Computational Thinking for All

Jeannette M. Wing

Corporate Vice President, Microsoft Research
Consulting Professor of Computer Science, Carnegie Mellon University

Centrality and Dimensions of Computing Panel
Workshop on the Growth of Computer Science Undergraduate Enrollments
National Academies, Washington, DC
15 August 2016

My Vision

Computational thinking will be a fundamental skill used by everyone in the world by the middle of the 21st Century.

Viewpoint | Jeannette M. Wing

J.M. Wing, "Computational Thinking," *CACM* Viewpoint, March 2006, pp. 33-35.
Paper off http://www.cs.cmu.edu/~wing/

Computational Thinking

It represents a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use.

Computational thinking builds on the power and limits of computing processes, whether they are executed by a human or by a machine. Computational methods and models give us the courage to solve problems and design systems that no one of us would be capable of tackling alone. Computational thinking confronts the riddle of machine intelligence:

cisely. Stating the difficulty of a problem accounts for the underlying power of the machine—the computing device that will run the solution. We must consider the machine's instruction set, its resource constraints, and its operating environment.

In solving a problem efficiently, we might further ask whether an approximate solution is good enough, whether we can use randomization to our advantage, and whether false positives or false negatives are allowed. Computational thinking is reformulating a seemingly difficult problem into one we

J.M. Wing, "Computational Thinking, Ten Years Later, CACM blog, March 2016.
http://cacm.acm.org/blogs/blog-cacm/201241-computational-thinking-10-years-later/fulltext

everyone, not just for computer scientists. To reading, writing, and arithmetic, we should add computational thinking to every child's analytical ability.

both the virtues and the dangers of aliasing, or giving someone or something more than one name. It is recognizing both the cost and power of indirect

Definition

Technical: Computational thinking is the thought processes involved in formulating a problem and expressing its solution(s) in such a way that a computer—human or machine—can effectively carry out.

For Today: Computational thinking = computational concepts, methods, algorithms, languages, tools, and systems.

Or if you prefer, CT = CS (exceptions will be noted).

X = Science and Engineering

Context (2005-09)

- Science 2020, Wing's Computational Thinking CACM article, NSF Cyber-enabled Discovery and Innovation (CDI) Program, Jim Gray's Fourth Paradigm

Claims

1. All science and engineering disciplines will rely on computing to make progress
2. Computing will expedite progress in all science and engineering disciplines.
3. Research in science and engineering leads to educational changes in those disciplines

Evidence

- For 1&2: NSF, NIH, DOE/OS, DARPA proposals
- For 3: Curricula requirements for degree programs in non-CS majors

X = Arts, Humanities, Social Sciences, …

2006: AT CMU, there were 24 "Computational X" courses, degree programs, or departments, where X came from every single school/college on campus. Since then the Computational Biology Department was established as a department in the School of Computer Science.

Today: Data Science (which overlaps with Computational Thinking) at universities

- Core: computer science, statistics, and operations research (optimization)
- Applications: All fields of study
- Evidence: Berkeley (DS+CS), Columbia (DSI), MIT (IDSS), Stanford, … have university-wide institutes, degree programs, courses

Future:

- Data is not going away. Volume, velocity, variety, variability, veracity of data will continue to grow due to technology for collecting and generating data.
- All fields of study have data and will rely on *computing* to discover new concepts, patterns, and relationships. *Computing* will transform the very conduct of these fields.

Computational Thinking at Carnegie Mellon (2006)

- Computational and applied mathematics
- Computational biology
- Computational chemistry
- Computational design
- Computational economics
- Computational finance
- Computational linguistics
- Computational mechanics
- Computational neuroscience
- Computational photography
- Computational physics
- Computational and statistical learning

- Algorithms, combinatorics, and optimization (joint between CS, math, business)
- Computation, organizations, and society
- Computer-aided language learning (CS and modern languages)
- Computer music
- Electrical and computer engineering
- Electronic commerce (CS and business)
- Entertainment technology (CS and drama)
- Human-computer interaction (CS, design, and psychology)
- Language technologies (CS and linguistics)
- Logic and computation (CS and philosophy)
- Pure and applied logic (CS, math, and philosophy)
- Robotics (CS, electrical and computer engineering, and mechanical engineering)

X = Professions and Sectors

Professions:

- Traditional jobs: Demand for computer scientists and IT professionals continues to outgrow supply
- New Job Titles: Data Scientist, Applied Data Scientist

Sectors

- Medicine (personalized healthcare)
- Law (LexisNexis)
- Finance (high-frequency automated trading)
- Manufacturing (robotics)
- Industrial (jet engines as a service)
- Pharmaceutical (personalized drugs)
- Automotive (self-driving cars)
- Education (MOOCs)
- Retail (e-commerce)
- Government (e-government)
- …

