

Welcome to “Solving Problems with Computers I”

CS 16: Solving Problems with Computers I
Lecture #1

Ziad Matni
Dept. of Computer Science, UCSB

A Word About Registration for CS16 Lecture and Lab

FOR THOSE OF YOU NOT YET REGISTERED:

- The lecture and the lab selections are separate items
- When you register for the lecture, you MUST also register for one of the labs
 - They're quickly filling up, so not all of them will be available.
- Please go through the official registration process
 - I will not consider adding more students until all spots are filled up with the registration process.

Your Instructor

Your instructor: **Ziad Matni** (*zee-ahd mat-knee*)

Email: ***zmatni@cs.ucsb.edu***

(please put **CS16** at the start of the subject header)

My office hours: Mon & Wed, **2:00 PM – 3:30 PM**, at **SMSS 4409**

Your TAs

TA NAME	LAB SECTION	OFFICE HOURS
Harshitha Chidananda Murthy	M 8am, 9am	Friday 1pm – 3pm
Magzhan Zholbaryssov	M 8am, 11am	Tuesday 3pm – 5pm
Lin Chai	M 9am, 10am	Thursday 9am – 11am
Hongyuan You	M 10am, 1pm	Friday 4pm – 6pm
Dasha Rudneva	M 11am, 12pm	Thursday 4pm – 6pm
Hemanth Aditya	M 12pm, 1pm	Wednesday 3pm – 5pm

All labs will take place in **PHELPS 3525**

All TA office hours will take place in **TRAILER 936**



TRAILER 936



You!

With a show of hands, tell me... how many of you...

- A. Are Freshmen? Sophomores? Juniors? Seniors?
- B. Are CS majors? Other?
- C. Have programmed before in something other than C++?
- D. Have programmed before “just for fun”?
- E. Have programmed before “for work or school”?
- F. Have used a Linux or UNIX system before?

- G. Have (or don’t have) an iClicker?

This Class

- An **intermediate** (not a beginner's) class in computer science
 - You WILL need to have taken a beginner's class somewhere
- Covers the **basic building blocks for solving problems** using computers, in general, and using **C++ programming** specifically
 - **Why C++?**
- By the end of the course, you will be able to:
 - Intelligently take apart complex problems into more manageable parts
 - Come up with a method to solve the problem
 - Translate that into creating C++ programs that solve the problem
 - Apply Linux tools to create, submit, compile and run C++ programs
 - **Go on to take other exciting classes in programming!!!! OMG!!!**

Why Are We Using C++ in this Course?

- *C++ is one of the most widely used and in-demand programming languages*
 - Allows you to create programs to do **almost anything you need to do**
 - Dominant language for **large-scale commercial applications**
 - Used to program **all sorts of applications** in computers
 - Especially used in systems that need **raw compute speed**
 - For a list of commercial applications written in C++, see
<http://www.stroustrup.com/applications.html>
- If you can learn C++, you can more easily learn (or even teach yourself) most other popular programming languages
 - Like Python, Java, Ruby, etc...
- It looks great on your resume!
 - Actually, it's a must on any "decent" CS major's resume...

How Is This Class Taught?

- Every class has a **lecture based on the readings**
YOU MUST DO THE READINGS BEFORE CLASS!!!
- The readings are from the textbook
YOU MUST DO THE READINGS BEFORE CLASS!!!
- You are all registered to take a lab on Mondays
YOU MUST READ YOUR LAB ASSIGNMENT BEFORE YOU GO TO LAB!!!
- You have to do a lot of (kinda short) homeworks and lab assignments
PRACTICE MAKES PERFECT!!!

