

# Welcome to CSCI 10: Intro to Computer Science

with Michael J Bannister

Introductions,  
Pre-history of Computing  
and Number Systems

CSCI 10 - Santa Clara University - Fall 2016  
Michael J. Bannister

## Introductions



# What is this course?

In this course you will learn:

- algorithmic problem-solving skills
- to write simple programs solving real problems
- about the impact computing has on society

(syllabus/webpage)



# Your responsibilities

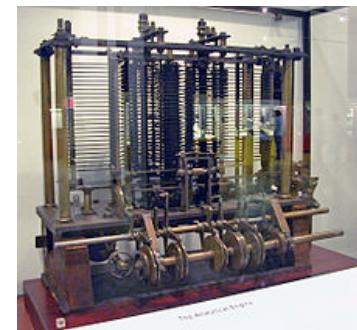
- Come to every class and lab meeting
- Do not use laptops during lecture unless told to do so
- You **must** bring a laptop to lab
- Spend two hours outside of class for every hour in class
- Ask questions right away when confused!

Any questions about the course?

# Pre-history of computing

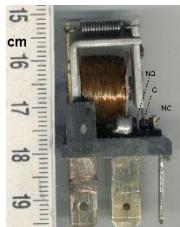
## Analytic Engine

- Designed by Charles Babbage in 1837
- First proposed general-purpose computer
- Essentially the same design as a modern computer
- Too complicated to build at the time



# Digital Circuits

- In 1705 Gottfried Wilhelm Leibniz showed, using binary numbers, that the rules of arithmetic and boolean logic could be combined
- In 1937 Claude Shannon showed that electrical circuits were capable of expressing boolean logic

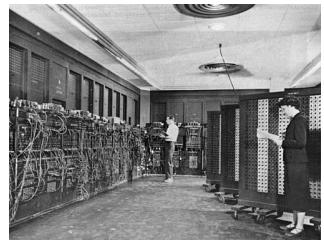
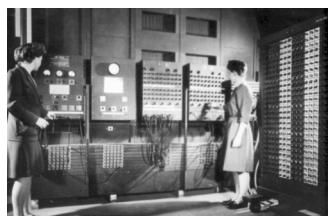


Combined we have the 1st step to modern computers!

## History

### ENIAC

- Univ. Pennsylvania (1935-45)
- 1st Electric General Purpose Computer
- 5k math ops per second
- Total weight is 30 tons
- 1st program: calculation for design of the hydrogen bomb

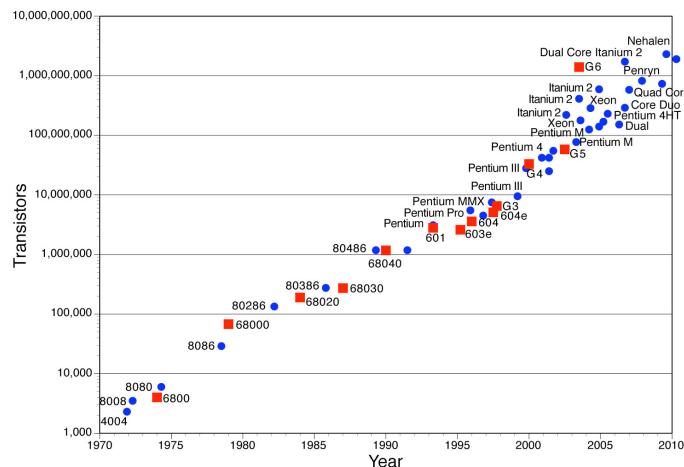


### Transistors (1950s)

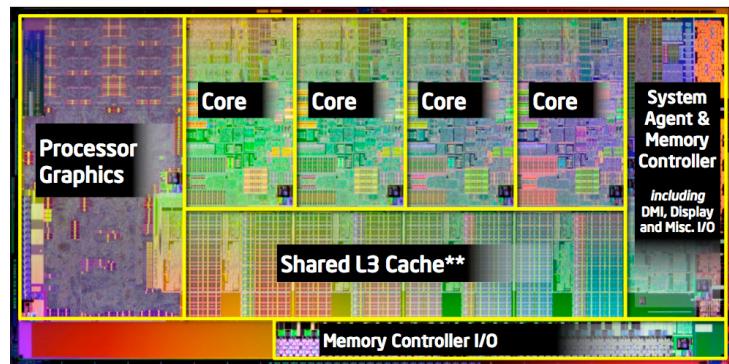
- Smaller, faster, lower power and more reliable than vacuum tubes
- Typical CPU 1-10 billion transistors
- Can be as small as 14nm; nm = a billionth of a meter!
- Developed by Shockley using silicon in Mountain View



## Moore's Law



## Modern "Chip" Design



## Hardware



## Central Processing Unit (CPU)



CPU



Cooler

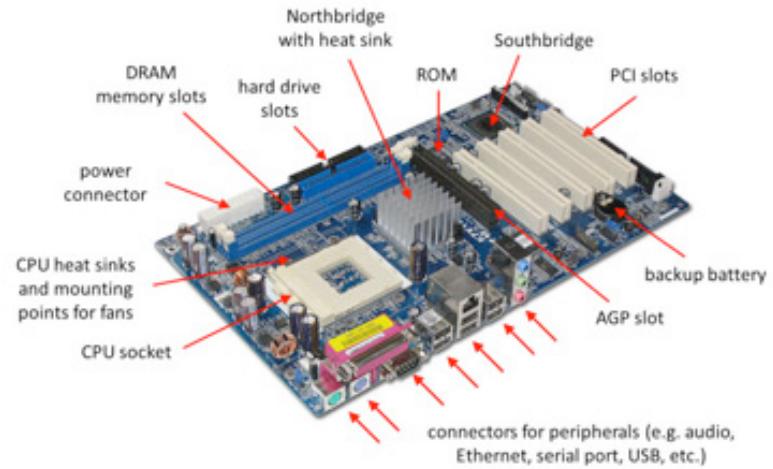
## Random-access memory (RAM)



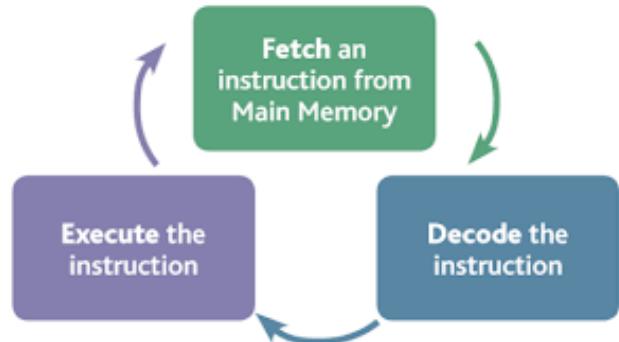
## Hard Disk & Solid State Drives



## Motherboard & Components



## Fetch-decode-execute cycle



## Software

## Operating Systems



## Programming

### Program:

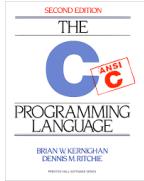
A set of instructions to be executed by the computer

Instructions consist of one or more 64 bit integer

### Programming language:

A formal languages convertible to CPU instructions, typically with emphasis on human readability

# Programming Languages



Assembly

