



Teaching AI in K-12:
Examples, Issues &
Guidance from K-12
CS Education Research

Shuchi Grover, PhD | @shuchig
Senior Research Scientist
Looking Glass Ventures

ISSEP 2023 | Oct. 24
HEP Vaud, Lausanne, Switzerland







September 1990



Why K-12 AI
Ed?
*(aka AI & the
coming
tsunami!)*

State of AI
education in
K-12
+
Examples

Lessons from K-12
CS Ed
*(i.e. let's not repeat
the same mistakes
or reinvent the
wheel!)*

Challenges, open
issues, &
recommendations
*(→ we'll get
through this!!!)*

New Amazon grant to boost computer science education in Duval elementary schools

Web Supervisor / August 4, 2023 / District, Elementary, Stem

Bringing computer science education to 11 million students

Branstad: Incentivize computer science in schools

Teaching China's next generation to express themselves in code

K-12 schools may see new computer science mandate

Microsoft expands computer science education program to Juarez, Mexico, and expands reach across 18 U.S. cities to improve high school students' access and equity

Southampton participates in global computer science campaign

[Computer Science Education](#)

3 ways teachers can navigate the evolving field of computer science



the **A.I.**
TSUNAMI

Artificial Intelligence

ng the industries
sharp increases in
consumption of
resources and
energy.

industries
consumption
of resources
and energy.

The New ChatGPT Can 'See' and 'Talk.' Here's What It's Like.

Job postings mentioning AI have more than doubled in two years, LinkedIn data shows

Published Wed, Oct 4 2023 1:49 AM EDT

Home > Strategy

AI, Automation and Growth Mindset Key to 2024 Plans

CHRIS CAMPBELL | OCTOBER 4, 2023



Next-Level Care: Generative AI Poised for \$22B Growth in Healthcare

The future is here

AI promises to remake the world. In many ways, it already has. Do we like what we've created?



[Can AI predict, and try to prevent, homelessness?](#)

'Counterfeit people': The dangers posed by Meta's AI celebrity lookalike chatbots

Generative AI Is the Newest Tool in the Dictator's Handbook

Trapped in a Dangerous Loop: Humans Inherit Artificial Intelligence Biases

TOPICS: Artificial Intelligence Psychology

By DEUSTO UNIVERSITY OCTOBER 3, 2023

AI advances 'supercharge' online disinformation, censorship and surveillance in growing threat to human rights, report warns

ARTIFICIAL INTELLIGENCE

Actor Tom Hanks Warns of Ad With AI Imposter

Overcoming Racial Bias In AI Systems And Startlingly Even In AI Self-Driving Cars

Racial bias in a medical algorithm favors white patients over sicker black patients

AI expert calls for end to UK use of 'racially biased' algorithms

AI Bias Could Put Women's Lives At Risk - A Challenge For Regulators

Gender bias in AI: building fairer algorithms

Bias in AI: A problem recognized but still unresolved

Amazon, Apple, Google, IBM, and Microsoft worse at transcribing black people's voices than white people's with AI voice recognition, study finds

Millions of black people affected by racial bias in health-care algorithms

Study reveals rampant racism in decision-making software used by US hospitals -- and highlights ways to correct it.

When It Comes to Gorillas, Google Photos Remains Blind

Google promised a fix after its photo-categorization software labeled black people as gorillas in 2015. More than two years later, it hasn't found one.

Google 'fixed' its racist algorithm by removing gorillas from its image-labeling tech

The Week in Tech: Algorithmic Bias Is Bad. Uncovering It Is Good.

Artificial Intelligence has a gender bias problem - just ask Siri

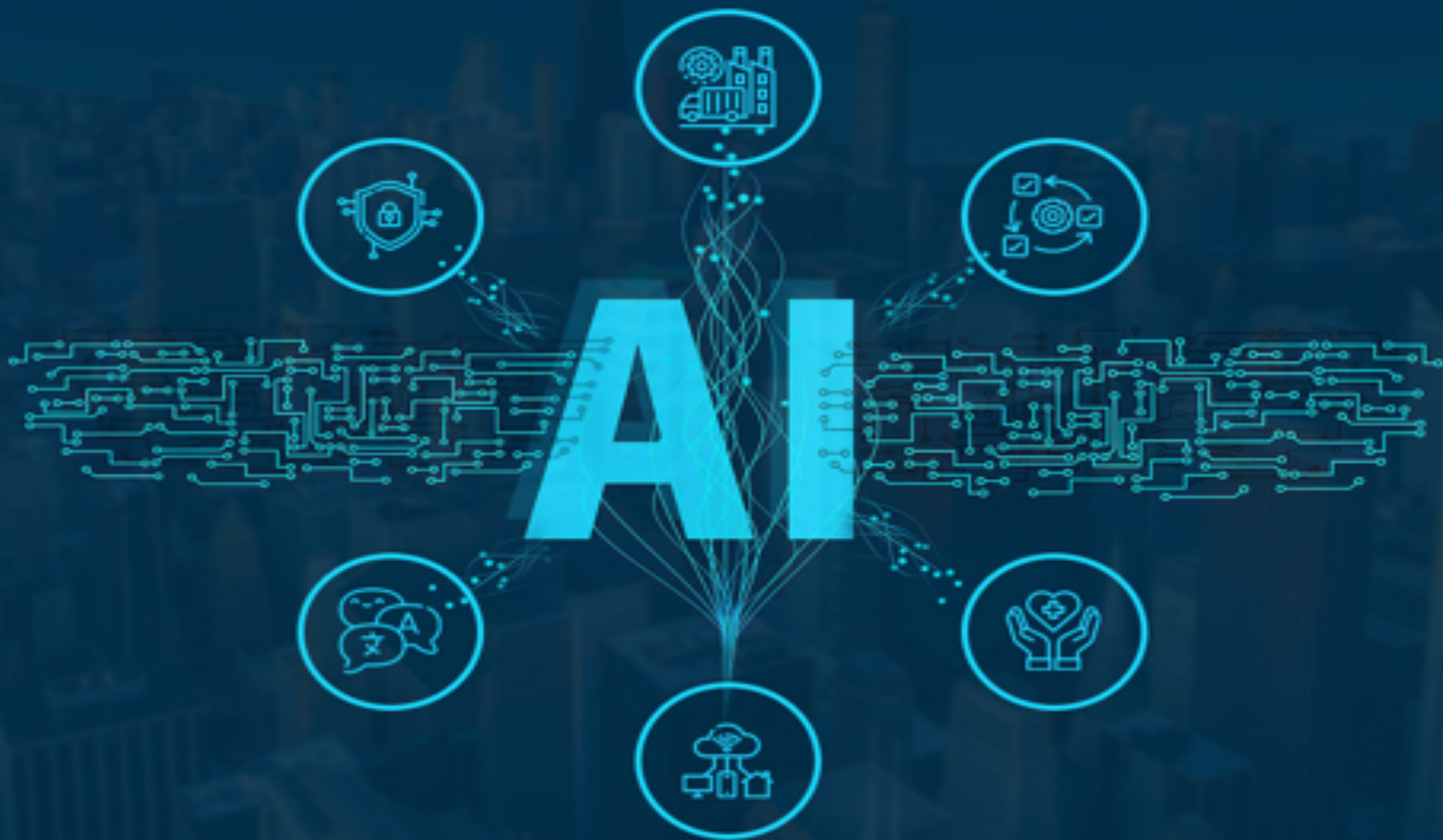
The Best Algorithms Struggle to Recognize Black Faces Equally

US government tests find even top-performing facial recognition systems misidentify blacks at rates five to 10 times higher than they do whites.

Experts disagree over threat posed but artificial intelligence cannot be ignored

Education:
the best
strategy
FOR thriving &
the best
inoculation
AGAINST misuse





**AI is transforming
our world.**

Let's #TeachAI

How To Use This Toolkit

[Incorporate AI in an Education System](#)

[Apply Seven Principles for AI in Education](#)

[View Sample School Guidance](#)

[Revise Existing Policies](#)

[Customize a Presentation](#)

[Engage Parents, Staff, and Students](#)

[Learn How AI was Used in This Toolkit](#)

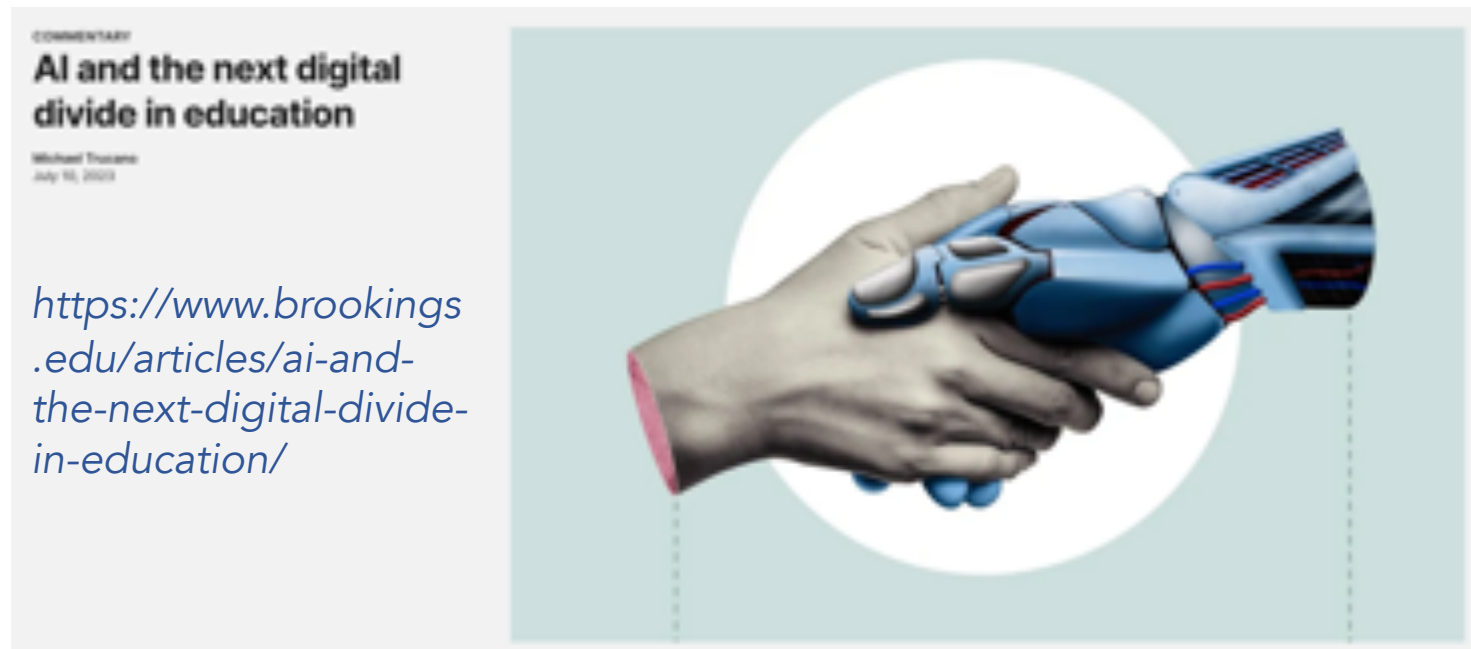
Doing nothing is not an option

Old Digital Divide

Computers in Schools, High Speed Internet, 1:1 devices...

New Digital Divide

Who is empowered to learn with and about AI



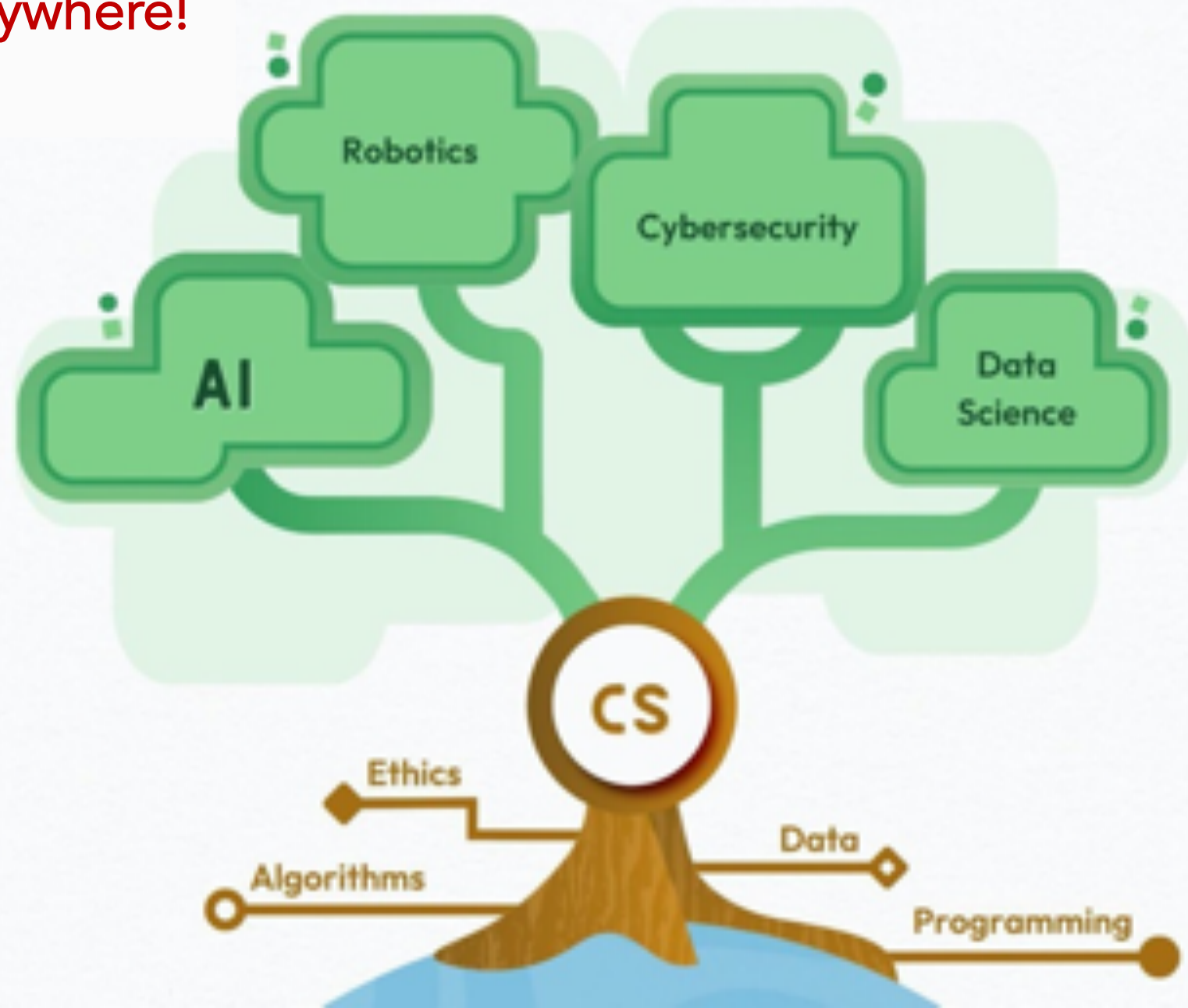
**K-12 CS
Education**

**K-12 AI
Education**



CS Education is not going anywhere!

