

What is Computation?

A High-Level Look at Computer Science & Programming



A Poem about Software

“... The world seems to be a big organism, with the birds as components, like big, feathered cells. Why they are flying around in 28 degree weather, seemingly just enjoying the morning, I don't know.

Of course their feathers are made up of cells themselves. And the cells are made up of a nucleus, DNA, RNA, and the whole replication machine that is the living cell. The purpose of that cell is a component of the bird itself.

A feather cell, a stomach cell, a cell in the bird's eye, looking at the red orange glow as the sun rises clear in the cold east. The feather, the bird, the world, is all a single thing with hierachal components making it all work.

Software and computing is the same.”

- Excerpt from a LinkedIn Post by Brantley Coile, Systems Programmer



A Few Big Questions

- ▶ What is a computer?
- ▶ What is computation?
- ▶ What is programming?
- ▶ How do we solve problems with computers?

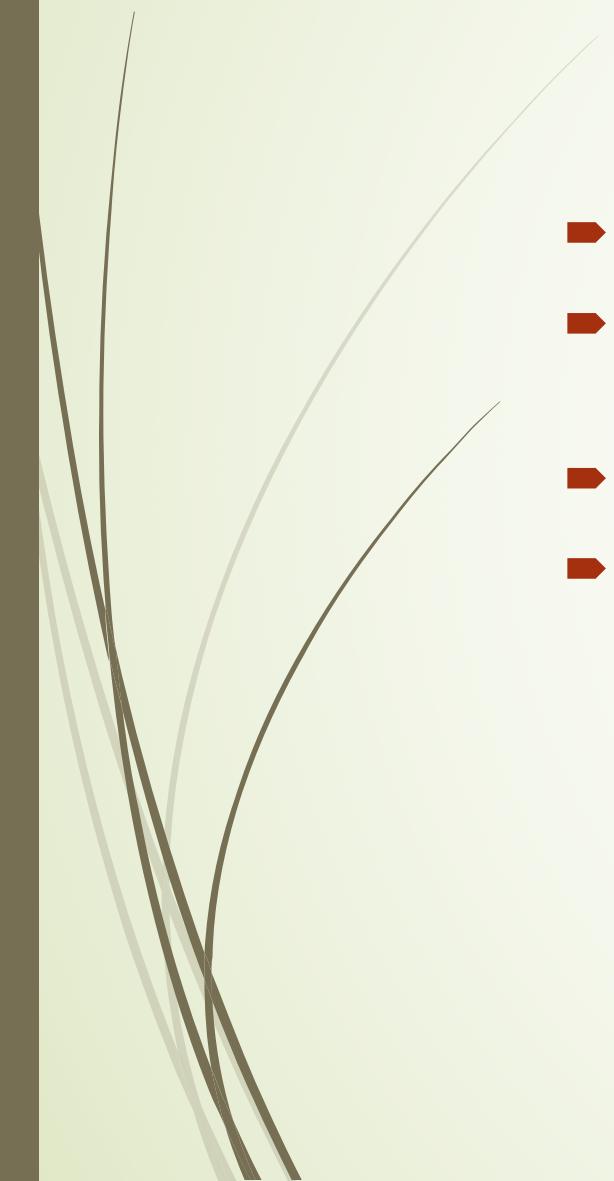
All Questions Answered (Kind of)

- ▶ Computer Science delivers 3 Big Insights. (Rapaport)
 - ▶ All the information we need can be represented as binary data: 1 or 0. **[Representational]**
 - ▶ Only a handful of simple operations are needed to compute anything with a reel of binary data: move left/right, print 0/1, erase. **[Processing]**
 - ▶ Only a handful of grammar rules are needed to arrange those basic operations as programs: sequence, selection, repetition. **[Structural]**



The Representational Insight

- ▶ 1948 - Claude Shannon starts Information Theory with “A Mathematical Theory of Communication”
- ▶ All of the information we need can be represented as 1s and 0s.
- ▶ In other words, all information is a series of distinctions between one thing and another.
- ▶ "Up until that time, everyone thought that communication was involved in trying to find ways of communicating written language, spoken language, pictures, video, and all of these different things— that all of these would require different ways of communicating. Claude said no, you can turn all of them into binary digits. And then you can find ways of communicating the binary digits." —Robert Gallager, quoted in (Soni and Goodman, 2017)



The Processing Insight

- ▶ 1936 - Church Turing Computability Thesis
- ▶ Turing's Model is an Abstract Tape Machine that Can Compute Anything.
- ▶ The Instructions Available are: Move Left/Right, Print 0/1, and Erase.
- ▶ Everything that can compute anything is Turing Complete:
 - ▶ Javascript
 - ▶ Python
 - ▶ C
 - ▶ Haskell
 - ▶ Java
 - ▶ Assembly Language
 - ▶ ... are all equivalent to Turing Machines!

Lego Turing Machine in Action.

► <https://www.youtube.com/watch?v=FTSAiF9AHN4>



The Structural Insight

- ▶ 1966 Boem and Jacopini
- ▶ Created in reaction to Dijkstra's "Go-To Considered Harmful"
- ▶ Only 3 basic grammar tools are needed to organize the Turing Machine operations into programs:
 - ▶ Sequence – Basic Control Flow
 - ▶ Selection – If/Else Branches in Control Flow
 - ▶ Repetition – Loops in Control Flow



And from these atoms... the universe!

- ▶ We build in hierachal layers of abstract machines to create a rich world of digital experience.
 - ▶ How might I construct an image on a screen using binary?
- 

Building Up A Computational World

- ▶ <https://www.youtube.com/watch?v=xnyFYiK2rSY&list=PLzdnOPI1iJNcsRwJhvksEo1tJqjlqWbN-&index=6>