

# Welcome to CS 101!

## Introduction to Programming

CS101 Lecture #1

2016-09-26

<https://relate.cs.illinois.edu/course/zuics101fa16/>

# Grading

---

20%	Homework
25%	Labs
10%	Lecture Participation
20%	Midterms
25%	Final Exam

---

# Required Supplies

CodeLab account

Instructions in hw01

# Homework Policies

No late homework submissions.

# Homework Policies

No late homework submissions.

All machine-generated grades are final.

# Homework Policies

No late homework submissions.

All machine-generated grades are final.

Late registrants should keep up with work.

Corollary: No extensions or exceptions for late registration.

# Homework Policies

No late homework submissions.

All machine-generated grades are final.

Late registrants should keep up with work.

Corollary: No extensions or exceptions for late registration.

Get help at Blackboard forum.

Be civil to staff and peers.

All posts containing solutions should be marked as private.



Lab #1 this Friday!

# Modern calculation



David Hilbert

Alan Turing

Kurt Gödel

Alonzo Church

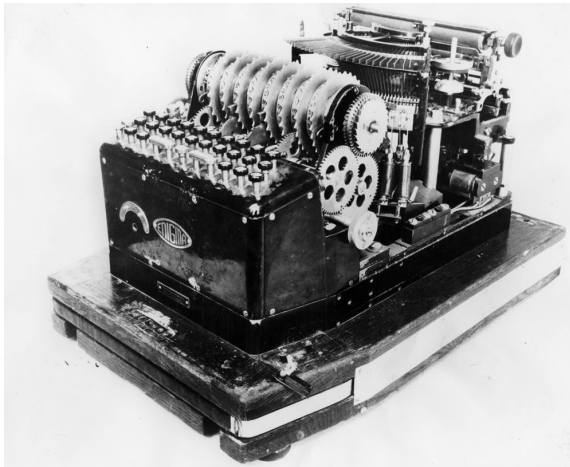
[https://en.wikipedia.org/wiki/Church%E2%80%93Turing\\_thesis](https://en.wikipedia.org/wiki/Church%E2%80%93Turing_thesis)

<https://www.bigquestionsonline.com/2013/04/30/>

[what-did-turing-establish-about-limits-computers-nature-mathematics/](#)

[http://www.alanturing.net/turing\\_archive/pages/reference%20articles/Bio%20of%20Alan%20Turing.html](http://www.alanturing.net/turing_archive/pages/reference%20articles/Bio%20of%20Alan%20Turing.html)

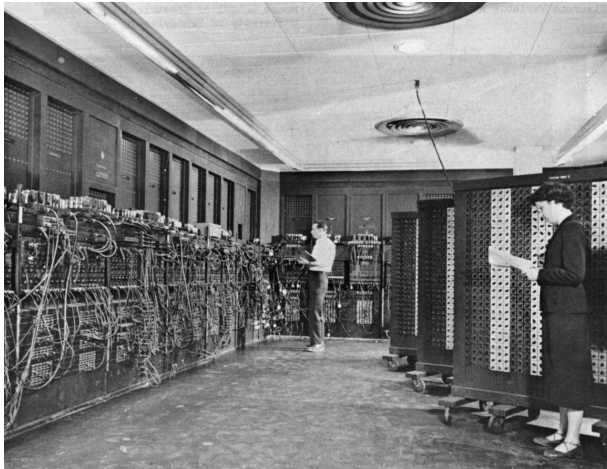
# Modern calculation



[https://en.wikipedia.org/wiki/Enigma\\_machine](https://en.wikipedia.org/wiki/Enigma_machine)

[https://en.wikipedia.org/wiki/Cryptanalysis\\_of\\_the\\_Enigma](https://en.wikipedia.org/wiki/Cryptanalysis_of_the_Enigma)