In an age of AI,
CS isn't just
mandatory, it is
foundational

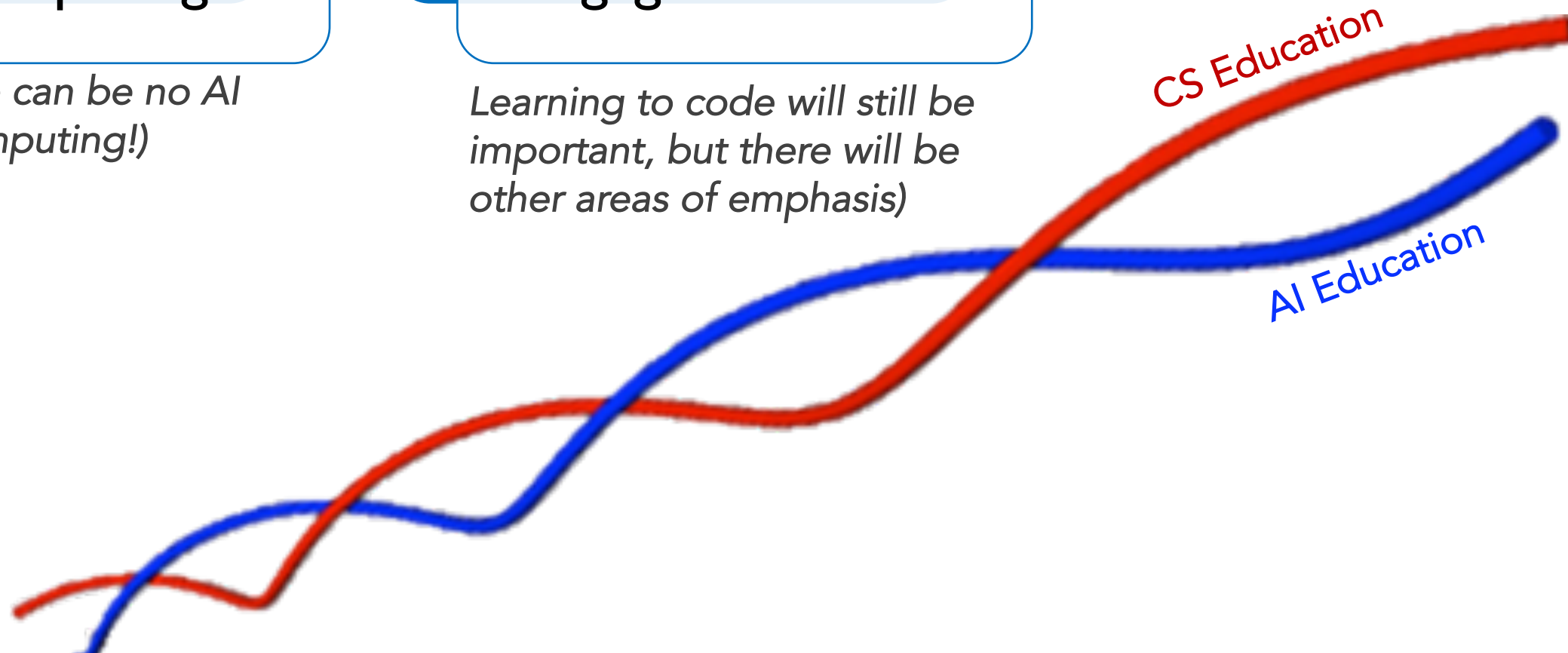


There is no AI education without some understanding of computing

(Psst...there can be no AI without computing!)

There is no CS education (today) without including engagement w/ AI

Learning to code will still be important, but there will be other areas of emphasis)





Why K-12 AI
Ed? (aka AI &
the coming
tsunami!)

State of AI
education in
K-12
+
Examples

Lessons from
K-12 CS Ed for
K-12 AI Ed

Challenges, open
issues, &
recommendations

Teaching AI to K-12 Learners: Lessons, Issues, and Guidance

Shuchi Grover

Looking Glass Ventures

Austin, TX, USA

shuchigrover@acm.org

ABSTRACT

There is growing recognition of the need to teach artificial intelligence (AI) and machine learning (ML) at the school level. This push acknowledges the meteoric growth in the range and diversity of applications of ML in all industries and everyday consumer products, with Large Language Models (LLMs) being only the latest and most compelling example yet. Efforts to bring AI, especially ML education to school learners are being propelled by substantial industry interest, efforts such as AI4K12, as well as technological developments that make sophisticated ML tools readily available to learners of all ages. These early efforts span a variety of learning goals captured by the AI4K12 "big ideas" framework and employ a plurality of pedagogies. This paper provides a sense for the current state of the field.

worse) in a myriad different ways every single day. Learning about a consequential new technology is an imperative, and especially one as powerful and versatile as AI, that is also considered accessible for motivating STEM learning and facilitating creativity [61].

The flurry of research, design, and development activity as well as academic writing in AI education in these past years has been intense. While still useful, landscape papers on the state of the art of ML teaching and learning in K-12 from just three years ago (for example, [38]), already seem dated! Symposia and panels on this topic have become a must-attend agenda item at every recent researcher and practitioner STEM education conference. Roughly 5 years into a period of intense research and development in K-12 AI/ML education, this position paper helps make sense of these efforts. Our goal is to synthesize recent literature, identify and

Teaching AI in K-12

Reviews

Ongoing/early research on K-12 AI education

Synthesizes

Key themes from ongoing efforts



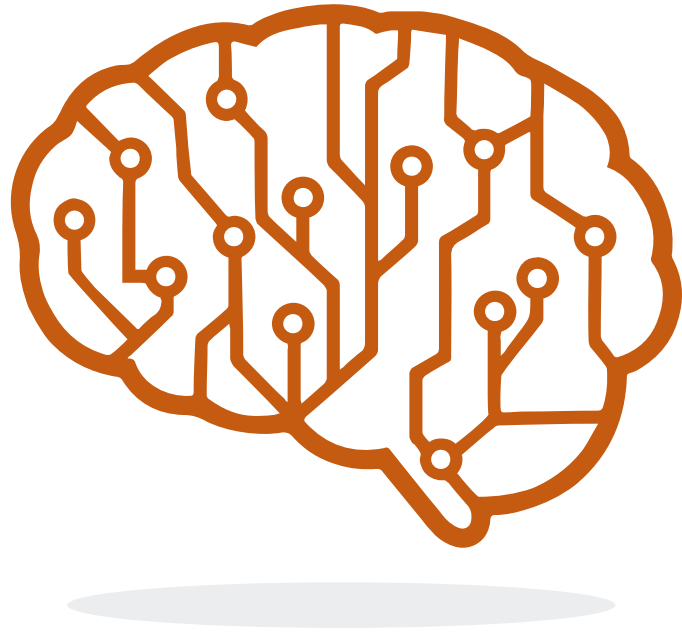
Identifies

Challenges & tensions in tackling "the next new thing"

Suggests

Takeaways & Guidance

Diverse goals of AI research projects



Developing broad AI literacy (awareness about AI)



Developing understandings of AI tools & AI/ML techniques



Designing ethics- and fairness-focused experiences



Teaching AI basics from a CS topics lens



Examining the appropriateness of diverse pedagogies for developing understanding about AI (a very technical topic)



Examining pedagogies that serve well-defined purpose(s) (e.g., broadening participation among women or youth belonging to specific groups,



Lifting the hood on how AI/ML works (making AI less magical)



Integrating AI learning into/with other subjects



Co-designing with teachers and teacher preparation

Key Emergent Themes & Outcomes

What we should be teaching in schools about AI

Data Primacy & 'CT 2.0'

Importance of ethics, bias, and critical examination of AI

Pedagogies & instructional approaches

Ways to integrate AI into other subjects

Curricular co-design & teacher preparation

+ A plethora of free curricular resources

Key Emergent Themes & Outcomes

What we should be teaching in schools about AI

Data Primacy & 'CT 2.0'

Importance of ethics, bias, and critical examination of AI

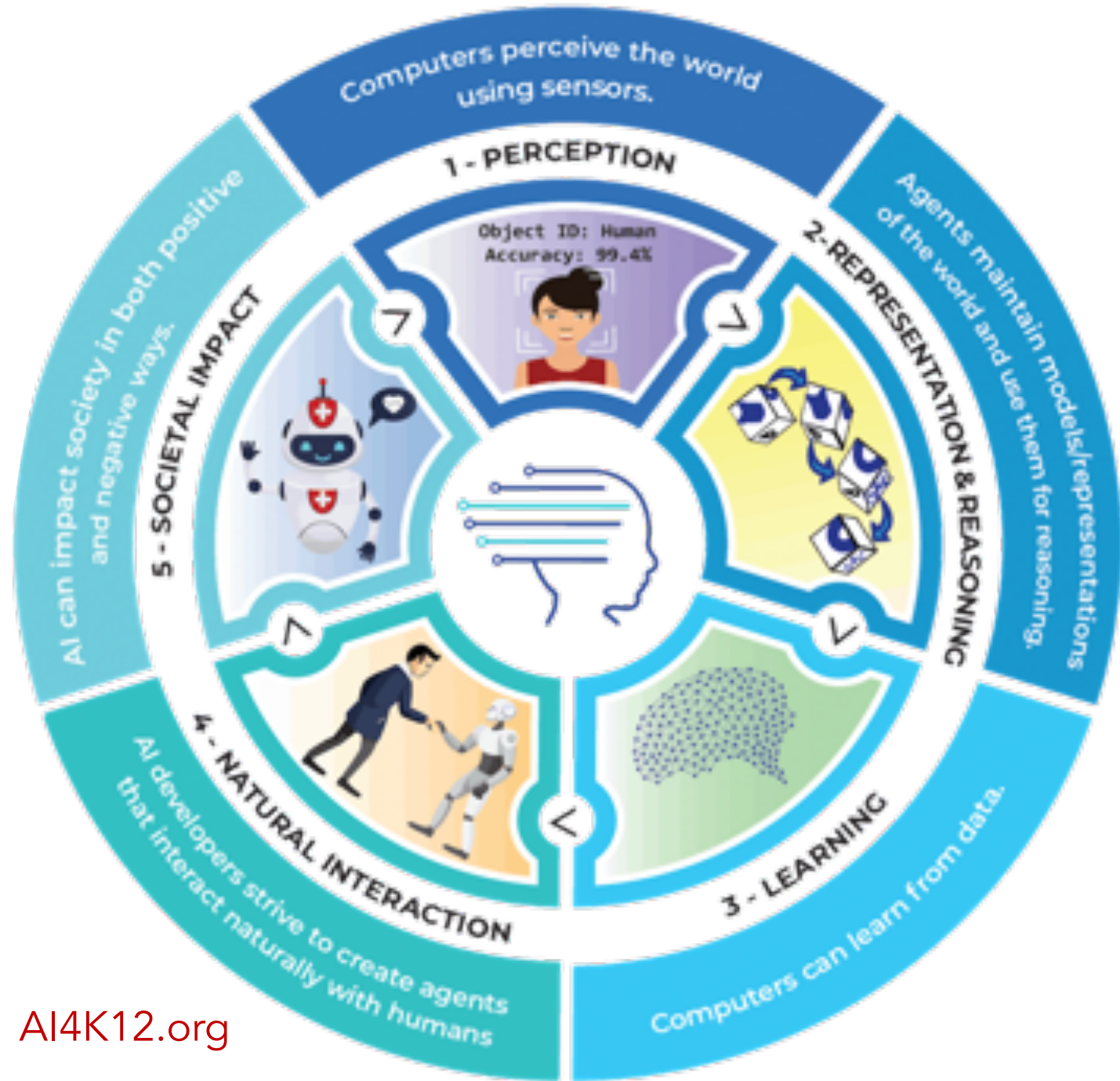
Pedagogies & instructional approaches

Ways to integrate AI into other subjects

Curricular co-design & teacher preparation

+ A plethora of free curricular resources

What learners should know & be able to do



Five Big Ideas in Artificial Intelligence

1. Perception

Computers perceive the world using sensors. Perception is the process of extracting meaning from sensory signals. Making computers “see” and “hear” well enough for practical use is one of the most significant achievements of AI to date.

2. Representation & Reasoning

Agents maintain representations of the world and use them for reasoning. Representation is one of the fundamental problems of intelligence, both natural and artificial. Computers construct representations using data structures, and these representations support reasoning algorithms that derive new information from what is already known. While AI agents can reason about very complex problems, they do not think the way a human does.

3. Learning

Computers can learn from data. Machine learning is a kind of statistical inference that finds patterns in data. Many areas of AI have progressed significantly in recent years thanks to learning algorithms that create new representations. For the approach to succeed, tremendous amounts of data are required. This “training data” must usually be supplied by people, but is sometimes acquired by the machine itself.

5. Societal Impact

AI can impact society in both positive and negative ways. AI technologies are changing the ways we work, travel, communicate, and care for each other. But we must be mindful of the harms that can potentially occur. For example, biases in the data used to train an AI system could lead to some people being less well served than others. Thus, it is important to discuss the impacts that AI is having on our society and develop criteria for the ethical design and deployment of AI-based systems.

4. Natural Interaction

Intelligent agents require many kinds of knowledge to interact naturally with humans. Agents must be able to converse in human languages, recognize facial expressions and emotions, and draw upon knowledge of culture and social conventions to infer intentions from observed behavior. All of these are difficult problems. Today's AI systems can use language to a limited extent, but lack the general reasoning and conversational capabilities of even a child.



Cinq grandes idées en intelligence artificielle

5. Impact social

L'IA peut avoir un impact sur la société qui peut prendre à la fois des formes positives et négatives. Les technologies de l'IA modifient nos rapports au travail, au voyage, à la communication et à la manière de nous occuper les uns des autres. Mais nous devons être conscients d'éventuels dommages qui pourraient se produire. Par exemple, les biais dans les données utilisés pour alimenter un système de l'IA peuvent engendrer des disparités de traitement envers des personnes. Pour ces raisons, il est important de débattre les impacts de l'IA sur notre société et de développer des critères éthiques pour la conception et l'application des systèmes basés sur l'IA.

4. Interaction naturelle

Les agents intelligents acquièrent de nombreux types de connaissances pour interagir naturellement avec les humains. Les agents doivent être capables de communiquer en langage humain, de reconnaître les expressions faciales et les émotions, de disposer des connaissances des conventions sociales et culturelles pour en déduire des intentions à partir du comportement observé. Ce sont là des problèmes complexes. Les systèmes de l'IA actuels utilisent de manière limitée le langage, ils ne possèdent même pas les capacités générales de raisonnement et de conversation d'un enfant.

