L00: Introduction to Theory of Computation (Pre Lecture)

Dr. Neil T. Dantam

CSCI-561, Colorado School of Mines

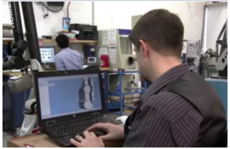
Fall 2022

What is Computer Science?

A car analogy

Mechanic

Engineer

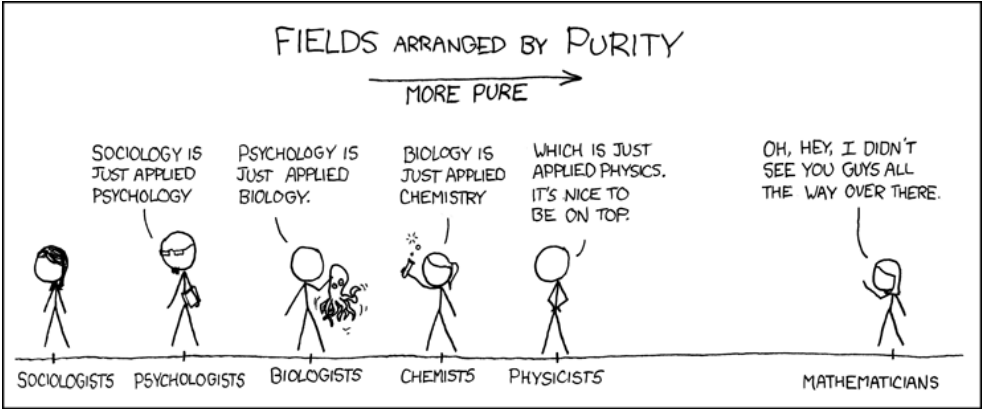


Physicist

Applied ⟷ Theoretical

Where is computer science on the theory/applied scale?

From the Internets



https://xkcd.com/435/

Laws of Physics

More Car Analogy

$$F = m\ddot{x}$$

$$\ddot{x} = \frac{F}{m}$$

Physical Laws → Capabilities of Physical Systems

Laws of Computation?

Are there "laws of physics" in computer science?

A Physical Law

$$\frac{d\mathbf{x}}{dt} = \mathbf{g}(\mathbf{x}, \mathbf{u})$$

- $\mathbf{x}$: current state
- $\mathbf{u}$: current input
- $\frac{d\mathbf{x}}{dt}$: change in state
- $\mathbf{g}$: process function

A Computational Law

$$q_{k+1} = \delta(q_k, \sigma_k)$$

- q_k: current state
- σ_k: current input
- q_{k+1}: successor state
- δ: transition function

Computational Laws → Capabilities of Computational Systems

MINES

Attitudes on Theory

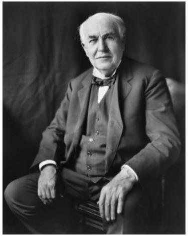
"There is nothing so practical as a good theory."
–Kurt Lewin

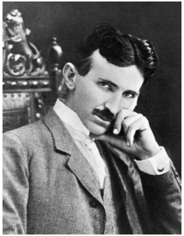
"In theory, there is no difference between theory and practice. But, in practice, there is."
–Jan L. A. van de Snepscheut

Applications of Theory

“Genius Is One Percent Inspiration, Ninety-Nine Percent Perspiration”
–Thomas Edison

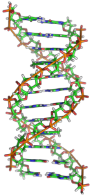
“Just a little theory and calculation would have saved [Edison] ninety percent of his labor.”
–Nikola Tesla





In practice, theory is useful.

Applications of CS Theory



Compiler Construction

Circuit Verification

DNA Matching

Industrial Control

Direct applications as wide-reaching as differential calculus

Programmer to Computer Scientist

Programmer	SW Engineer	Computer Scientist
`print("Hello world.")`	`static LIST_HEAD(klist);`	$\lambda f \,.\, (\lambda x \,.\, f\ (x\,x))\ (\lambda x \,.\, f\ (x\,x))$

Applied ⟵⟶ Theoretical

- View 0: Theory makes one a better programmer.
  - Bigger "bag of tricks."
  - Building understanding of how to represent/solve previously-unsolved problems.
- View 1: Theory takes one from "Programmer" to "Computer Scientist."
  - Going from consumer to producer of algorithms.
  - Solving previously unsolved problems.

Example Computing Problems

- Capitalize all instances of “internet” in a text document.
- Check for matching tags in an XML document.
- Find all memory leaks in a C program.

