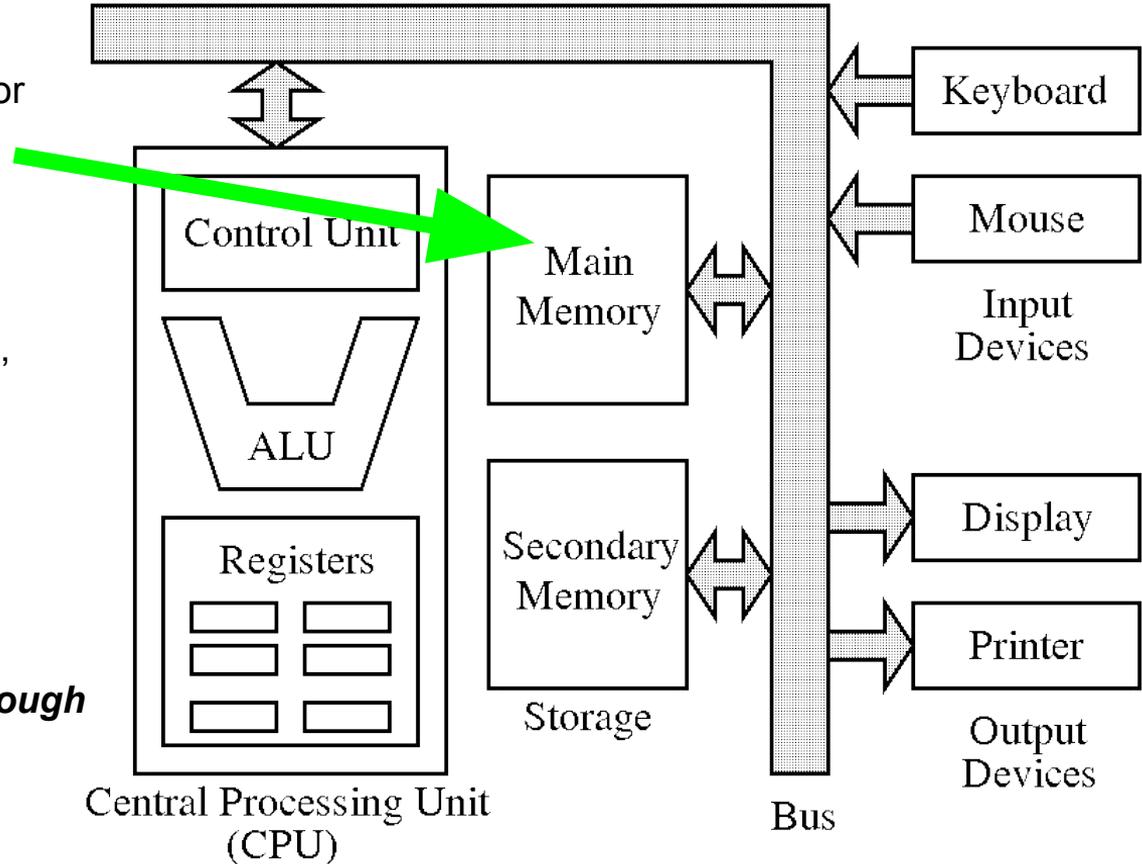
**RAM:** Random Access Memory.

Purpose: Additional "scratch space" for your CPU to do its work. "Working memory". Volatile.

When you run a program (ie Firefox), the program needs space to store its bookmarks, Facebook profile pictures, and text. The CPU will store this stuff into RAM.

In 2016, laptops tend to have 8 or 16 Gigabytes of RAM.