1. Perception

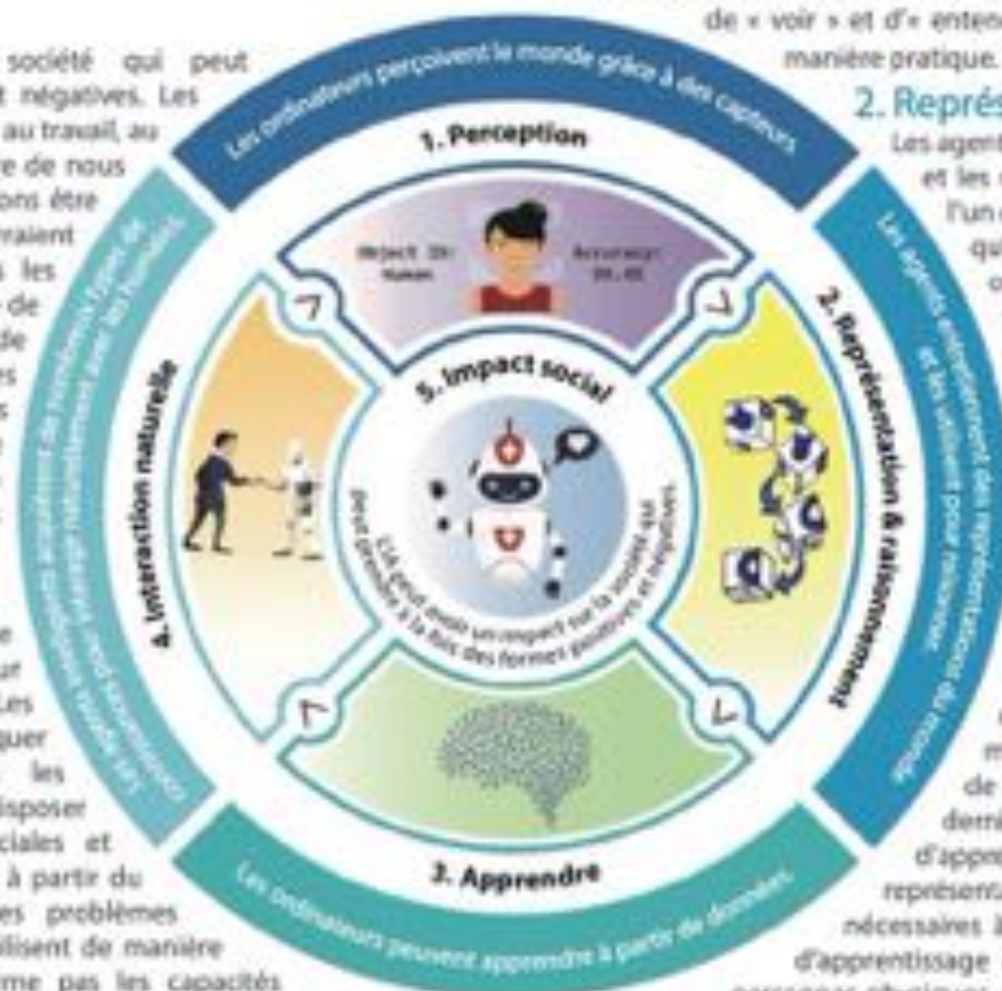
Les ordinateurs perçoivent le monde grâce à des capteurs. La perception est le processus qui consiste à extraire un sens des signaux sensoriels. L'une des réalisations les plus significatives de l'IA à ce jour est de permettre aux ordinateurs de « voir » et d'« entendre » suffisamment bien pour être utilisés de manière pratique.

2. Représentation et raisonnement

Les agents entretiennent des représentations du monde et les utilisent pour raisonner. La représentation est l'un des problèmes fondamentaux de l'intelligence, qu'elle soit naturelle ou artificielle. Les ordinateurs construisent des représentations en utilisant des structures de données. Ces représentations sont la base de raisonnements algorithmiques qui déduisent de nouvelles informations à partir de ce qui est déjà connu. Bien que les agents de l'IA soient capables de raisonner des problèmes complexes, ils ne les abordent pas de la même manière que les humains.

3. Apprendre

Les ordinateurs peuvent apprendre à partir de données. L'apprentissage automatique est une sorte d'inférence statistique qui permet de modéliser les données. De nombreux domaines de l'IA ont progressé de manière significative ces dernières années grâce à des algorithmes d'apprentissage qui créent de nouvelles représentations. D'énormes quantités de données sont nécessaires à la réussite de cette approche. Ces données d'apprentissage doivent être généralement fournies par des personnes physiques, mais parfois elles peuvent être acquises par la machine elle-même.



Primacy of Data & Data Science

Data Agency

Data agency extends the concept of data literacy by emphasizing people's ability to not only understand data, but also to actively control and manipulate information flows and to use them wisely and ethically.



Data Literacy

Involves understanding what data one creates, what happens to them, and with what consequences

Data Agency

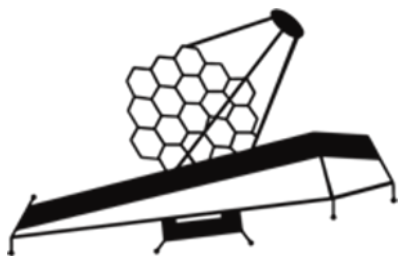
Refers to people's volition, skills, attitudes, and capacity for informed actions that make a difference in their digital world.

A more **active concept**.

Matti Tedre, Henriikka Vartiainen, Juho Kahila, Tapani Toivonen, Ilkka Jormanainen, Teemu Valtonen. Machine learning introduces new perspectives to data agency in k—12 computing education. In 2020 IEEE Frontiers in Education Conf., pages 1–8. IEEE, 2020.

Computational Thinking

A way of solving problems and designing systems that draw on concepts fundamental to computer science. Learning how AI works is an opportunity to learn computational thinking



- **Decomposition:** Understanding complex AI problems requires breaking them down into smaller, more manageable parts.
- **Pattern Recognition:** Machine learning relies on recognizing patterns in data. Students practice and develop skills in identifying patterns and trends.
- **Algorithmic Thinking:** Learning about AI exposes students to algorithms, step-by-step solutions to a problem, from simple decision trees to more complex processes.
- **Debugging:** As with any computational task, AI models can sometimes produce unexpected or incorrect results. Solving these issues is central to both AI and computational thinking.
- **Evaluation:** AI frequently demands the assessment of different solutions; this mirrors a key aspect of computational thinking, where solutions are tested and refined based on outcomes.

PROBLEM SOLVING WORKFLOWS

CT 1.0 (RULE-DRIVEN)	CT 2.0 (DATA-DRIVEN)
Formalize the problem	Describe the job and collect data from the intended context
Design an algorithmic solution	Filter and clean the data. Label the data
Implement a solution in a stepwise program	Train a model from the available data
Compile and execute the program	Evaluate and use the model

Matti Tedre, Tapani Toivonen, Juho Kahila, Henriikka Vartiainen, Teemu Valtonen, Ilkka Jormanainen, and Arnold Pears. 2021. Teaching Machine Learning in K–12 Classroom: Pedagogical and Technological Trajectories for Artificial Intelligence Education. IEEE Access 9 (2021), 110558–110572.

Paradigmatic Differences in Machine Learning

~~CONCEPTUAL CHANGES IN COMPUTING EDUCATION~~

CT 1.0	CT 2.0
Correctness can be formally proven	Models may display higher or lower confidence, efficiency
Debugging: Tracking and tracing	Evaluate the model wrt predictions
Deductive problem-solving	Inductive problem-solving
Transparent structure	Black-boxed
Stepwise, deterministic, discrete flow of program through states	Parallel, possibly nondeterministic passing data through a network
Structured data	Unstructured data

Paradigmatic Differences in Machine Learning

~~CONCEPTUAL CHANGES IN COMPUTING EDUCATION~~

CT 1.0	CT 2.0
Reductionism	Emergence
Formal verification	Statistical measures
Black/glass box testing	Black box testing
No tinkering, toying, trial-and-error	Experimenting with data, parameters, hyperparameters
Prepare for worst-case complexity, optimize for average case	No time/space variance between passes of data through the network
Tedious to ensure portability	Straightforwardly portable

AI Ethics

Real world patterns of health inequality and discrimination



Unequal access and resource allocation



Discriminatory healthcare processes



Biased clinical decision making

Discriminatory data



Sampling biases and lack of representative datasets



Patterns of bias and discrimination baked into data distributions

Application injustices



Disregarding and deepening digital divides



Exacerbating global health inequality and rich-poor treatment gaps



Hazardous and discriminatory repurposing of biased AI systems

World → Data

Use ← Design

Biased AI design and deployment practices



Power imbalances in agenda setting and problem formulation



Biased and exclusionary design, model building and testing practices



Biased deployment, explanation and system monitoring practices

Ethics, bias, and critical examination of AI baked into all curricular experiences

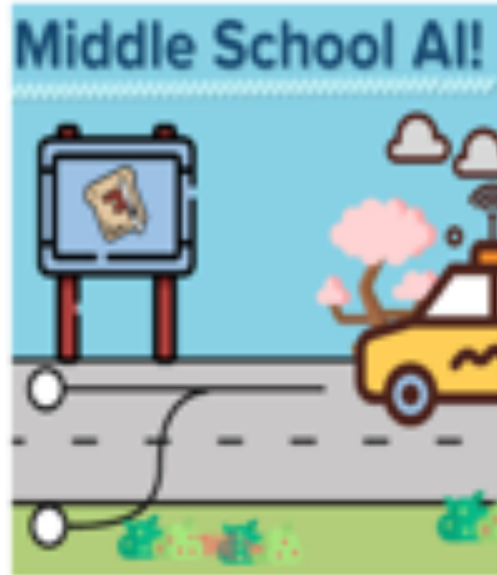


Inclusive AI Literacy & Learning



Inclusive AI Literacy & Learning is a curriculum project that seeks to teach AI literacy while emphasizing accessibility, equity and adaptability for all.

DAILY Curriculum for Middle School



DAILY is a middle school AI curriculum focusing on AI concepts, ethical issues in AI, creative expression using AI, and how AI relates to your future.

AI & Ethics for Middle School



The AI & Ethics Project seeks to develop an open source curriculum for middle school students on the topic of artificial intelligence and its ethical implications.

Personal Image Classifier: PICaboo



Students can create and train image classification models in App Inventor.

Supervised Reinforcement Learning



Black Life In the Age of Artificial Intelligence (AI)



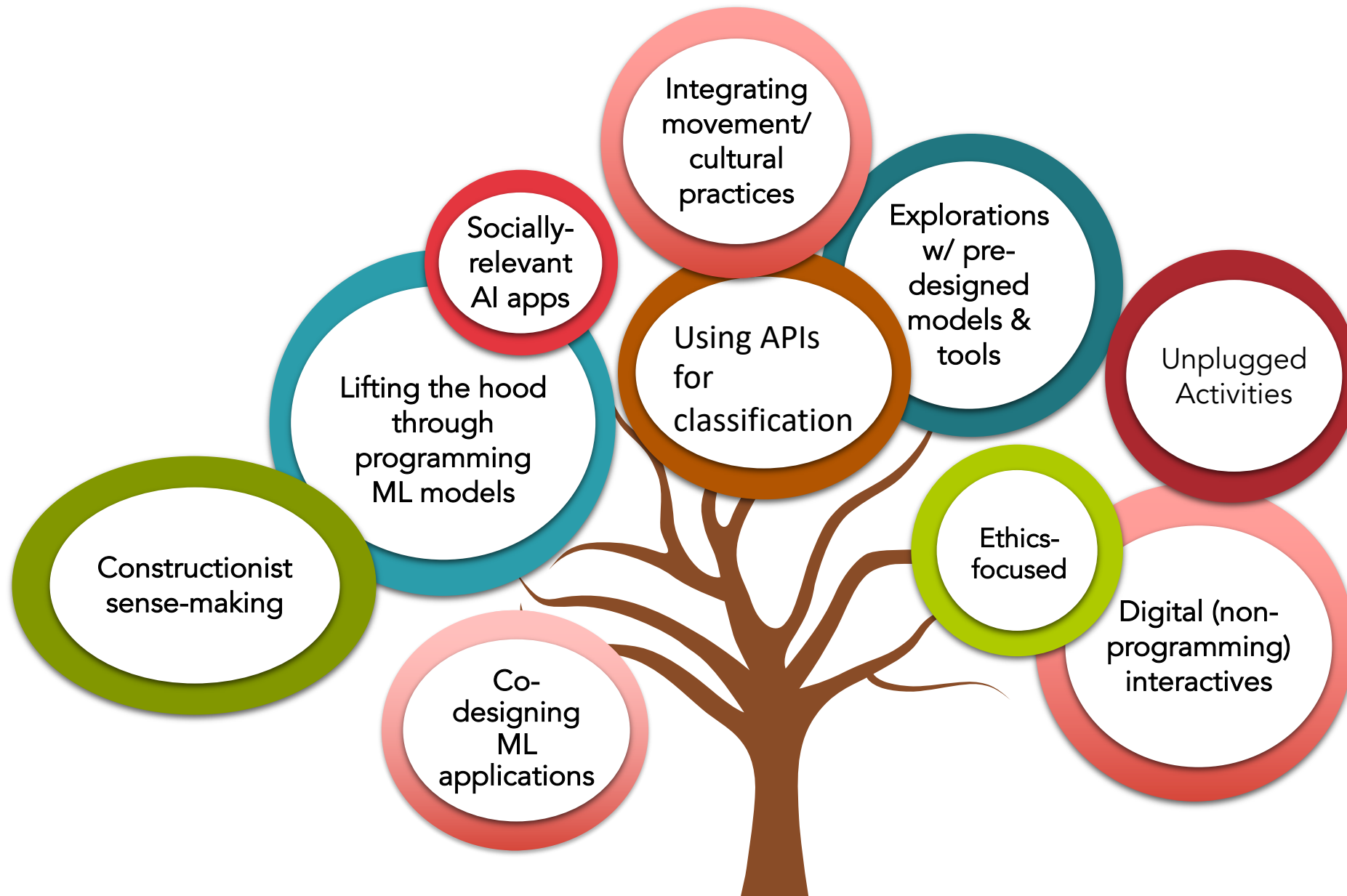
BLACK LIFE
IN THE AGE OF AI

Introduction:

This syllabus aims to bring learners of many backgrounds together to explore content related to Black Life and AI. The term Black Life refers to the varied experiences of Black folks including in terms of identity and experience. Thus we explore race, gender, sexuality, and ability alongside topics such as the internet, surveillance, and education. The syllabus can be used linearly, with a set of information explored per week, or at your own leisure. Over the course of engagement, learners will be exposed to a wide variety of topics that push you to 1) grapple with the ethics of technology, 2) communicate the implications of technology in multiple contexts and create visual media. Learners should leave the site with both tools for conversation with the public, and having had opportunities to reflect on the role of technology in their own lives.