How **fast** *can we solve these problems?*

Can *we solve these problems?*

Course Questions

What is a computer?

Different models of computation

What can we compute?

Problems that are solvable/unsolvable using different models of computation.

How well can we compute?

Performance capabilities/limits for various models and problems

Outline

Course Logistics

Brief History of Computation

Common Misconceptions

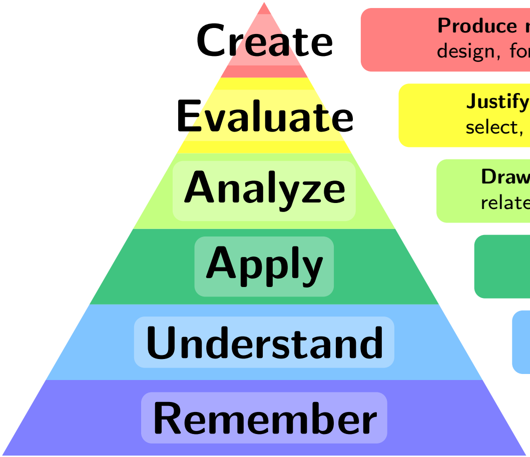
Outline

Course Logistics

Brief History of Computation

Common Misconceptions

Bloom's Taxonomy



Produce new or original work
design, formulate, investigate

Justify a position or decision
select, appraise, critique, defend

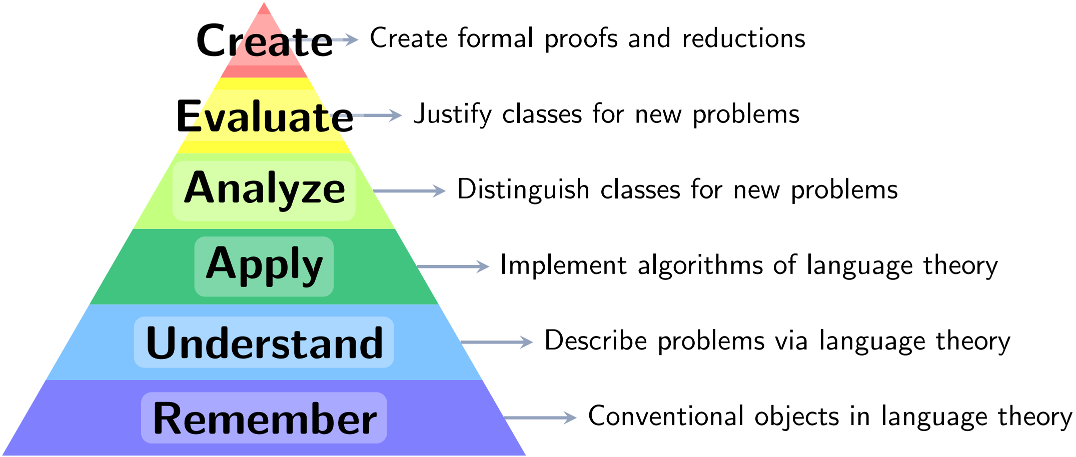
Draw connections among ideas
relate, compare, distinguish, contrast

Use information in new situations
implement, execute, solve

Explain ideas or concepts
describe, discuss, recognize, explain

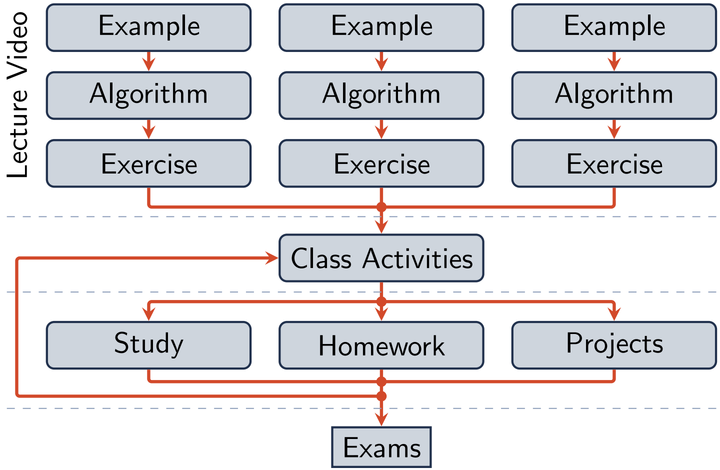
Recall facts and basic concepts
memorize, define, duplicate, repeat