***"640 KB (of RAM) ought to be enough for anybody" -- Bill Gates, 1970's***



# (Simplified) View of a Computer

**RAM:** Random Access Memory.

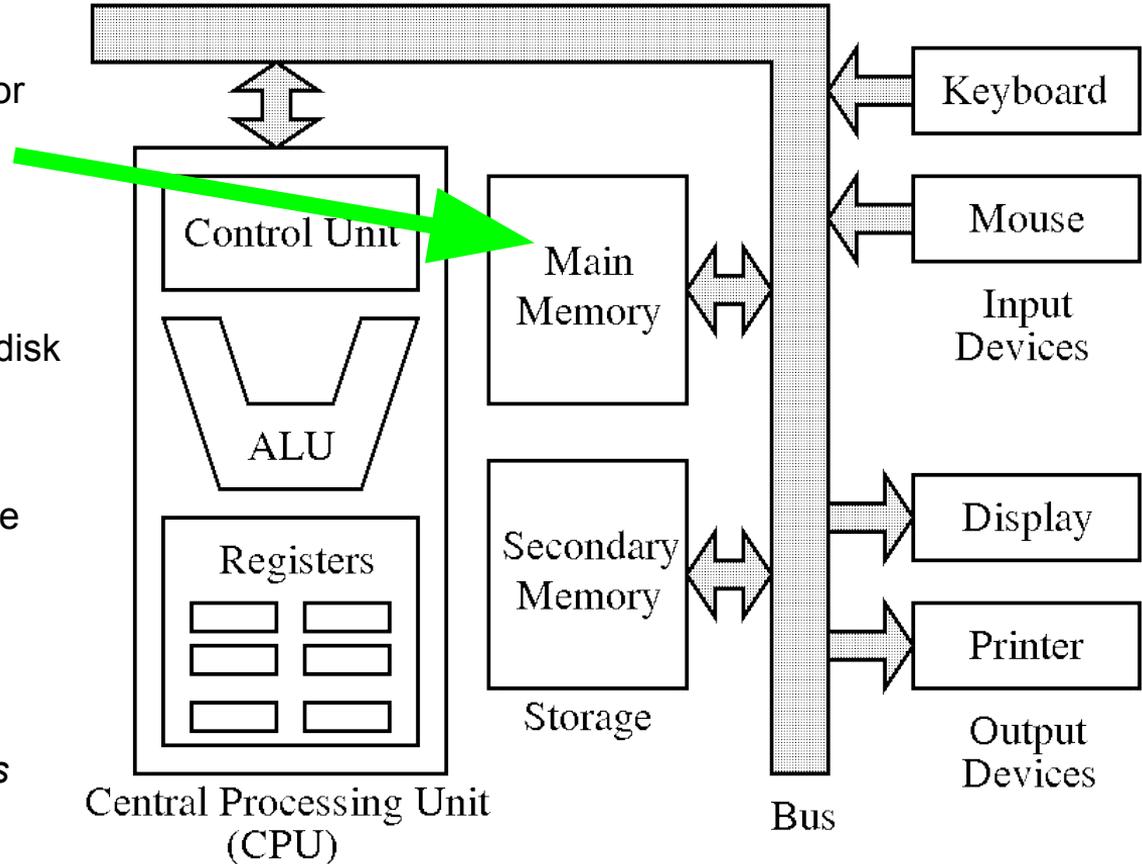
Purpose: Additional "scratch space" for your CPU to do its work. "Working memory". Volatile.

For most tasks, too few registers, and accessing disk is too slow.

RAM is a middle ground: **faster** than disk and **larger** than registers. But: slower than registers, smaller than disk.

**In a nutshell:** More RAM means more programs you can have open at the same time without running into slowdowns.

*For researchers like me: more RAM means I can process larger datasets without killing my computer :)*