Below you will find sections organized by topic with [links](#) to varying resources such as articles, reports, books, podcasts, and videos.

Plurality of Pedagogies for Teaching AI/ML



Two NSF-Funded R&D Efforts to Design AI learning for HS Students

CS Frontiers (CSF)

- 4 year effort to design & test a year-long modular curriculum
 - Distributed Computing, AI/ML, Cybersecurity/Internet Of Things, Software Engineering (capstone)
- Focus on engaging HS girls in CS
- Interdisciplinary connections
- Team: Vanderbilt, NCSU, LGV

AI & Cybersecurity for Teens (ACT)

- 2 year exploratory effort
- Focus on (a) Integration of AI and Cybersecurity (NSF SaTC-Edu) → Teaching AI in the context of real-world CyS issues (eg. bot detection, DDoS mitigation) (b) Lift the hood on how AI/ML works
- Team: LGV, Vanderbilt, UNO

Both rely on NetsBlox (an extension of Snap! with amazing capabilities)



Brian Broll

Limitless

Just two simple distributed programming concepts - message passing and remote procedure calls - open a world of possibilities.



Institute for Software Integrated Systems Vanderbilt University

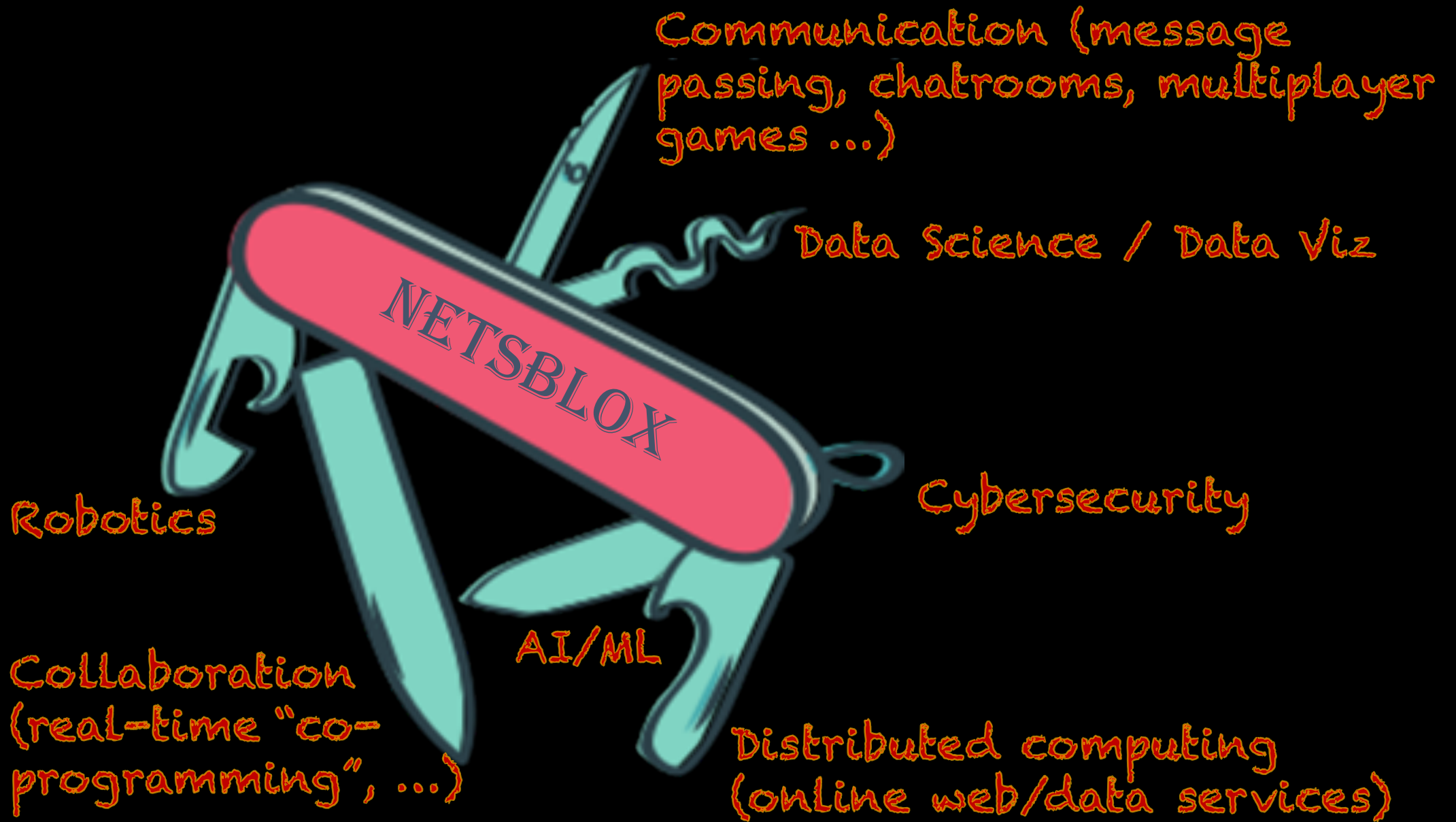


Akos Ledeczi

NetsBlox is a visual programming language and cloud-based environment that enables novice programmers to create networked programs such as multi-player games. Its visual notation is based on Scratch and it uses the open source JavaScript code base of Snap! NetsBlox opens up the internet with its vast array of public domain scientific and other data sources making it possible to create STEM projects, such as displaying seismic activity anywhere on Earth using an interactive Google Maps background. Similarly, weather, air pollution, and many other data sources such as the Open Movie Database and the Sloan Digital Sky Server are available. NetsBlox also supports collaborative editing similar to Google Docs.



Opening up Snap! to the web unleashes a world of possibilities





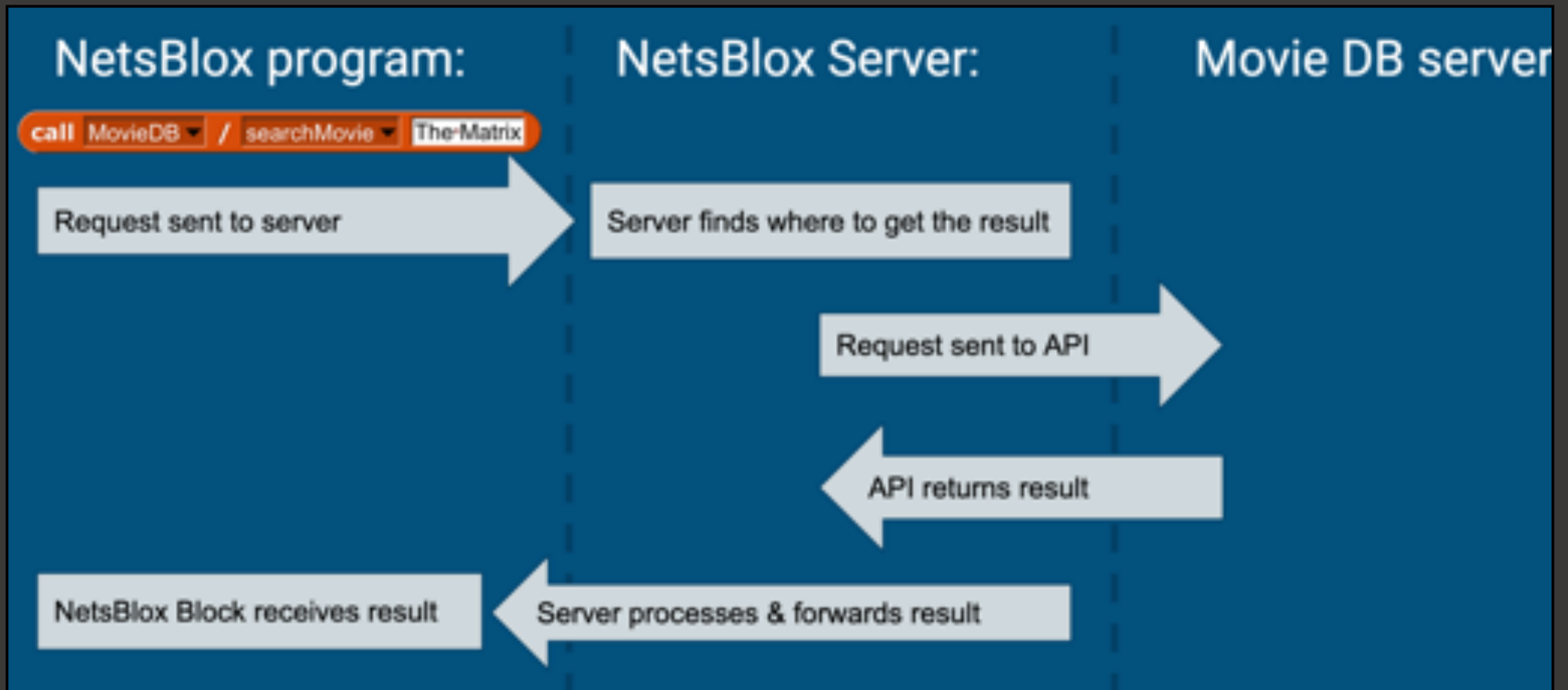
TWO SIMPLE ABSTRACTIONS

Remote Procedure Calls

- RPC: call functions on the server
 - Multiple input arguments
 - One output argument
 - Blocking call (waits until it gets response)
 - Works like any other custom block

call /

- Similar RPCs are grouped into a 'service'
 - Google Maps, weather, earthquake, museums, movies etc.



How a NetsBlox RPC works

ONLINE DATA AND WEB SERVICES

call AirQuality latitude longitude

- Alexa
- ArtificialIntelligence ▶
- Autograders
- Chart
- Climate ▶
- CloudVariables
- Community ▶
- Database ▶
- Execute
- Games ▶
- Geolocation
- GoogleMaps
- GoogleStreetView
- History ▶
- KeyValueStore
- Language ▶
- MetMuseum
- MovieDB
- Music ▶
- NewYorkTimes
- NexradRadar
- PhoneloT
- Pixabay
- PublicRoles
- RoboScape
- Science ▶
- ServiceCreation
- Thingspeak
- Traffic
- Twitter

- AirQuality
- COVID-19
- EarthOrbit
- Earthquakes
- Eclipse2017
- HistoricalTemperature
- HumanMortalityDatabase
- HurricaneData
- IceCoreData
- MaunaLoaCO2Data
- NASA
- OceanData
- PaleoceanOxygenIsotopes
- StarMap
- WaterWatch
- Weather

call CloudVariables

- Alexa
- ArtificialIntelligence ▶
- Autograders
- BingTraffic
- Chart
- Climate ▶
- CloudVariables
- Community ▶
- Database ▶
- Execute
- Games ▶
- Geolocation
- GoogleMaps
- GoogleStreetView
- History ▶
- IoTScope
- KeyValueStore
- Language ▶
- MetMuseum
- MovieDB
- Music ▶
- NewYorkTimes
- PhoneloT
- Pixabay
- PublicRoles
- RoboScape
- Science ▶
- ServiceCreation
- Thingspeak
- Twitter

- 18001767679 ▶
- Device ▶
- Knemeyer ▶
- TheMapotakes ▶
- Wangs33 ▶
- Yutong ▶
- brian ▶
- cbrady ▶
- des1303 ▶
- des1303netsblox ▶
- devinjean ▶
- jacobmorrison ▶
- ledeczi ▶
- nocalbruin ▶
- roadlabs ▶

- SummerOlympicMedals
- brian's First Service
- brian's soccer service

Mauna Loa v2

Sprite
draggable

Scripts Costumes Sounds Room

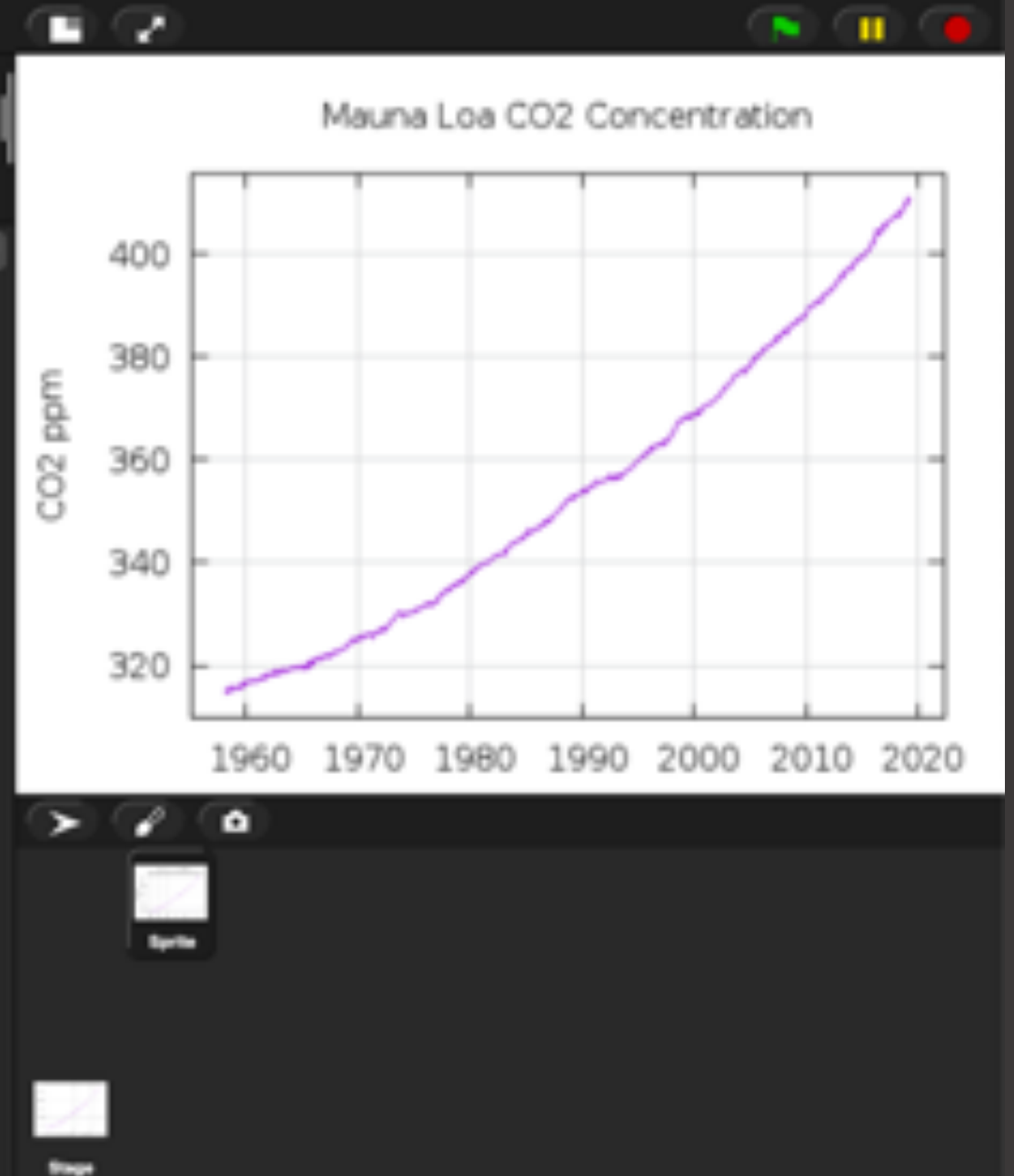
when clicked

switch to costume

call Chart / draw call MaunaLoaCO2Data / getCO2Trend startyear endyear

ChartSetup

Check out the "low floor" !!!