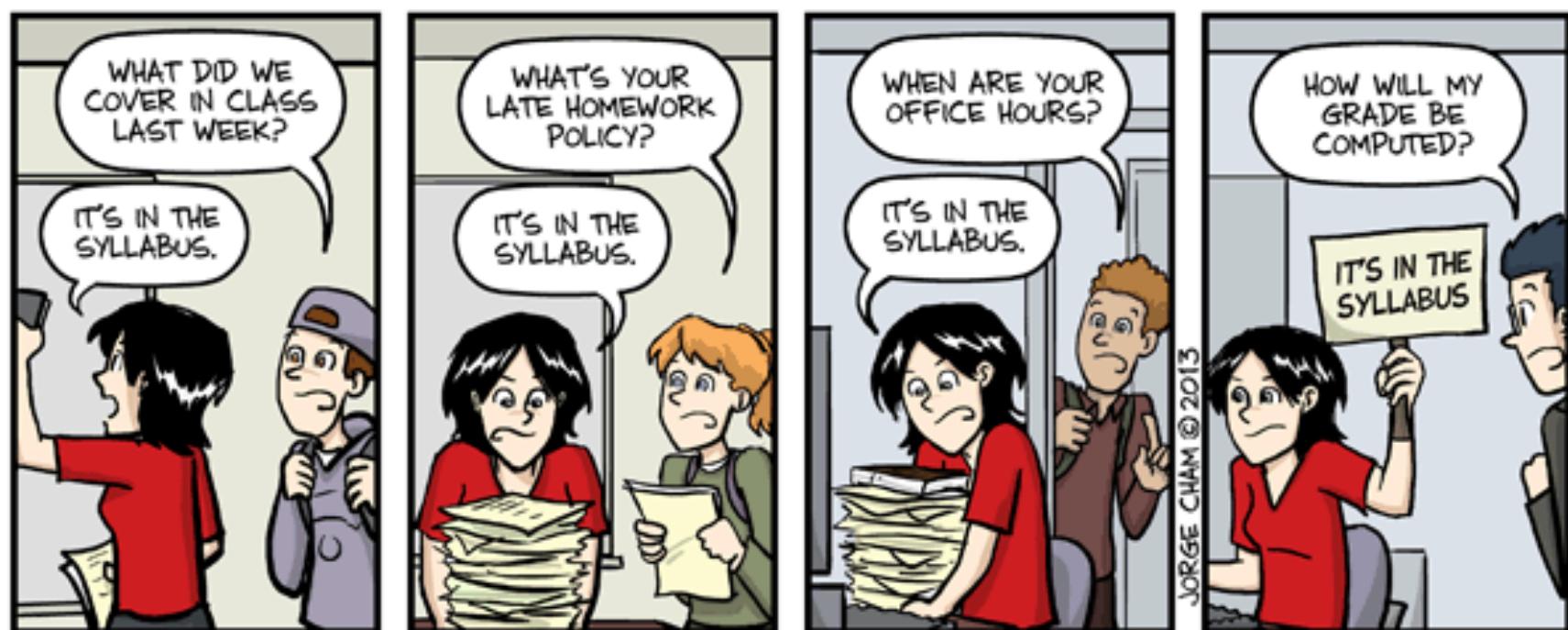


There's also actual “work” to do...

- 16 Homeworks
- 10 Lab Assignments
- 2 Midterm Exams
- 1 Final Exam

... and a partridge in a pear tree...

Just in Case...



IT'S IN THE SYLLABUS

This message brought to you by every instructor that ever lived.

WWW.PHDCOMICS.COM

Matni, CS16, Fa16

So... let's take a look at that syllabus...

Electronic version found at:

<https://ucsb-cs16-f16.github.io/info/syllabus/>

What YOU have to do before Monday

FOR THE LAB:

- Log into **Piazza** and have a look around
 - This is where you can pose questions and seek answers from fellow students, as well as the TAs and the prof.
 - Sign up for this class' page. Go to:
<https://piazza.com/ucsb/fall2016/cs16>
- Go to the class Github site and have a look around
 - This is where the prof. will put up all class information, including lecture slides, homework, lab assignments
 - Go to: <https://ucsb-cs16-f16.github.io/>
- Read the lab assignment (lab01) before you go into your lab to be better prepared

What YOU have to do before Tuesday

FOR LECTURE:

- On the class Github site (<https://ucsb-cs16-f16.github.io/>):
 1. Click on your first homework assignment (h01)
 2. Print it
 3. Do the homework in pen or pencil
 4. Bring the hardcopy of the homework to class with you on Tuesday and hand it in

A Refresher Lecture on Computers

What is this “computer” you speak of?