# (Simplified) View of a Computer

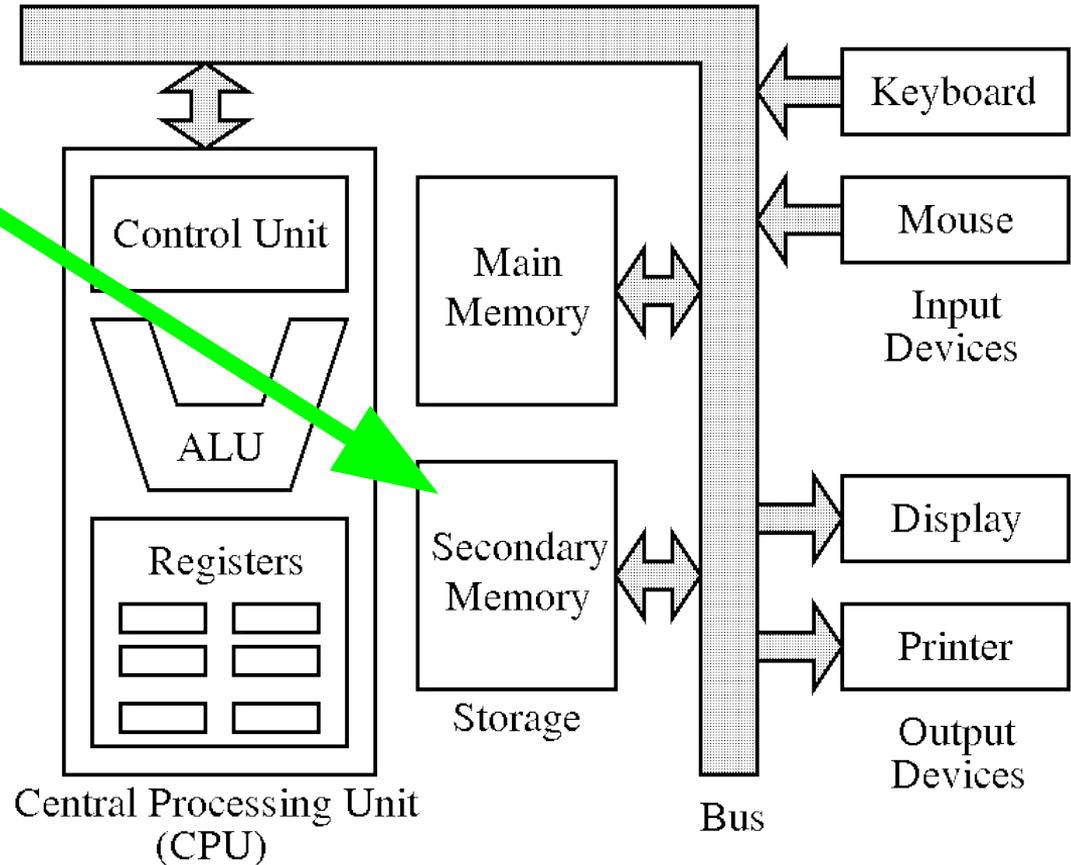
**Secondary Storage:** Ex: Hard Disk

Purpose: Long-term storage of data. Your pictures, movies, and PIC 10A code live here! Nonvolatile.

These days (2016), can buy a 2 TB external hard drive for ~80\$!

Caveat: Accessing disk-based hard drives is **\*SLOW\***, ie millions of times slower than RAM.

There are spinning magnetic disk(s) that store your data, like a record player.