# Modern calculation



<https://en.wikipedia.org/wiki/ENIAC>

# Modern calculation



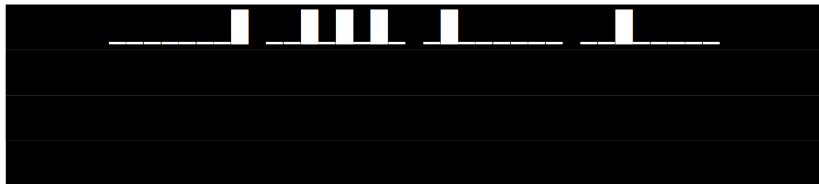
<https://en.wikipedia.org/wiki/ILLIAC>

# Algorithms

# Algorithms

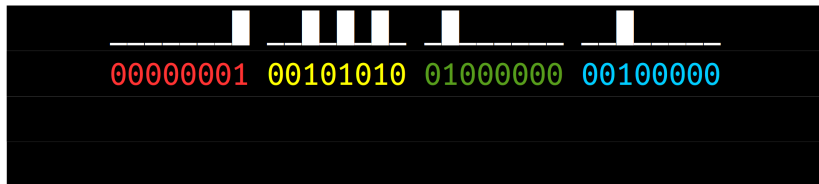


# Computing

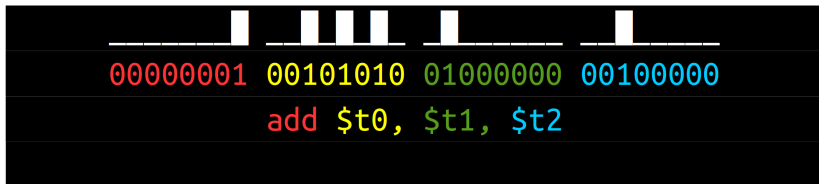




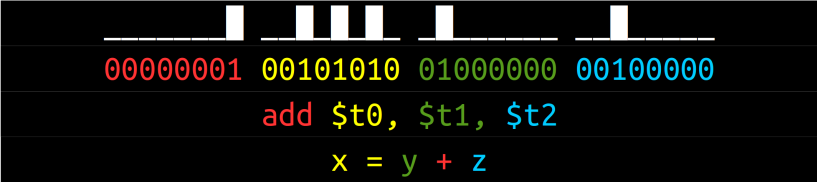
# Computing



# Computing



# Computing



A diagram of the MIPS instruction format is shown at the top of the slide. It consists of a horizontal bar divided into four sections by vertical lines. Above each section are several small white rectangles representing bits. The sections are: 1. A red section containing the binary value 00000001. 2. A yellow section containing the binary value 00101010. 3. A green section containing the binary value 01000000. 4. A cyan section containing the binary value 00100000.

00000001 00101010 01000000 00100000

add \$t0, \$t1, \$t2

x = y + z

# Algorithms



# Algorithms

`depth * area = volume`

# Algorithms

$\text{depth} * \text{area} = \text{volume}$

$\text{volume of rain} / \text{volume per raindrop}$   
 $= \text{number of raindrops}$

# Algorithms

$\text{depth} * \text{area} = \text{volume}$

$\text{volume of rain} / \text{volume per raindrop}$   
 $= \text{number of raindrops}$

$\text{volume\_rain} = \text{area} * \text{depth}$

# Algorithms

`depth * area = volume`

`volume of rain / volume per raindrop  
= number of raindrops`

`volume_rain = area * depth`

`n_raindrops = volume_rain / volume_raindrop`



# What is a program?

# What is a program?

A set of instructions a computer executes to achieve a goal.

# What is data?

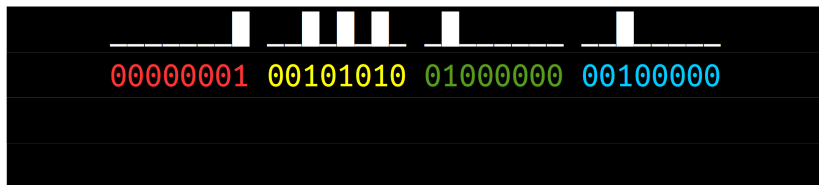
# What is data?

Information stored in a computer.

# What is data?

Information stored in a computer.

All data is stored in binary.



# What is data?

Binary data must be interpreted:  
instruction

# What is data?

Binary data must be interpreted:  
instruction  
value (number, character)

# What is data?

Binary data must be interpreted:

- instruction

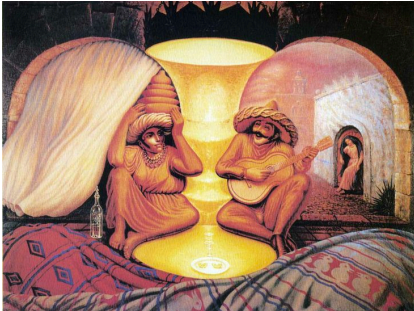
- value (number, character)

- memory location



# What is data?

Binary data must be interpreted:  
instruction  
value (number, character)  
memory location



# What is a program?

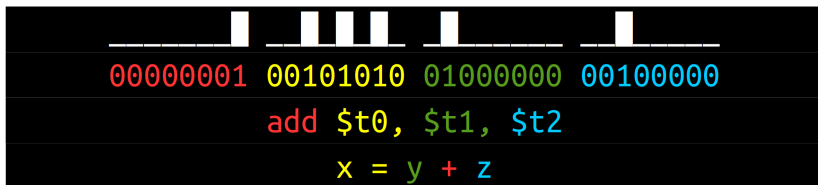
# What is a program?

Programs are data!

# What is a program?

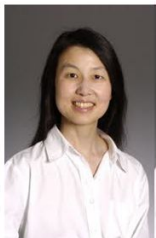
Programs are data!

Instructions are encoded in binary.



# Computational Thinking

## Computational Thinking



Computational thinking is a **fundamental skill for everyone**, not just for computer scientists. To reading, writing, and arithmetic, **we should add computational thinking to every child's analytical ability**. Just as the printing press facilitated the spread of the three Rs, what is appropriately incestuous about this vision is that computing and computers facilitate the spread of computational thinking.

# Engineer Joke: Engineering Thinking

Four engineers traveling in a car and the car breaks down ...

**Mechanical engineer:** "Sounds to me as if the pistons have seized. We'll have to strip down the engine before we can get the car working again"

**Chemical engineer:** "It sounded to me as if the fuel might be contaminated. I think we should clear out the fuel system."

**Electrical engineer:** "I thought it might be an grounding problem or maybe a faulty plug lead."

**Software/computer engineer:** "Ummm perhaps if we all get out of the car and get back in again?"

## Reality in Industry: Engineering Thinking

Researchers working on a robot arm for assembling pens.  
They face challenges, e.g., lacking sufficient accuracy.  
Any directions for solving the problem?



# Reminders

<https://relate.cs.illinois.edu/course/zuics101fa16/>