Learning Outcomes

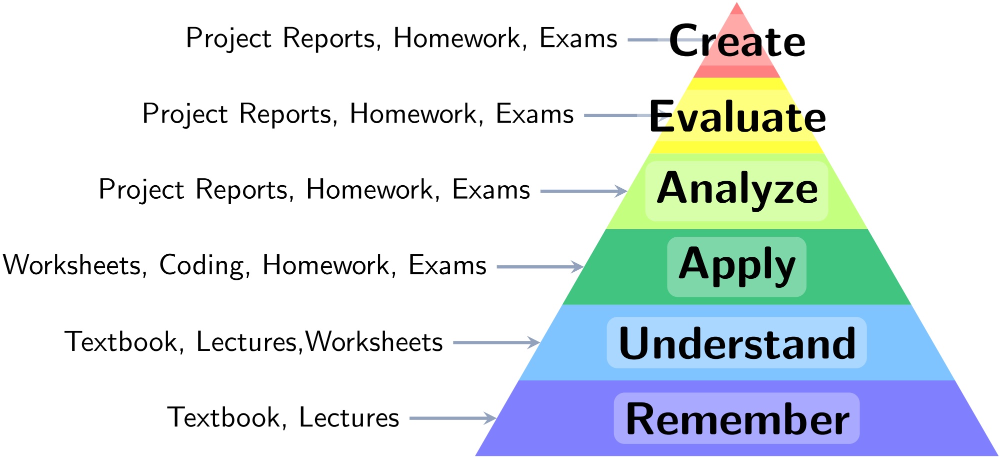


Flipped Classroom Process

Approximate, Not to Scale



Learning Activities

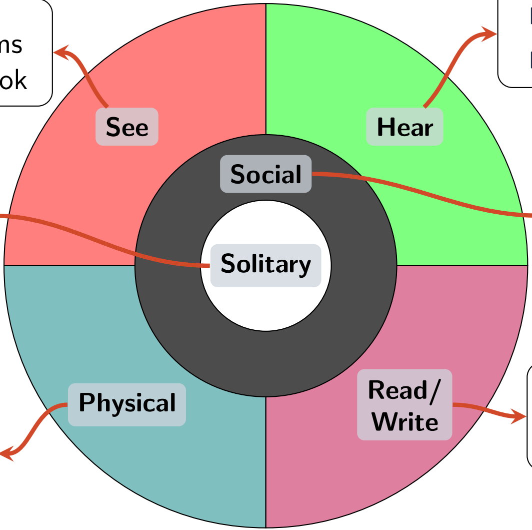


Activities by Learning Style (general)

- Look at diagrams in slides/textbook

- Listen to Lecture
- Discuss w/ classmates

- Homeworks
- Exams
- Study Alone



- Ask questions
- Discuss Exercises
- Group Projects
- Study in Groups

- Take notes

- Projects
- Read textbooks

Learning Styles (course-specific)

Verbal
- Textbook descriptions
- Lecture bullets / speech

Visual
- Diagrams in Textbook and lecture slides

Symbolic
- Equations
- Pseudocode

Expectations

- This course is a graduate level computer science course
  - You already know how to program
  - You can learn new programming languages and frameworks
  - You will spend, on average, 15 hours per week (out of class)
- This course is about the *Theoretical Foundations of Computer Science*
  - **Theoretical** and **Foundational**
  - This course is **NOT** about programming or particular applications
  - This course **IS** about the mathematics that underlie computation
- Guiding Objectives:
  - **Challenging:** go beyond your (prior) comfort zone.
  - **Fair:** doable; even and consistent evaluation.
  - **Useful:** preparation for graduate-level ("computer *scientist*") work

Syllabus

How to Succeed (or not) in this Course

Succeed	Or Not
Participate in lecture	Skip lecture
Ask questions / come to office hours	Don't ask questions
Study early and often	Cram for exams
Start projects and homeworks early	Delay starting assignments

Theory takes time to learn.

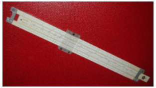
Outline

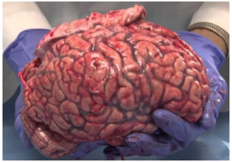
Course Logistics

Brief History of Computation

Common Misconceptions

What is a computer?





What can a computer do?