# (Simplified) View of a Computer

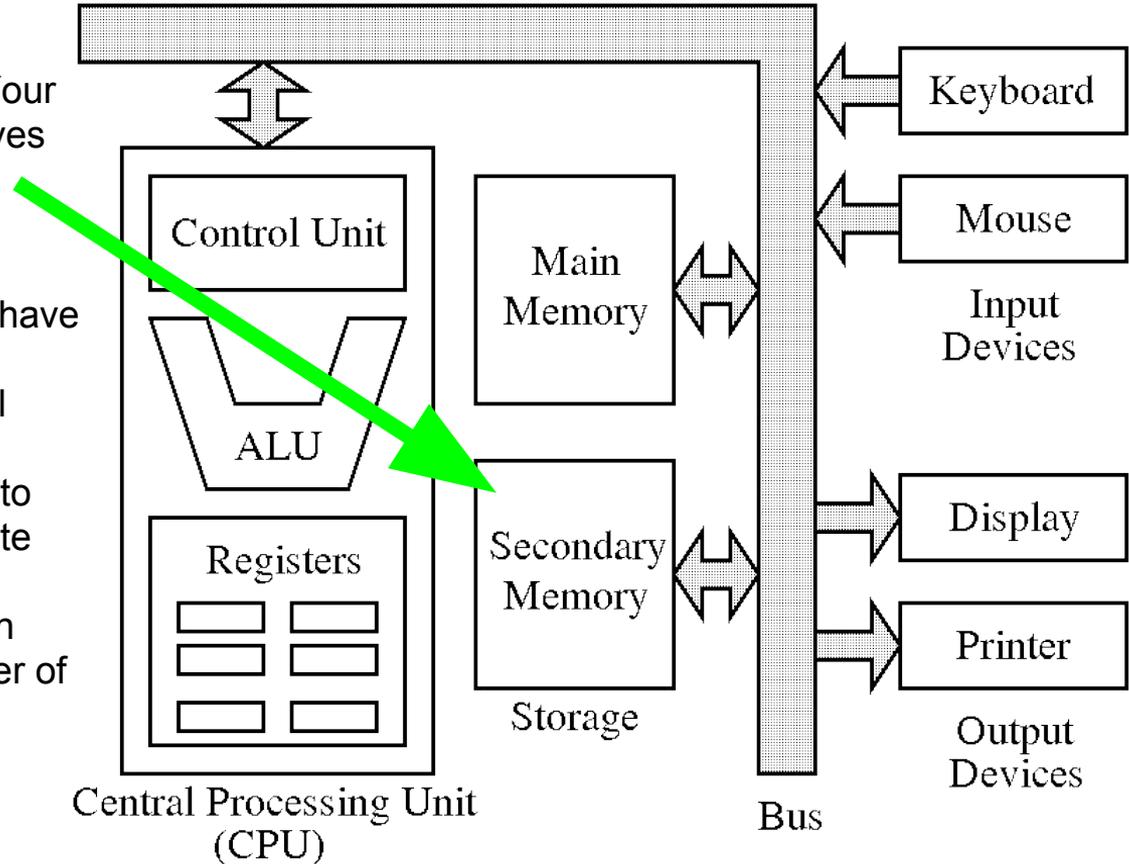
**Secondary Storage:** Ex: Hard Disk

Purpose: Long-term storage of data. Your pictures, movies, and PIC 10A code lives here! Nonvolatile.

New Trend: **Flash drives**, which offer \*much\* faster read/write speeds, and have no moving parts! Likely will see these replace disk-based drives for personal machines.

However, disk-based drives will likely to continue to be used on servers for quite some time, since:

- (a) Disk-based is cheaper than flash
- (b) Flash memory has a little number of read/writes.



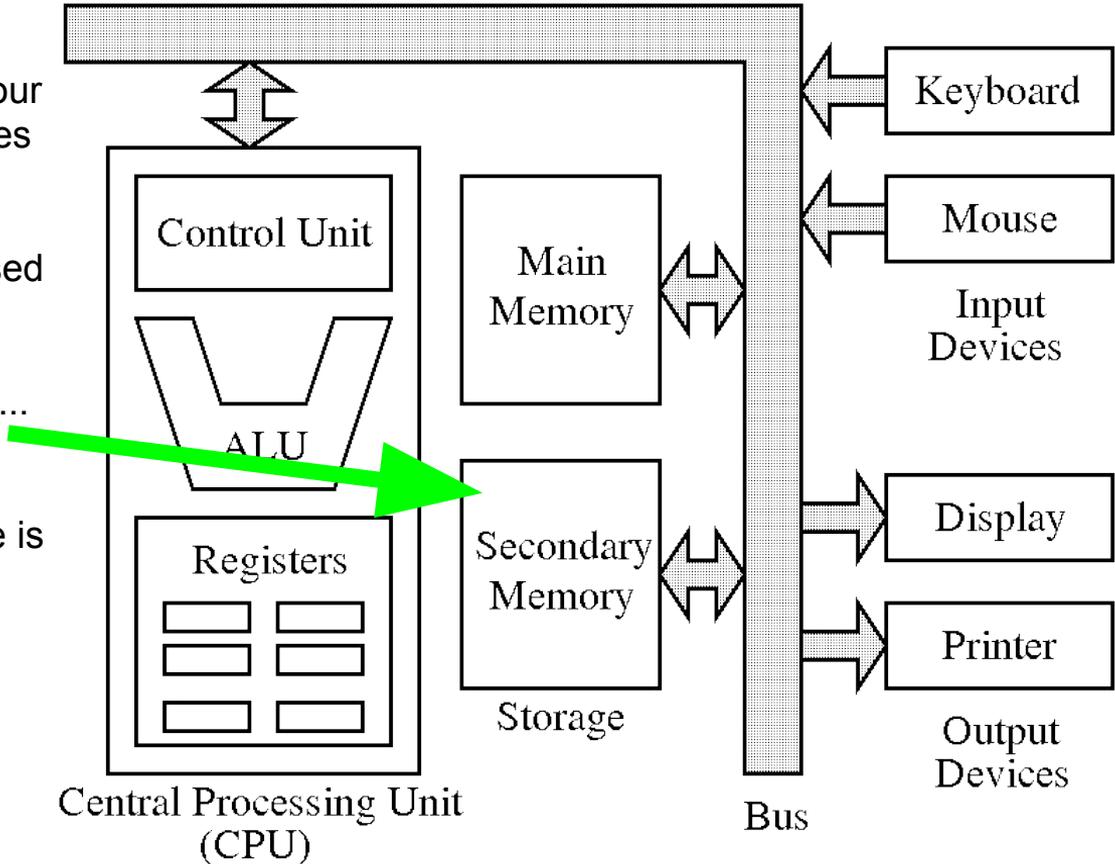
# (Simplified) View of a Computer

**Secondary Storage:** Ex: Hard Disk

Purpose: Long-term storage of data. Your pictures, movies, and PIC 10A code lives here! Nonvolatile.