... and how can it help me “solve problems”???

Let’s define a “computer”

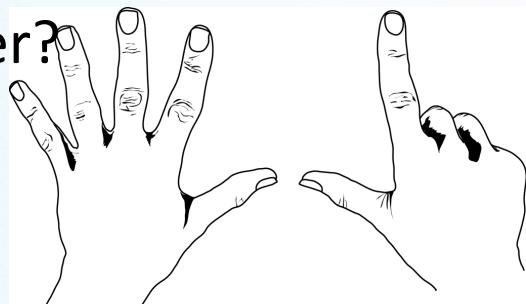
- Computer (n.): a computing device
- A device **that can be instructed** to carry out an **arbitrary** set of **arithmetic or logical operations** automatically

Computers = Computing Devices

Compute

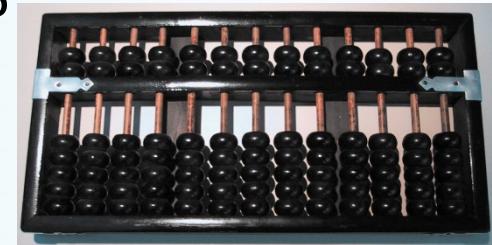
(v) To make sense of ; to **calculate** or reckon

- What was the first computing device ever?



Invented around when humans fell out of the trees

- What about for counting bigger numbers?
or more than one thing at once?



Abacus →

Invented about 5000 years ago

Matni, CS16, Fa16

TOOLS!