Technology Disrupters and Trends

Technology Categories

- Data
- Cloud
- Devices: sensors, actuators, VR/AR, big displays
- Mobility: phones, drones, people
- Decentralization: crowdsourcing, social networks, uberization (shared economy), blockchain
- End of Moore's Law
  - "smarter" in silicon, biological computing, quantum computing

Technology Areas

- Machine learning (deep learning, reinforcement learning)
- Artificial intelligence (related to ML)
  - speech, vision, NLP, personalized agents (Siri, Cortana, Google Now, chatbots)
- Search to Q&A, "knowledge" to decision-making, multi-media (text, video, maps)
  - at scale, fine-grained resolution, near real-time fidelity
- Cybersecurity: sophistication and number of threats and attacks
- Crypto made practical: homomorphic encryption, secure multi-party computation

Fundamental Difference

The fundamental difference between Computer Science and every other discipline is **software**.

Software is easy to create, change, copy, store, and disseminate. It is unlike any other natural or engineered artifact.

Systems we build in software are limited in design only by the limits of *human creativity.*

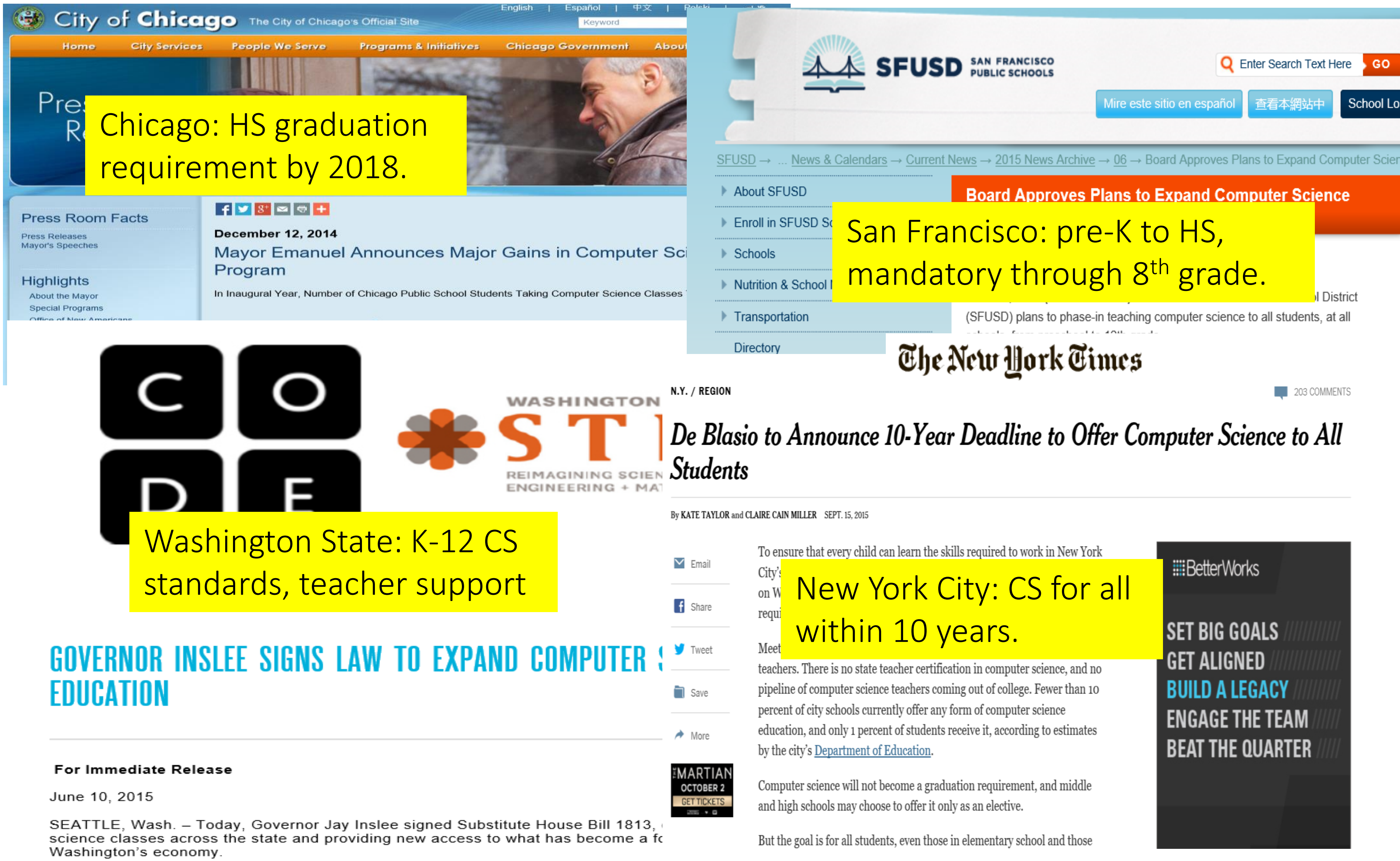
Thank you!