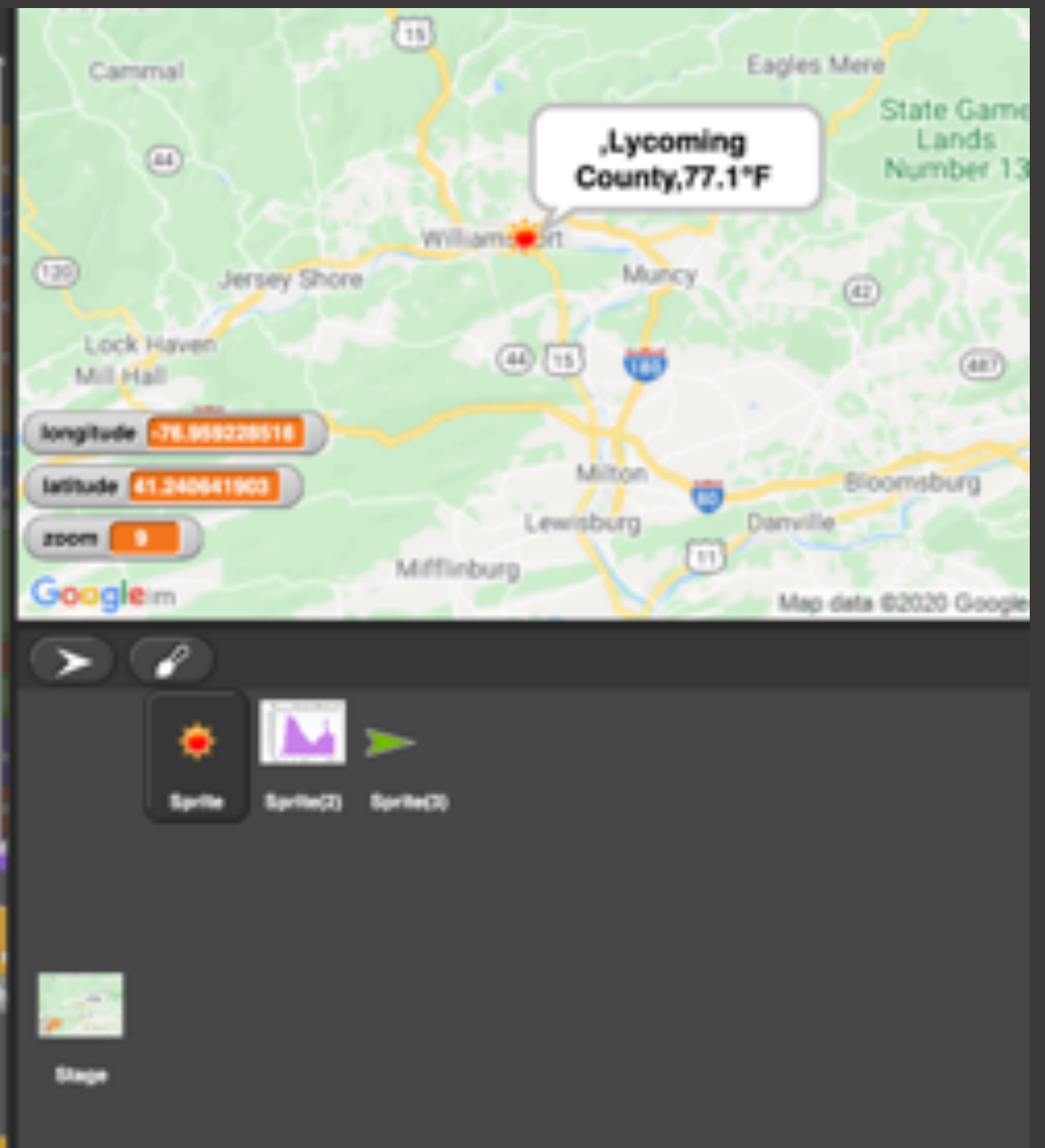
Charting climate data using Temp, CO2, Ice core, ... data over time

Scripts Costumes Sounds Room

```
when I receive Map Clicked
hide
point in direction 90
go to mouse pointer
set latitude to call GoogleMaps / getLatitudeFromY y position
set longitude to call GoogleMaps / getLongitudeFromX x position
set sprite lat to latitude
set sprite long to longitude
set country to call Geolocation / country latitude longitude
hide
say
join
  join call Geolocation / city sprite lat sprite long
  join call Geolocation / county sprite lat sprite long
call Weather / temperature sprite lat sprite long
unicode 123 as letter
hide
switch to costume call Weather / icon sprite lat sprite long
broadcast update map click
set zoom to zoom
go to x: 0 y: 0
show

when I receive second show
sprite show

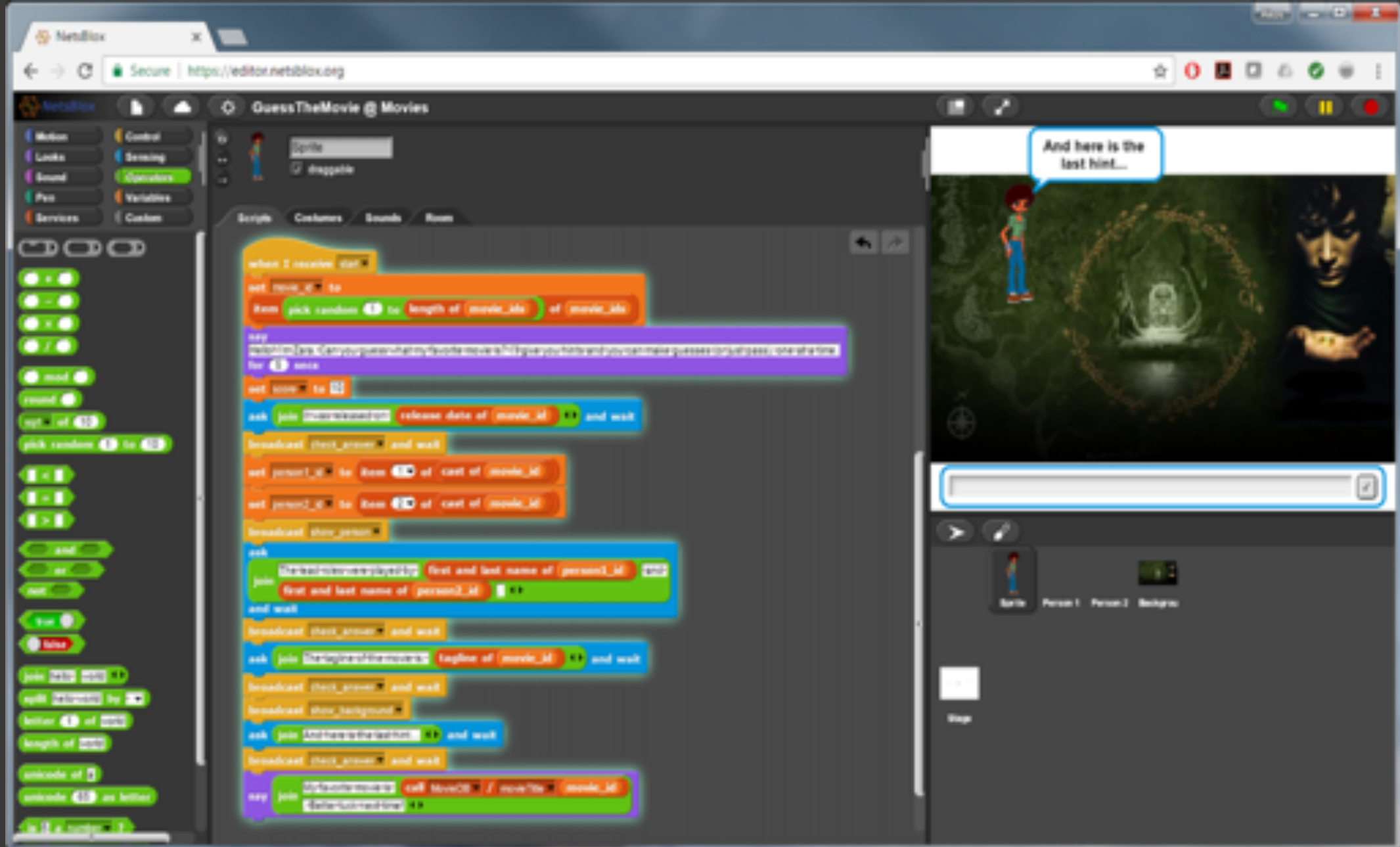
when clicked
go back 2 layers
hide
```



Weather/AQI/... using the Google Maps/Geolocation/Weather



Sky navigator app using the Sloan Digital Sky Survey



Movie guessing game using The Movie Database (TMDB)

CastShow


Scripts Costumes Sounds Room

when clicked
go to x: 133 y: 6
set title to **Hidden Figures**
broadcast Find Movie
forever
ask and wait
set title to answer
broadcast Find Movie

when I receive show
set size to 100 %
CyclePhotos item 1 of cast

when I receive Find Movie
set film to item 1 of call MovieDB / searchMovie / title
set cast to call MovieDB / movieCastPersonIDs / film
set title to call MovieDB / movieTitle / film
broadcast show

Hidden Figures



Search

1 2 3 Scribble

Hidden Figures Cast Show

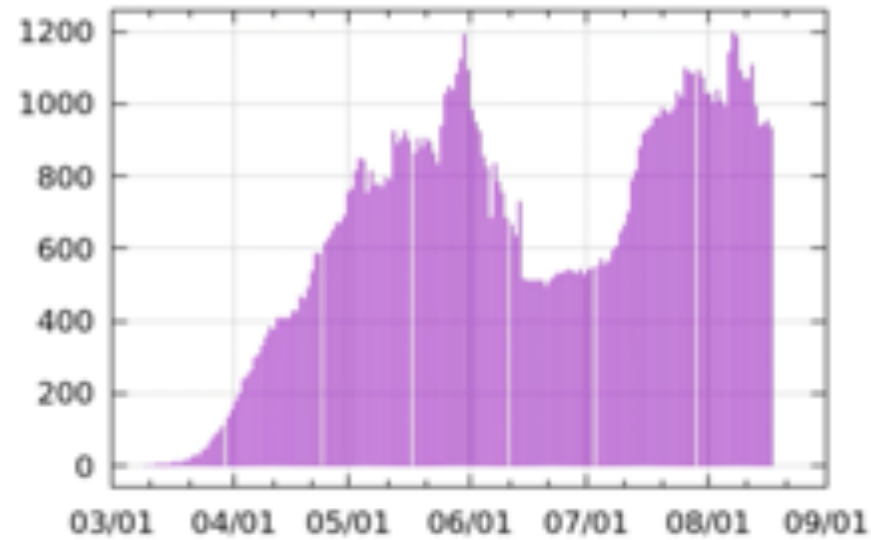
```

Sprite(2)
[7] draggable

Scripts
Costumes
Sounds
Room

set x long to longitude
set country to call Geolocation / country / s lat / s long
set state to call Geolocation / state / s lat / s long
if not is country a list?
  play sound
else
  set options to
  list [TimeSeries true] list [linePlotFormat NumRad]
  list [lineDisplayFormat NumRad] list [type] list [area]
  list [height] stage height list [width] stage width
  if country = United States
    set cases to
    process data call COVID-19 / getConfirmedCounts / state / # of days averaged
    set value of [0] in options to
    join
    New-COVID-19 state Cases [averaged] # of days averaged [days]
  else
    set cases to
    process data call COVID-19 / getConfirmedCounts / country / # of days averaged
    set value of [0] in options to
    join
    New-Daily-COVID-19-Cases-in country [averaged] # of days averaged [days]
  if is cases a list?
    go to x: 0 y: 0
    switch to costume call Chat / draw / list cases / options
  
```

New COVID-19 India Cases (averaged over 7 days)



Navigation icons: Home, Edit, Play, Stop, Refresh, Zoom In, Zoom Out, Full Screen, Close.

Stage area: Sprite, Sprite(2), Sprite(3), Stage.

Covid-19 visualization by region using Google Maps/Geolocation/COVID-19/gnuplot

Data & Transdisciplinary Learning

Data is / are everywhere!