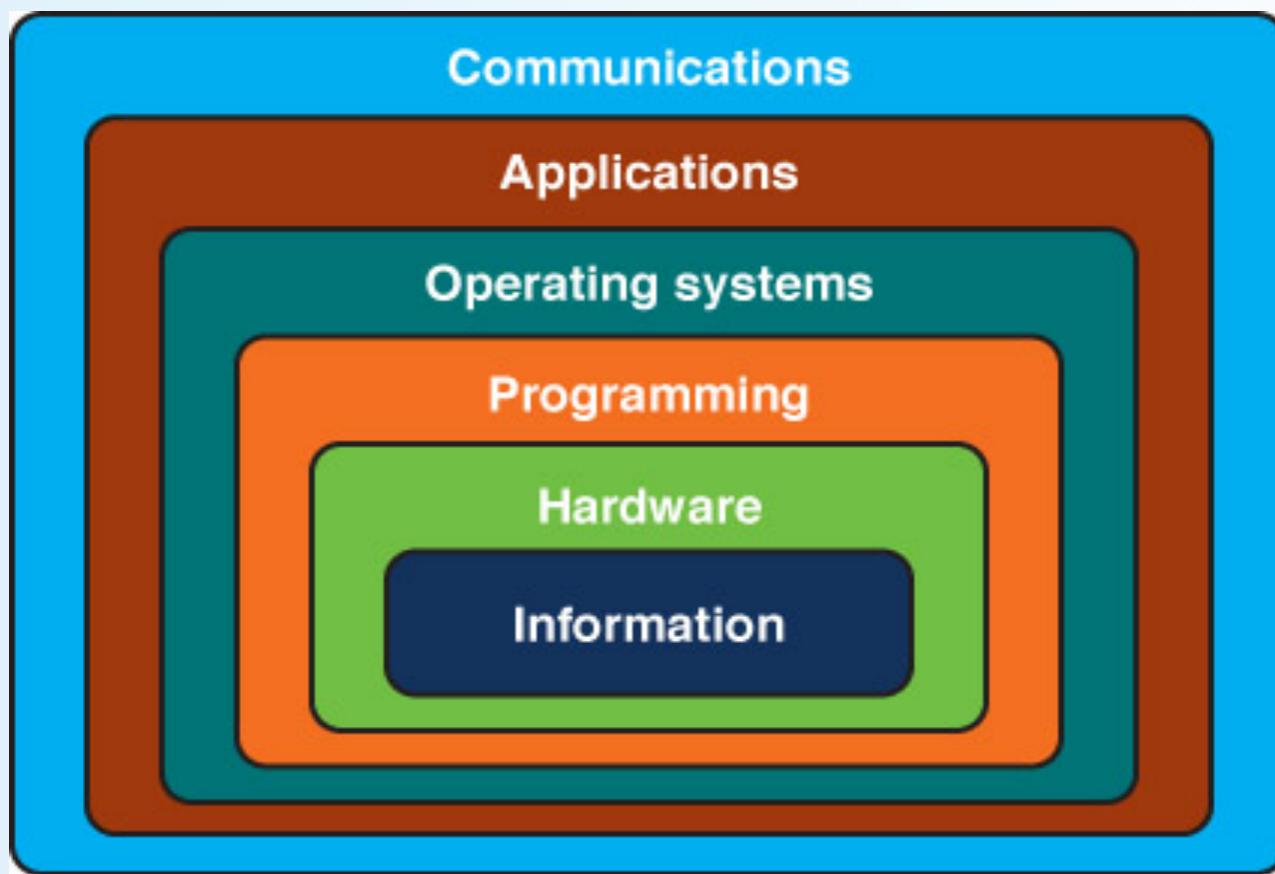
9/22/16

Images from Wikimedia.org

18

Layers of a Computing System

From *Computer Science Illuminated* by Dale and Lewis



The Concept of Abstraction

Abstraction

(n) A mental model that *removes complex details*



Images from wikimedia.org, azquotes.com

Computer Science is a science of abstraction -creating the right model for a problem and devising the appropriate mechanizable techniques to solve it.