Fun fact: As a rule of thumb, a disk-based hard drive (internal and external) has a life expectancy of ~5 years. So, if your laptop/drive is about that old... buy a new drive!

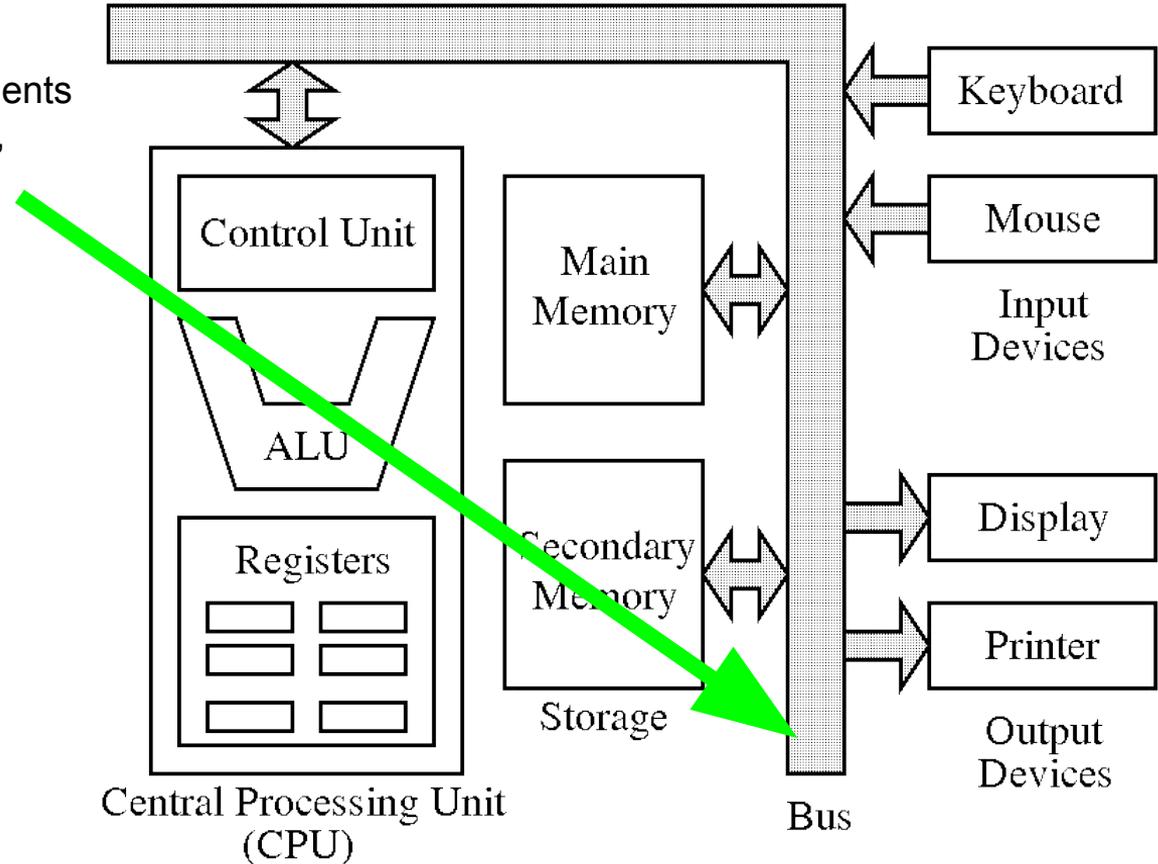
Regular backups to a (not too old) drive is good practice too.



# (Simplified) View of a Computer

## Bus:

Purpose: Connects different components together. Allows CPU to talk to RAM, disk, keyboard, mouse, monitor, etc.



# Question from Wednesday's Lecture (Jan. 6)

**Q: Complete the analogy: a desktop work surface is to its storage drawers as ...**

- (A) The CPU is to the bus
- (B) An algorithm is to a program
- (C) Read only memory is to secondary storage
- (D) Secondary storage is to read only memory
- (E) Random access memory is to secondary storage
- (F) Secondary storage is to random access memory

**Answer: E**

# Explanation: Volatile vs Nonvolatile

- "Desktop work surface stores stuff **temporarily**, but storage drawers store things more **permanently**."
- RAM is **volatile**
  - When you power down your computer, anything living in RAM is lost.
- Secondary storage is **nonvolatile**
  - Hard drive
  - Photos on hard drive remain even when computer is shut down.