TWO SIMPLE ABSTRACTIONS

Message Passing

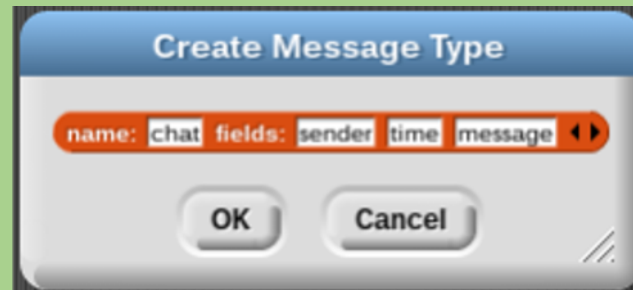
- Events in Scratch:



- Messages in NetsBlox:



- Message type editor:



– CS Frontiers: Engaging Female High School Students in New Frontiers of Computing (CSFrontiers)

Research Team



Tiffany Barnes
(PI)
NCSU



Shuchi Grover
(PI)
LGV



Akos Ledeczki
(PI)
Vanderbilt



Powered by
NetsBlox



Veronica Catete
NCSU



Brian Broll (Co-PI)
Vanderbilt



Marnie Hill
NCSU



#1949472,
1949488,
1949492

CSF Teacher Partners



Omar Ali
(Northeast HS, PA)



Madeline Drayton
(Providence HS, NC)



Maryellen Magee
(WAH HS, NC)



Ian VanBuskirk
(Garner Magnet, NC)



April Collins
(MLK, Nashville)



Tammy Gose
(Wilson County, TN)



Susan Ettenheim
(ERHS, NYC)



Padmaja Bandaru
(AMSA Charter School)





Distributed
Computing



AI/ML



Cybersecurity



Software Engineering



Project Based Learning

Engage non-dominant grps (esp. girls)

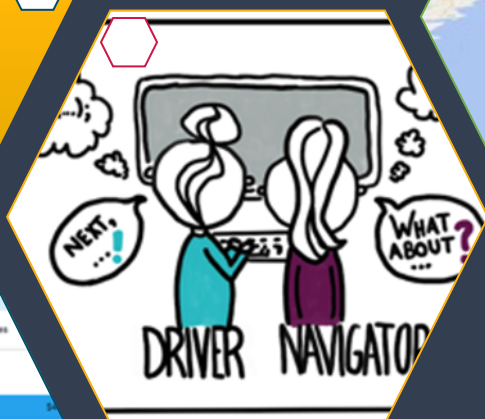


Engage in Data Science

Collab. Activities

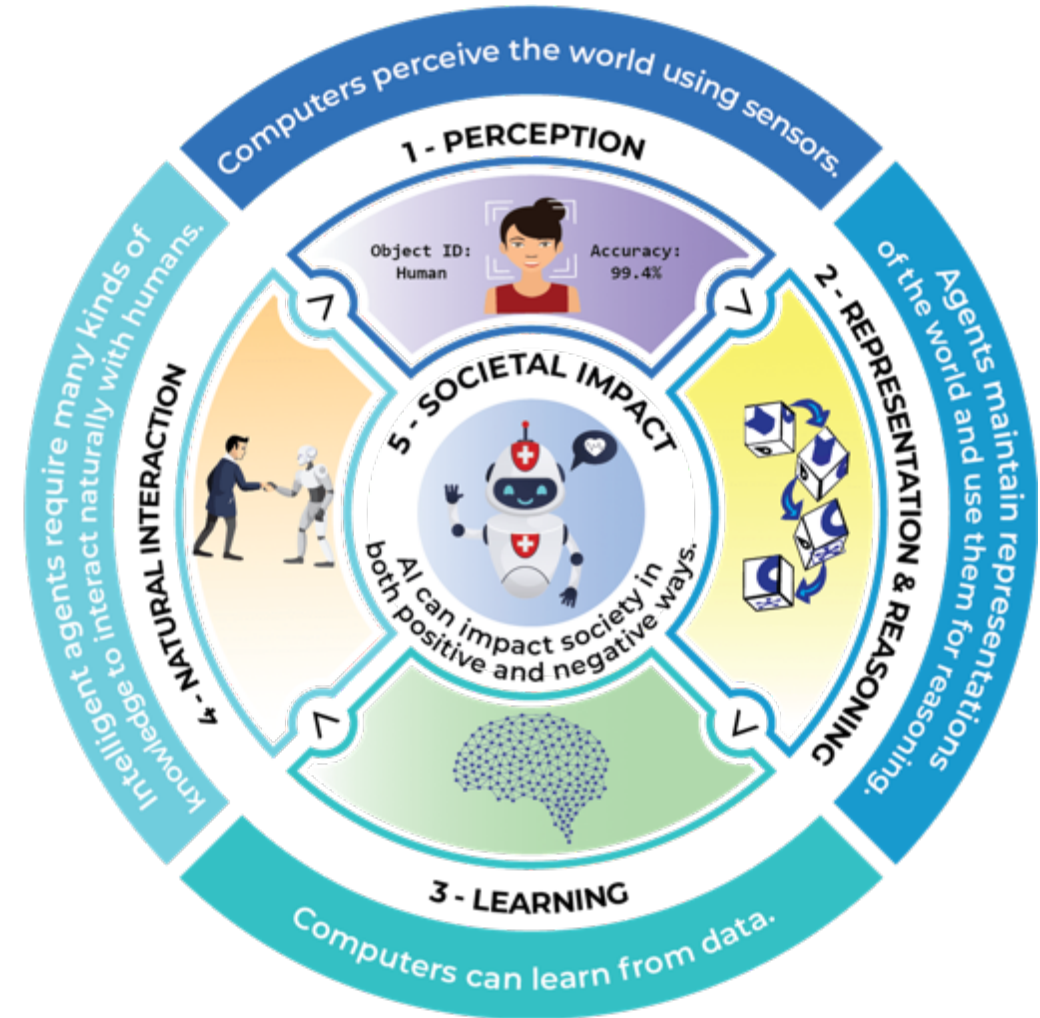


Integration & multi-disci. learning



AI/ML Curriculum Design/Curation

1. NetsBlox Activities
2. Open-source AI and ML materials for K-12 students:
 - i. AI4All: AI Bytes Units
 - ii. AI4K12
 - iii. Ecraft2learn
3. Connected activities to the AI4K12 learning objectives in the AI4K12 Big Ideas Progression chart



CS Frontiers AI / ML Module Overview

Unit	Main Topic		Activities
Unit 0	Netsblox refresher		Introduction to Netsblox Activity
Unit 1	Introduction to AI	AI	Google Quick, Draw!
Unit 2	Algorithms and Search		Map Breadth First Search
Unit 3	Introduction to Machine Learning		Twitter Bot Classification
Unit 4	Introduction to Natural Language Process and Sentiment Analysis	ML	Music Sentiment Using Genius API
Unit 5	Sentiment Analysis		Students expand on above activity
Unit 6	Bias in Datasets, Ethics in AI/ML		Students create presentation on bias
Unit 7	Other ML Techniques	More ML and Other Topics	Imitation Learning Game
Unit 8	Deep Neural Networks		Square Root Predictions with Neural Networks
Unit 9	Real World Application of AI/ML		AI & Criminal Justice, AI & Environment

“Sentimental Writer” (NLP/Text Analysis)

“Classify” a text as positive, negative, or neutral sentiment using a publicly-available text classification API called “Parallel Dots” an AI powered NLP API
https://apis.paralleldots.com/text_docs/index.html




The image shows a REST client interface. On the left, there is a call configuration bar with the following elements: a red button labeled 'call', a dropdown menu showing 'ParallelDots', a slash separator, another dropdown menu showing 'getSentiment', and a text input field containing 'text'. On the right, there is a response window titled 'sentiment' containing a table with the following data:

	A	B
1	negative	0.012
2	neutral	0.412
3	positive	0.575

Part 1 (Unplugged/Pseudocode)

Read through the following [pseudocode](#) and think about what this program is doing and how the program should be coded

```
When  clicked
  Continue Forever
    Ask a question and wait
    Set message to the answer of the question
    Set sentiment to a RPC that identifies the sentiment of message
    Set strongest sentiment to the first item in a list of
sentiments

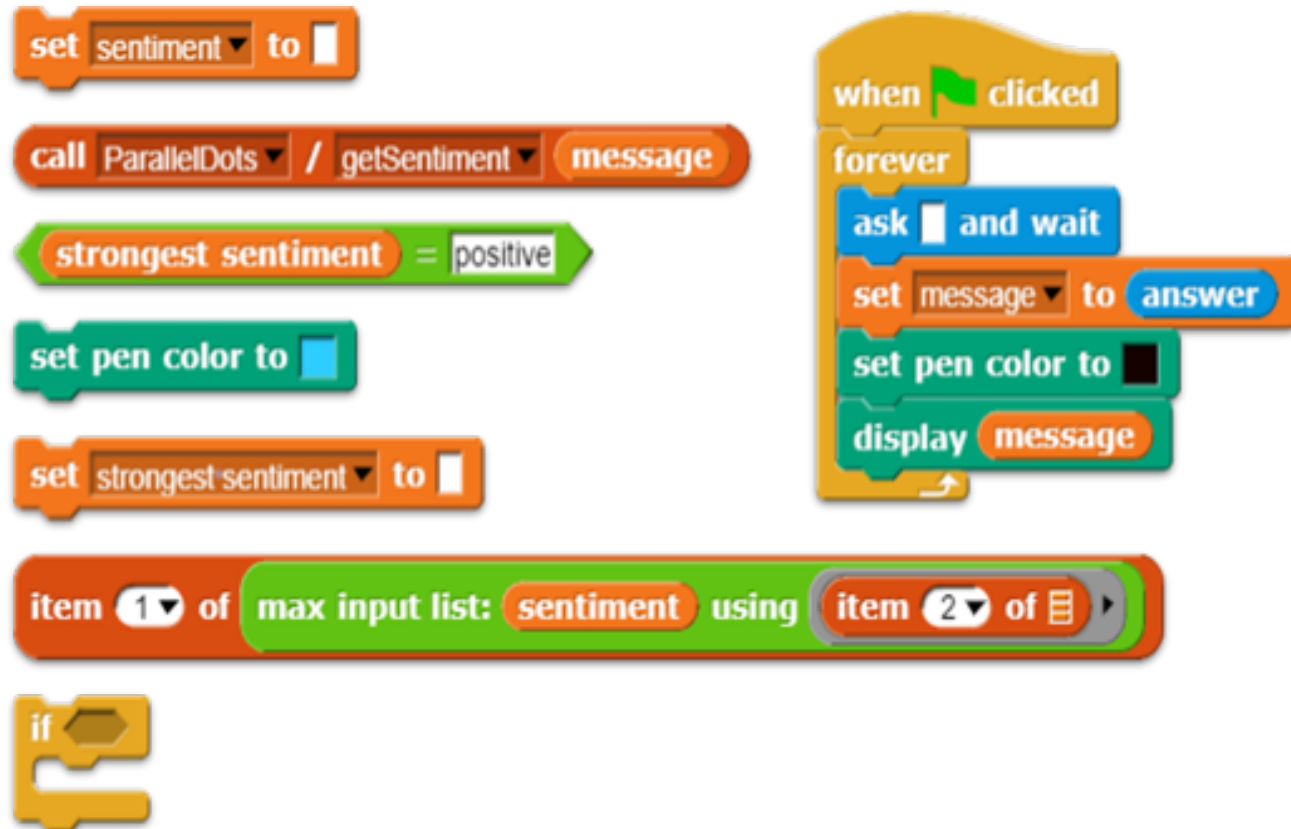
    Set pen color to neutral color
    If strongest sentiment is equal to positive
      Then set pen color to positive sentiment
    If sentiment sentiment is equal to negative
      Then set pen color to negative sentiment
    Display message to stage
  End Forever
```


Part 2 (NetsBlox Parson's Puzzle Activity)

Load up the Netsblox Activity :

Sentiment Writer [[link](#)]

Assemble the starter code provided to replicate the procedures in the pseudocode of Part 1.



LIVE 4m ago

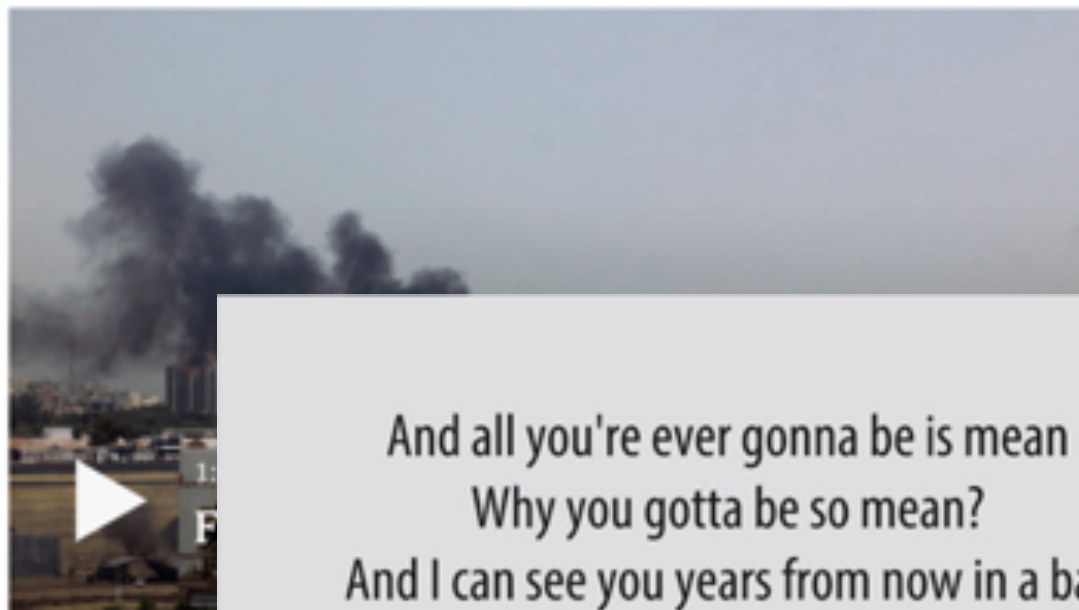
Rival Generals Duel for Power in Sudan, Dashing Hopes for Democracy

Mounting tensions broke into an open battle for control in Khartoum. At least 30 people were killed and about 400 were injured, U.N. officials said.

See more updates 

Who is battling for control in Sudan? Here's what to know about the clashes.

3 MIN READ



Leaked Pentagon Documents Who Is Jack Teixeira? What the Do

Analysis: The Pentagon Leaks Present New Twists in a Familiar Plot

There was little palpable alarm in Ukraine over what the leaked documents said about the war. Some welcomed the confirmation of the country's dire situation.

6 MIN READ



And all you're ever gonna be is mean
Why you gotta be so mean?
And I can see you years from now in a bar
Talking over a football game
With that same big loud opinion
But nobody's listening, washed up and ranting
About the same old bitter things
Drunk and grumbling on about how I can't sing
But all you are is mean
All you are is mean and a liar and pathetic





Can a neural network learn to recognize doodling?

Help teach it by adding your drawings to the [world's largest doodling data set](#), shared publicly to help with machine learning research.

Let's Draw!

<https://quickdraw.withgoogle.com/>

Quick, Draw! The data

Get the data Play the game

Now visualizing: apple

Randomize

You are looking at 139,898 apple drawings made by real people... on the internet.
If you see something that shouldn't be here, simply select the drawing and click the flag icon.

It will help us make the collection better for everyone.



Data Quick, Draw! has from your drawing

Information	Description
Key_id	Unique identifier for each drawing
word	Category user was told to draw
recognized	If the game recognized it
timestamp	When it was drawn
country_code	Which country it was drawn in
drawing	How you draw the image (in what order)

Dataset available here: <https://github.com/googlecreativitylabs/quickdraw-dataset>

This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

DISCUSSION

Is it OK for google to use your drawings as data? Why or why not?

- Did you realize, when you were playing with Quick, Draw!, that Google was keeping your images for use?
 - Can something have been done to make you clear-er on this?
- Does it help to know that the data is all anonymized, so there's no way to connect it directly to you?

This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

A Socially Relevant Focused AI Curriculum Designed for Female High School Students

[L Alvarez](#), [I Gransbury](#), [V Cateté](#), [T Barnes](#)... - ... on [Artificial Intelligence](#), 2022 - [ojs.aaai.org](#)

... For our **curriculum** framework, we wanted to implement an **AI** and ML module with the same goal. We believe introducing students to these **socially-relevant** topics early in their expo...