— Alfred Aho —

Abstraction vs. All the Gory Details



Do you need to know this?
← ← ←

To know how to do this?
↓ ↓ ↓



Computer Systems

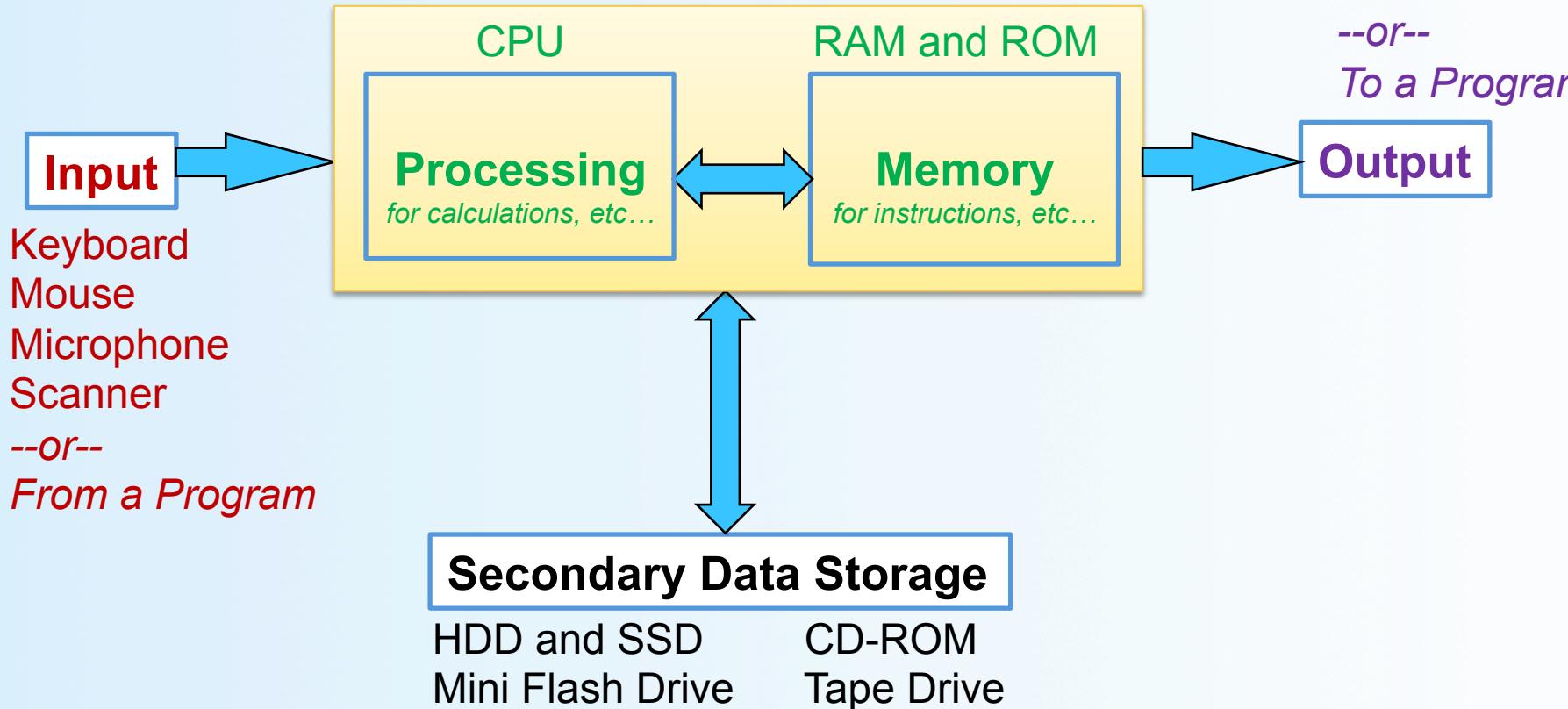
- Hardware
 - The physical
 - CPU and Memory ICs
 - Printed circuit boards
 - Plastic housing, cables, etc...
- Software
 - The instructions and the data
 - Programs and applications
 - Operating systems

Computer Architecture Acronyms

- *CPU* • *Central Processing Unit*
- *RAM* • *Random-Access Memory*
- *ROM* • *Read-Only Memory*
- *HDD* • *Hard Disk Drive*
- *SSD* • *Solid State Drive*
- *CD-ROM* • *Compact Disk – Read-Only Memory*
- *OS* • *Operating System*

A Map of Computer Components (Modern Computer Architecture)

von Neumann Architecture



Keyboard
Mouse
Microphone
Scanner
--or--
From a Program

Display screen
Speakers
Printer
--or--
To a Program

*CPU = Central Processing Unit
RAM = Random-Access Memory
HDD = Hard Disk Drive
SSD = Solid State Drive
CD-ROM = Compact Disk – Read-Only Memory
OS = Operating System*

5 Main Components to Computers

- Inputs
- Outputs
- Processor
- Main memory
 - Usually inside the computer, volatile
- Secondary memory
 - More permanent memory for mass storage of data

Computer Memory

- Usually organized in two parts:
 - Address
 - Where can I find my data?
 - Data (payload)
 - What is my data?
- The smallest representation of the data
 - A binary *bit* (“0”s and “1”s)
 - A common collection of bits is a byte (8 bits = 1 byte)
 - **Can one store *any* type of information building-block (like a number, or a letter) in 1 byte?**

What is the Most Basic Form of Computer Language?

- Binary *a.k.a* Base-2
- Expressing data AND instructions in either “1” or “0”
 - So,

“01010101 01000011 01010011 01000010 00100001 00100001”

could mean an *instruction* to “calculate 2 + 3”

Or it could mean a *number* (856783663333)

Or it could mean a *string of 6 characters* (“UCSB!!”)

</LECTURE>