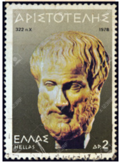
Computation vs. Computers

"Computer science is no more about computers than astronomy is about telescopes."
–Edsger W. Dijkstra

The process of reasoning vs. machines that do so

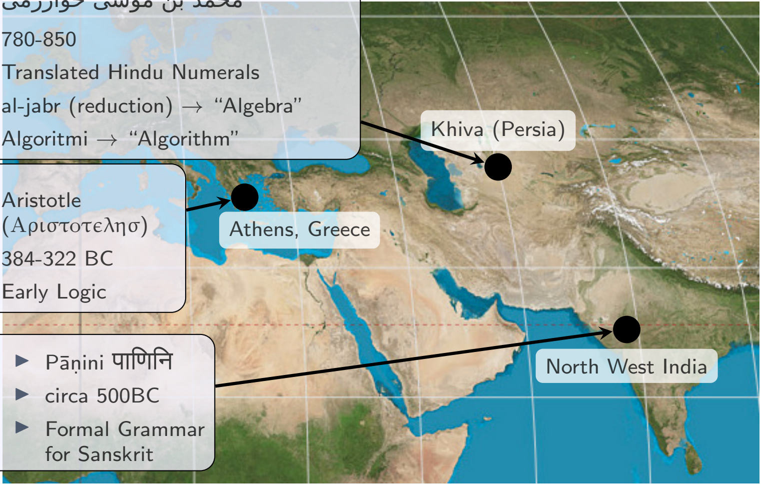
Computation in Antiquity

- Muhammad ibn Musa al-Khwarizmi
  محمد بن موسى خوارزمی
- 780-850
- Translated Hindu Numerals
- al-jabr (reduction) → "Algebra"
- Algoritmi → "Algorithm"

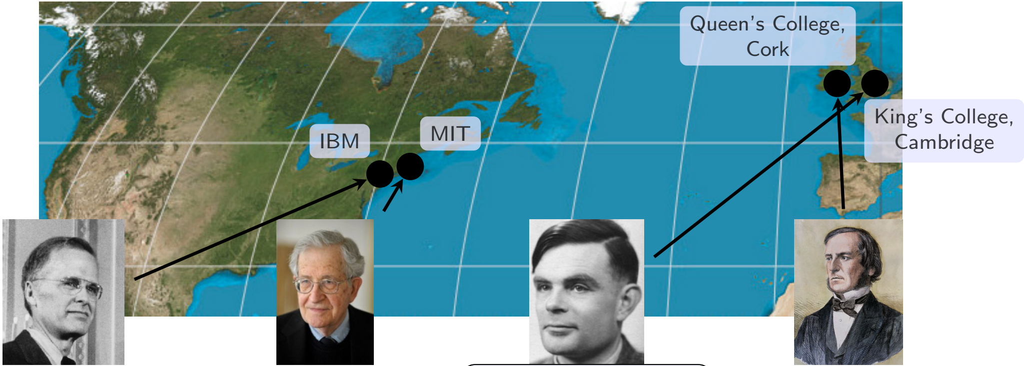


- Aristotle
  (Αριστοτελησ)
- 384-322 BC
- Early Logic

- Pāṇini पाणिनि
- circa 500BC
- Formal Grammar for Sanskrit



Modern History (abbreviated)



- John Backus
- 1924–2007
- Fortran

- Noam Chomsky
- 1928–
- Language Theory

- Alan Turing
- 1912-1954
- Theory of Computation
- Cryptography

- George Boole
- 1815-1864
- Boolean Algebra

History of Computation

We think of computing as a modern invention,
but its origins date to the earliest recorded history,
spanning cultures and civilizations.

Outline

Course Logistics

Brief History of Computation

Common Misconceptions

Attitudes on Math

- **Being good at math is about how smart you are. ~~Being good at math is about how smart you are.~~**
  - Mathematical ability is a learned skill.
  - ~~fixed mindset~~ vs. growth mindset

Attitudes on Computer Science

- **Computer science means coding.** ~~**Computer science means coding.**~~
  Astronomy is not (just) building telescopes. Engineering is not (just) turning wrenches.
  Computer science is not (just) writing code.
  Better: Computer Science is about how we model, analyze, and solve computational problems.

Attitudes on Grad School

- **Grad school is about memorizing more algorithms and software frameworks.**
  ~~**Grad school is about memorizing more algorithms and software frameworks.**~~
  Grad school is about reaching the frontiers of human knowledge and learning how to expand that frontier. It is less about memorizing facts and more about learning new ways to think.
- **I can complete course projects in a few days.**
  ~~**I can complete course projects in a few days.**~~
  Hard problems take time to internalize, understand, and solve; you need a well-rested mind to perform well. In this course, projects will be assigned multiple weeks in advance, and 95% of students require the majority of the allotted time to successfully finish.