☆ Save [Cite](#) Cited by 11 [Related articles](#) [All 2 versions](#) [↗](#)

EAAI "AI Education in K-12" Conference 2022

Student Attitudes During the Pilot of the Computer Science Frontiers Course

..., [I Gransbury](#), [V Cateté](#), [T Barnes](#), [S Grover](#)... - [Proceedings of the ...](#), 2023 - [dl.acm.org](#)

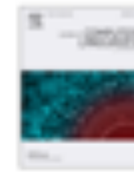
Motivation. We have created a modular project-based learning curriculum, Computer Science Frontiers (CSF) [1, 8], for secondary students in attempts to increase the persistence of ...

ACM SIGCSE 2023



Journal of Computer Languages

Volume 73, December 2022, 101156



Block-based abstractions and expansive services to make advanced computing concepts accessible to novices

[Corey Brady](#)[ⓐ], [Brian Broll](#)[ⓐ], [Gordon Stein](#)[ⓐ], [Devin Jean](#)[ⓐ], [Shuchi Grover](#)[ⓑ], [Veronica Coteté](#)[ⓒ], [Tiffany Barnes](#)[ⓒ], [Ákos Lédeczi](#)[ⓐ] [✉](#) [📄](#)

Journal paper on NetsBlox

Cybersecurity Education in the Age of AI: Integrating AI Learning into Cybersecurity High School Curricula



Shuchi Grover
(PI - Looking Glass Ventures)



Derek Babb
(UN-Omaha)



SaTC-Edu
#2113803
(EAGER)



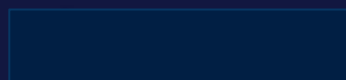
Brian Broll
(Co-PI - Vanderbilt)



Melissa Dark
(Advisor)

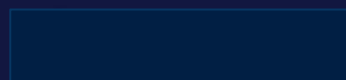
Since the start of the pandemic, the FBI has reported a

300%
Rise in
Cybercrime



Data breaches in the healthcare sector have

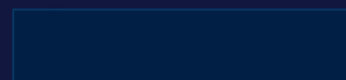
Risen by
58%



In just April 2020, Google clogged

Over 18 Mil
Malware

and phishing emails
linked to coronavirus daily.



Why Integrate Cybersecurity & AI learning?

- Both are fast-growing and increasingly important CS-related topics
- Leads to better understanding of how these fields look in the real world
 - Authentic (AI in Cybersecurity)
 - Learning in context (Cybersecurity in AI)
- Practicalities of introducing new subjects is challenging
 - Already packed CS and school curriculum
 - Easier for HS students to get a flavor of multiple emergent fields in one course than trying to introduce 2 separate courses

Cybersecurity & Artificial Intelligence

How & where do these 2 critical subfields of CS meet?

What does Cybersecurity look like in the age of AI?

How do AI models work to address issues of Cybersecurity?

How does AI raise (new) concerns related to Cybersecurity?

APPLICATIONS OF AI IN CYBERSECURITY

1. PASSWORD PROTECTION & AUTHENTICATION

AI is helping developers make biometric authentication even more accurate.



2. PHISHING DETECTION & PREVENTION CONTROL

AI & ML can be used to detect, track, react to & resolve phishing issues much more quickly than humans can.



3. VULNERABILITY MANAGEMENT

Systems based on AI & ML are proactive instead of reactive.



4. NETWORK SECURITY

AI is expediting the creation of security policies & determining organizations' network topographies.



5. BEHAVIORAL ANALYTICS

ML algorithms can learn & create a pattern of a user's behavior.



Learning Goals of ACT

Basics/Big Ideas of Cybersecurity

- Cybersecurity First Principles, The CIA Triad
- Cryptography, passwords, and ciphers, hacking
- Phishing, spamming
- Network security and vulnerabilities, DDOS attacks
- Misinformation campaigns, bullying, cyber warfare, bots

AI Ideas & ML Techniques

- Rule-based AI
- Decision Trees
- Classification & Prediction
- Supervised/ Unsupervised Learning
- Bias and Ethics
- Anomaly detection
- Adversarial examples
- Generative adversarial neural network (Deep fakes)

AI in context of Cybersecurity

- AI/ML concepts/techniques in Cybersecurity contexts
- Exploring various ML techniques & models in **NetsBlox** using code & (often actual) data
- Discuss use & implications for Cybersecurity and society
- Explore/play with pre-built games & examples

ACT Pedagogical Approach

- **Learning in Context:** Teaching AI *in context of* Cybersecurity
- **Making Connections:** Helping students see real-world connections through relatable examples/hooks and "cases"/"scenarios"
- **Games and guided exploration for engagement and for building intuitions of ML techniques**: Hands on unplugged, digital, and programming activities, e.g.,
 - <https://kennysong.github.io/adversarial.js/>, Google TensorFlow Playground, ThisXDoesNot Exist,
 - Pre-programmed games (Find the minimum, Registration Bot, ...)
- **Scaffolding:** complex ideas through various techniques including
 - **"Glass-boxing" / Levels of Abstraction:** Increasing the transparency on how the model is programmed - idea/unplugged exploration, pseudocode, "sub-goal" blocks, Parson's problems, actual code
- **Sense-making:** Pre-activity topical news/issues; post-activity reflection/discussion
- **Collaboration:** All activities are designed to be completed in pairs/small groups

Big Ideas / Goals Related to AI/ML

- Lift the hood on Machine Learning techniques
 - how the machine *learns*
 - how to examine/understand data and its features
- Optimization
- Generalization & Overfitting/Underfitting
- How bias can impact aspects (and phases) of machine learning
 - a. Issues of Ethics
- Adversarial Thinking: Whenever we discuss solving a cyber detection issue using AI, also *think of how it can be fooled*

Curricular Activities & AI/ML Concepts/Big Ideas

- Curricular Coding Activities in NetsBlox:
 - Rule-Based AI to Mitigate Denial of Service Attacks (through a chat room kids code)
 - Text Classification and Cyberbullying (using the **ParallelDots NLP API**)
 - Twitter Bot Classification and Decision Tree Building
 - Optimization & Find the Minimum Game
 - Registration Bot Detection with Gradient Descent
 - Generative Adversarial Models
- Additional (Recurring) Topics:
 - Adversarial Thinking
 - Generalization and Over/Underfitting
 - Understanding Bias and Critical Interrogation of the Impacts of ML Models
- Unplugged/Interactive (non-programming) activities before coding

```
else
  ask what is your name? and wait
  set myusername to answer
  set serverid to answer
  when space key pressed
    ask what would you like to say? and wait
    send msg chat my username answer to server

when I receive chat username message
  set category to
  item 1 of
  max input list: call ParallelDots / getAbuse message using item 2 of
  if not category = neither
    set message to join message {(censor this!)}
  if length of messages > 10
  add username message to messages
  display messages messages
```

```
brian: hello
brian: you suck! (censor this!)
brian: You are wonderful!
category neither

Waiting for messages...
```

Moderating Speech in NetsBlox Chat App Using ParallelDots

Scaffolded Progression to Understand Decision Trees Through Twitter Bot Classification

1.

Data exploration in CODAP; Examine features; Ways to split the dataset

2.

Scaffolded activity to manually build a rule-based classifier in NetsBlox & evaluate on the entire dataset

3.

DT Classifier: Understand the algorithm for automating data for “best split” ie lowest entropy (Pseudocode)

4.

DT Classifier: Explore the code for automating data for “best split” ie lowest entropy (Parson’s Puzzle)



Make a variable

Delete a variable

if Correct

Bots

Clean Twitter Data

Data Features

NotBots

Percent Correct

subset

Twitter Account Data

Twitter Data

set to

change by

show variable

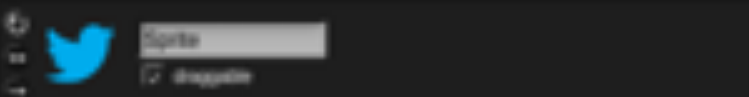
hide variable

script variables

set

numbers from to

to



Scripts Costumes Sounds Room

Import Twitter Data into CODAP with column names:
 item 1 of Twitter Account Data

set NotBots to keep items item 1 of

Import NotBots into CODAP with column names:

set Bots to keep items item 1 of

Import Bots into CODAP with column names: item 1 of

set subset to CODAP selection



In this activity, we will be trying to classify Twitter accounts as bots or not using a synthetic dataset. First, we will create variables for the account features and labels from the original table.

set labels to map item last of over Twitter Accounts

set account-features to map all but last of over Twitter Accounts

Next, we will import the data into CODAP so we can easily explore the dataset!

import Twitter Accounts into CODAP with column names: Twitter Account Fields

show CODAP

Can you write code that will classify the accounts correctly? Edit the "classify account" block below to give it a try!

This first data point should be classified as a bot.

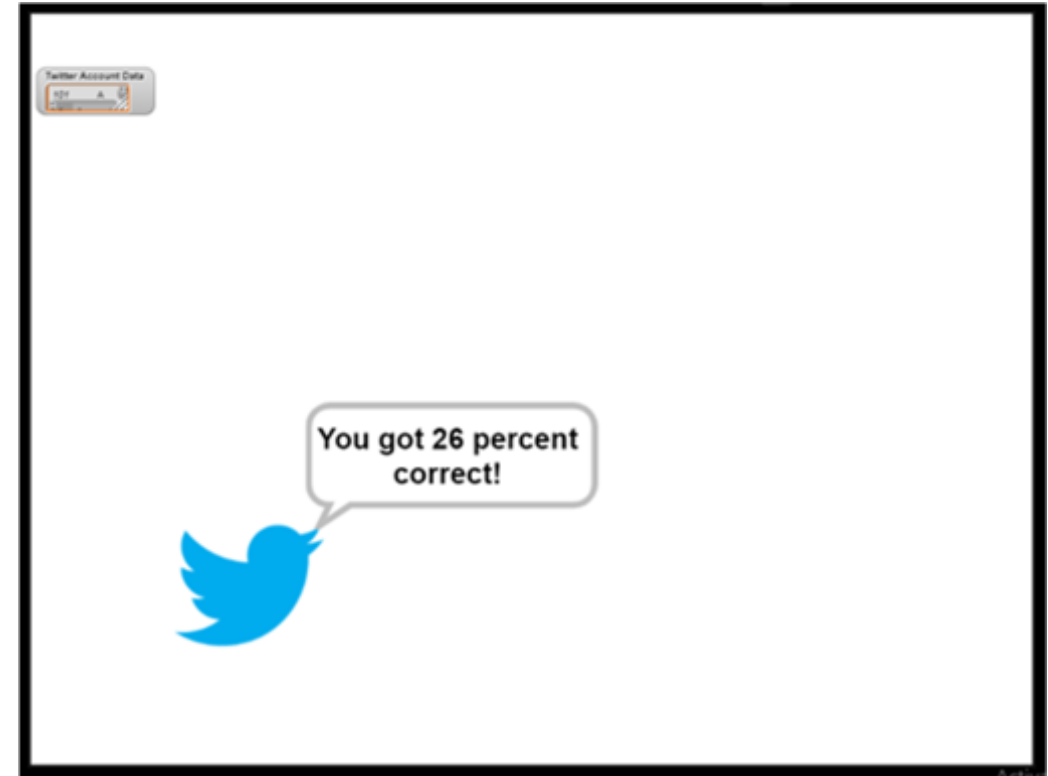
classify account item 1 of account features item 1 of labels

Use the following script to evaluate your classification block on the entire dataset!

The script will return a number between 0 and 1 representing the fraction of correct labels. 0 means every point was incorrect and 1 means the model had a perfect score!

```
script variables features label  
set incorrect to list  
for each account in Twitter Accounts  
  set label to item last of account  
  set features to all but last of account  
  if not classify account features = label  
    add account to incorrect  
import incorrect into CODAP with column names: Twitter Account Fields  
report 1 - length of incorrect / length of Twitter Accounts
```

Twitter Bot Detection Using a Rule-Based Classifier



Students create a classifier to classify twitter accounts as Bot or Not

Decision Tree Building Algorithm

set *possible branches* to all ways to split the data

for each *possible branch*:

 set *improvement* to the decrease in entropy using this branch

 if *improvement* > *best improvement* (seen so far):

 Save the (best) *improvement*, *branch*, *data partitions*

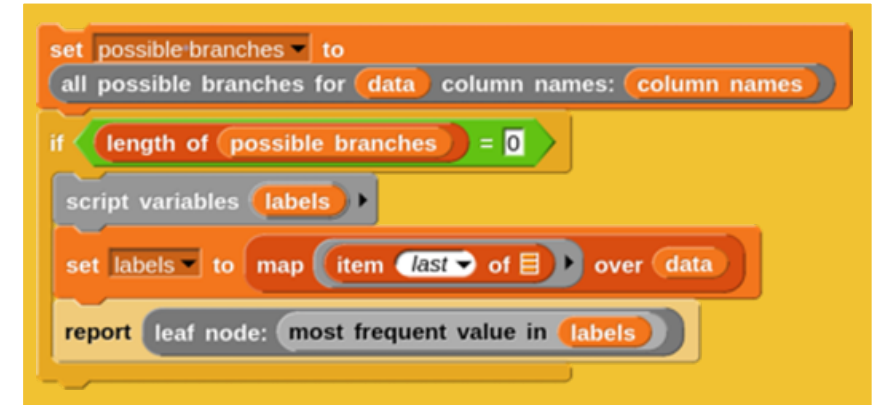
for each data partition created with the best branch:

 if the points have different labels:

 create a new tree from the partition

 else:

 create a leaf node



+ build + decision + tree + from + data : + with + column + names: + column names : +

script variables

possible branches data partitions entropy lowest entropy best branch

best partitioned data ◀ ▶

warp

Can you use the following blocks to create the high-level algorithm in the warp block on the left?

consider all possible branches to split the data

find the branch that best splits the data

build decision tree for each side of the branch

report best branch

The implementations for each of the above grey blocks can be found below. Can you put the implementations inside the correct grey block above?