Computational Thinking in K-12 Education

President Obama 2016 State of the Union Address

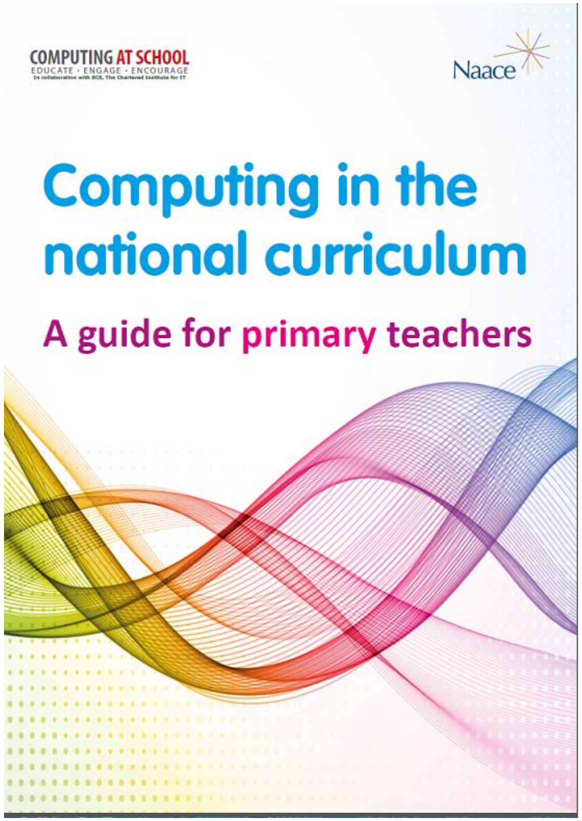
"In the coming years, we should build on that progress, by providing pre-K for all, offering every student the hands-on computer science and math classes that make them job-ready on day one.“ [Obama, January 12, 2016]

US Goal: Give Access to Computer Science to Every High School Student



UK: CT is Required at Each K-12 Level

COMPUTING AT SCHOOL
EDUCATE · ENGAGE · ENCOURAGE
In collaboration with BCS, The Chartered Institute for IT



Asia: China, Korea, Japan, Singapore, …

Computational Thinking, International

计算思维

周以真

计算思维代表着一种普遍的认识和一类普适的技能，每一个人，而不仅仅是计算机科学家，都应热心于它的学习和运用。

计算思维建立在计算过程的能力和限制之上，由人由机器执行。计算方法和模型使我们敢于去处理那些原本无

一步问：一个近似解是否就足够了，是否可以利用一下随机化，以及是否允许误报（false positive）和漏报（false negative）？计算思维

COMMUNICATIONS OF THE ACM March 2006/Vol. 49, No. 3

In *Bulletin of Specif*, December 2008

La pensée informatique

par Jeannette M. Wing

Cet article fait suite aux divers interviews que nous avons faits et qui nous invitaient à une réflexion sur les fondements de notre discipline et ses aspects philosophiques et épistémologiques. Aujourd'hui l'article de Jeannette Wing nous conduit à réfléchir sur l'utilité et l'ubiquité de la pensée informatique et ses implications, mais aussi sur l'essence même de cette pensée.

컴퓨팅적 사고

컴퓨팅적 사고는 컴퓨터 과학자뿐만이 아니라 누구나 배워서 활용할 수 있는 보편적인 사고이자 기술이다.

컴퓨팅적 사고는 사고의 주체가 컴퓨터건 사람이건 간에 전산처리의 힘과 한계에 기반해 있다. 컴퓨팅적 방법론과 모델을 통해 우리는 혼자서는 만들 수 없는 시스템을 설계하고 어려운 문제를 해결할 수 있을 거라는 자신감을 얻을 수 있다. 우리는 컴퓨팅적 사고를 통해 수수께끼와 같은 기계 지능의 난제에 도전한다. 인을 이용하는 것이 용이한지, 거짓 양성 (false positives)과 거짓 음성 (false negatives)[2]을 허락해도 될지 고민할 것이다. 우리는 컴퓨팅적 사고를 통해 축소, 내장, 변형이나 시뮬레이션과 같은 기법으로 무척이나 어려워 보이는 문제를 이미 해결 방법을 알고 있는 문제로 재구성할 수도 있다.

Computational Thinking

計算論的思考

基 般

Jeannette M. Wing（Microsoft Research and Carnegie Mellon University）
翻訳：中島秀之（公立はこだて未来大学）

［原文］Jeannette M. Wing：Computational Thinking, Communications of the ACM, Vol.49, No.3, pp.33-35（Mar. 2006）より許可を得て翻訳．

これは Wing の 2006 年のエッセイである．これが出た当時，我々日本の研究者仲間も似たような感覚を持っていたので，このエッセイを歓迎した．すぐに誰かが翻訳するものだと思っていたら，2014 年の現在に至るまでその気配はない．書いてあることが我々研究者には当たり前だった（のでわざわざ翻訳しようと思わなかった）し，