# Alternate Explanation: Access Speed

- "I can access things on the work surface **quickly** (RAM), but getting things from the storage drawers takes **time/effort** (Secondary Storage)."
- CPU can access RAM fairly quickly
- Accessing hard drive is slow!
  - Have to wait for magnetic disk to spin to correct location, wait, etc.

# What is a compiler?

- **Compiler:** A program that turns **source code** into **(binary) machine code**
  - This machine code is then turned into an executable by a **linker**
    - Visual Studio 2013, Xcode do all of this for you behind the scenes
- **Source code:** What programmers write. Example:

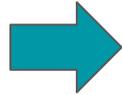
```
#include <iostream>
using namespace std;
int main() {
    cout << "Hi!" << endl;
    return 0;
}
```

- **Machine code:** Code that your actual physical CPU understands.
- **Executable:** Something I can run (ie double click on) to do something.  
Example: to open Firefox, I will run the Firefox executable.

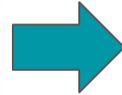
# What is a compiler?

## My Source Code

```
#include <iostream>
using namespace std;
int main() {
    cout << "Hi!" << endl;
    return 0;
}
```



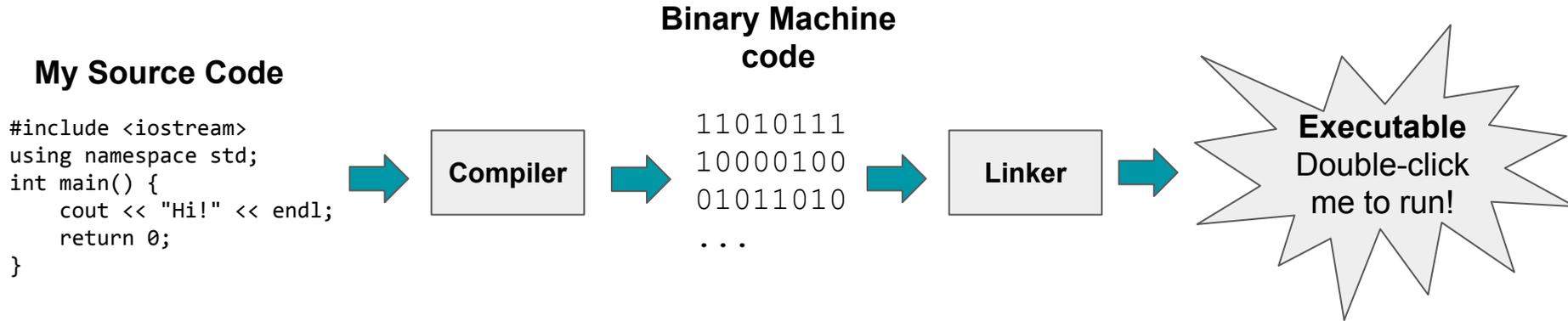
**Compiler, Linker**  
(ie VS 2013, gcc, etc.)



**Executable**  
Double-click  
me to run!

\* Note: the word "compiler" is also often used to refer to a program that turns source code into assembly code. Then, an **assembler** turns assembly code into machine code, and a **linker** turns the machine code into an executable.

# What is a compiler? (More details)

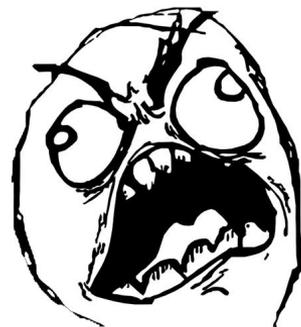


\* Note: the word "compiler" is also often used to refer to a program that turns source code into assembly code. Then, an **assembler** turns assembly code into machine code, and a **linker** turns the machine code into an executable.

# Compiler Ecosystem

- There are several popular C++ compilers in use today
  - gcc, LLVM Clang, Microsoft Visual C++, etc.
- For the most part, these compilers are compatible
  - ie my C++ program will produce the same results/behavior if I change compilers
- However, there are some disagreements between compilers
  - Ex: gcc supports some recent C++ feature, but Microsoft Visual C++ does not.
- For this class we only use Visual Studio 2013
  - Want to avoid compiler issues, an enormous headache!

**This scenario is not uncommon:** *"I ran my program using compiler X, and it worked fine. But when I switched computers and tried to run it using compiler Y, the program crashed!!!"*



# Demo: Visual Studio 2013

- Let's compile something together!