The implementations for each of the above grey blocks can be found below. Can you put the implementations inside the correct grey block above?

set lowest entropy to 1

for each branch in possible branches

set data partitions to split data with branch: branch

set entropy to average entropy input list: data partitions

if entropy < lowest entropy

set lowest entropy to entropy

set best branch to branch

set best partitioned data to data partitions

set possible branches to

all possible branches for data column names: column names

script variables branch sides

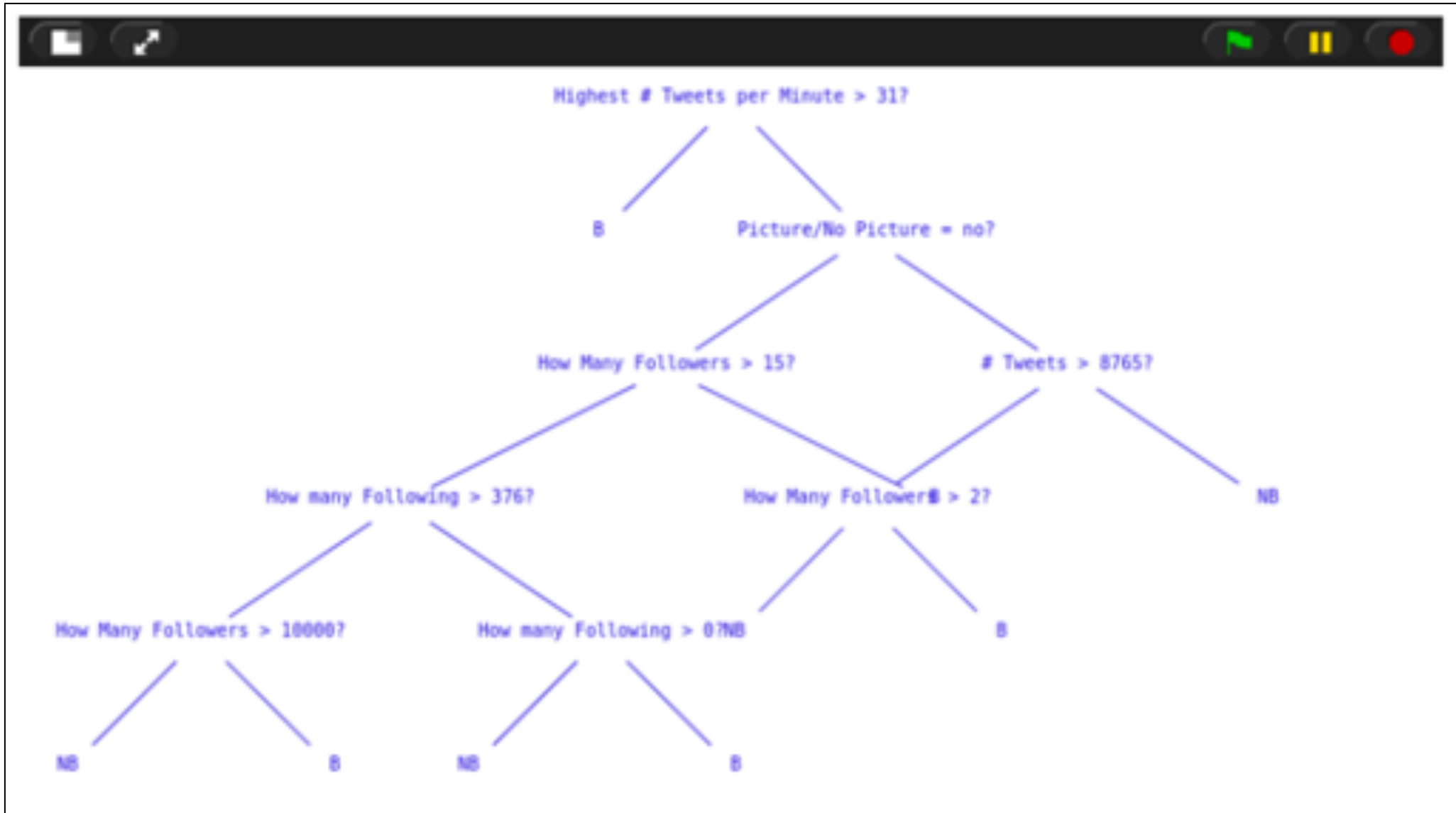
set branch sides to

do build decision tree from with column names: column names for each best partitioned data

set sides of branch: best branch to branch sides

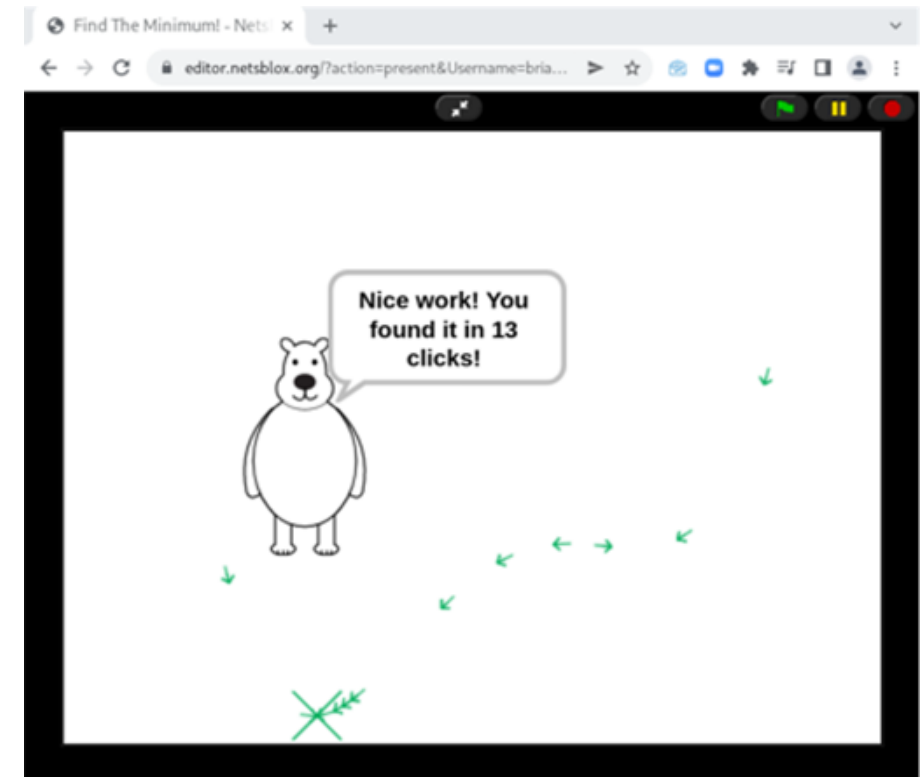
Nice work making it this far! There are two more

Adversarial Thinking: How Can The Model Be Fooled?



Example Sequence: Optimization using Gradient Descent & Registration Bot Classification

- Introduced using the “Find the Minimum” game
- The student is trying to find the minimum of an invisible function
- When the user clicks on the screen, the point of the function is shown to the user along with the gradient
- Functions increase in complexity as game progresses
- Discussion topics following the activity include:
 - What would the minimum represent if the function was concert ticket prices over time? Or something else?
 - What if it was the error of a machine learning model?



Example Sequence: Optimization + Bot Classification

- Given data collected from humans and bots while registering for a website, can we train a model to detect bots?
- Explore the data in CODAP
- Discuss a simple way we could try to predict if it is a bot (logistic regression)
- Can we “tweak” the unknowns (weights) of the model like in the “Find the Minimum” game?

Putting it all together!

Repeat many times:

Determine what to tweak/optimize (enable gradients for them)

For each data point:

make a prediction

multiply the features by their weights (and sum them)

compute the sigmoid of the above value

compute the error (loss) of the prediction

tweak the weights a small amount using the gradients



Now, can we complete the training algorithm?

First, let's plan out the high-level goals in the training algorithm by adding the blocks on the right into the training script on the left!

script variables many times

set many times to 100

repeat many times

compute error

tweak variables (w/ gradients enabled)

make prediction

enable gradients for variables to tweak

warp

for i = 1 to length of data

BLOG@CACM

AI Education for Teens: Lifting the Hood on Data and Machine Learning

By Shuchi Grover and Brian Broll
November 23, 2021
[Comments](#)

[MORE NEWS & OPINIONS](#)
[Is A.I. the Future of Test Prep?](#)

<https://cacm.acm.org/blogs/blog-cacm/256999-ai-education-for-teens-lifting-the-hood-on-data-and-machine-learning/>

Beyond Black-Boxes: Teaching Complex Machine Learning Ideas through Scaffolded Interactive Activities

Brian Broll¹ and Shuchi Grover²

¹Vanderbilt University

²Looking Glass Ventures

brian.broll@vanderbilt.edu, shuchig@cs.stanford.edu

EAAI "AI Education in K-12" Conference 2023

Cybersecurity Education in the Age of AI: Integrating AI Learning into Cybersecurity High School Curricula

Shuchi Grover

Looking Glass Ventures

Austin, TX, USA

shuchig@cs.stanford.edu

Brian Broll

Vanderbilt University

Nashville, TN, USA

brian.broll@vanderbilt.edu

Derek Babb

University of Nebraska, Omaha

Omaha, NE, USA

dvbabb@unomaha.edu

ACM SIGCSE 2023

CS Frontiers

- Website: csfrontiers.org
- Curriculum: csfrontiers.org/curriculum
- [Intro to CSF & Distributed Computing video](#)
- [AI/ML Module](#)

AI & Cybersecurity for Teens (ACT)

- Website: cyberai4k12.org
- Curriculum: <https://cyberai4k12.github.io/curriculum/>
- Resources: <https://github.com/cyberai4k12/awesome-cyberai4k12>

Key Emergent Themes & Outcomes

What we should be teaching in schools about AI

Data Primacy/Agency & 'CT 2.0'

Importance of ethics, bias, and critical examination of AI


Pedagogies & instructional approaches

Ways to integrate AI into other subjects

Curricular co-design processes & teacher preparation

+ A plethora of free curricular resources

Plethora of freely available activities, interactives, tools, curricula,...
(See the growing resource list on AI4k12.org)



FREE IS GOOD!



Challenges & Tensions

My Recommendations for K-12 AI Ed

Build on lessons from K-12 CS Education (pedagogies, equity, teacher prep, ...)

Integrate into/with core school subjects



Dovetail CS Education & AI Education

Focus on building an empirical base for age-appropriate progressions and pedagogies.



Why K-12 AI Ed? (aka AI & the coming tsunami!)

State of AI education in K-12

Lessons from K-12 CS Ed

Challenges, open issues, & recommendations

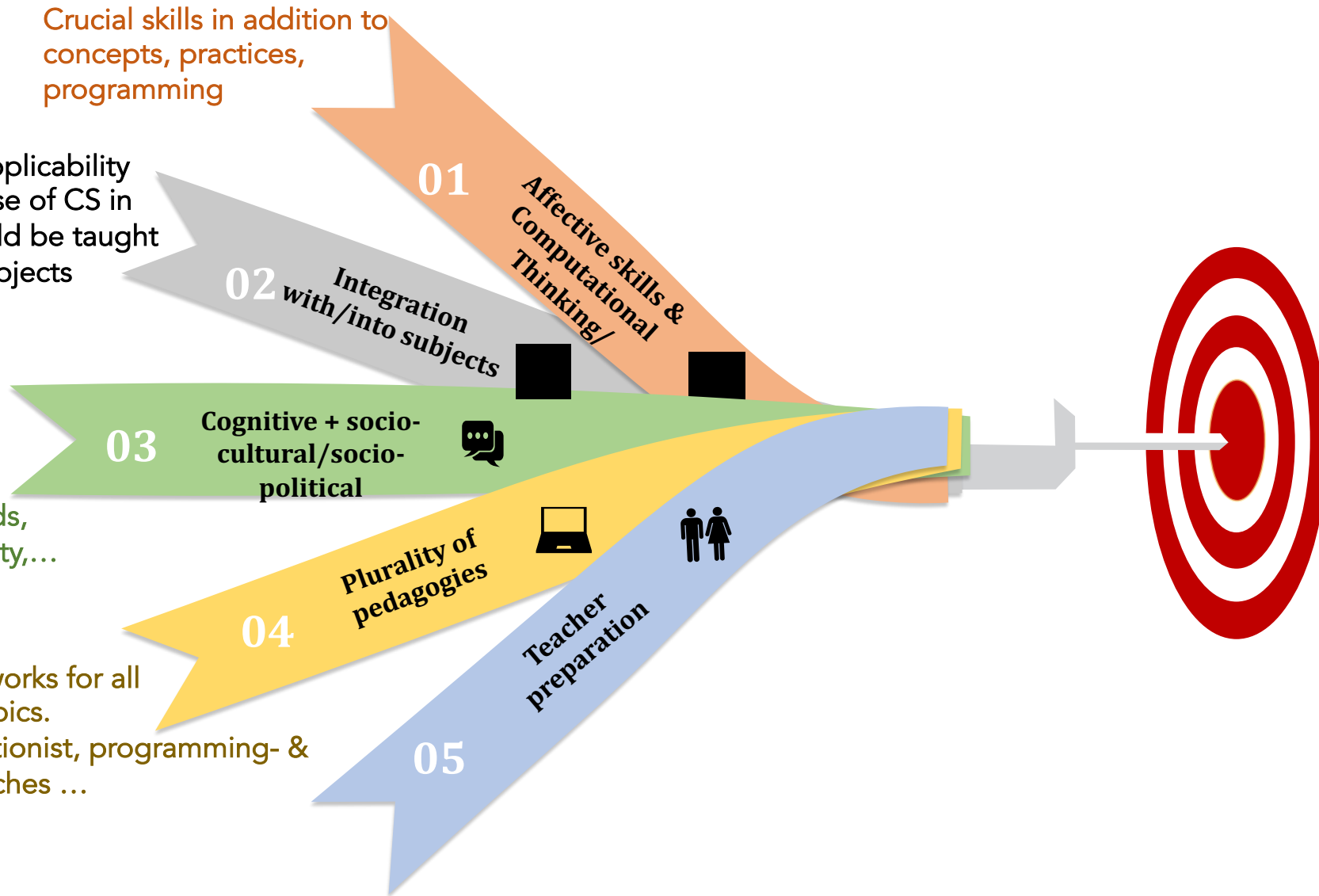
Crucial skills in addition to concepts, practices, programming

For students to see applicability and understand the use of CS in the real-world, it should be taught in conjunction with subjects

Attend to conceptual learning through the cognitive & affective--leveraging backgrounds, connections, community,...

No single approach works for all contexts, learners, topics. Unplugged, constructionist, programming- & game-based, approaches ...

No effort to introduce CS can succeed without preparing teachers; co-designing; building on their teaching experiences



**COMPUTER
SCIENCE
FOR ALL**



Computer Science in K-12

An A to Z Handbook on teaching programming

[Home](#)[Book ▾](#)[Resources ▾](#)[Testimonials](#)[Contact](#)[BUY NOW](#)csa2z.com

COMPUTER SCIENCE IN K-12

An **A to Z** handbook on teaching programming



Contributions by Leading Computer Science Educators and Researchers

Edited
by **SHUCHI GROVER**

Computer Science in K-12

An A to Z handbook on teaching programming

Coding teaches our students the essence of logical thinking and problem solving while also preparing them for a world in which computing is becoming increasingly pervasive. While there's excitement and enthusiasm about programming becoming an intrinsic part of K-12 curricula the world over, there's also growing anxiety about preparing teachers to teach effectively at all grade levels.

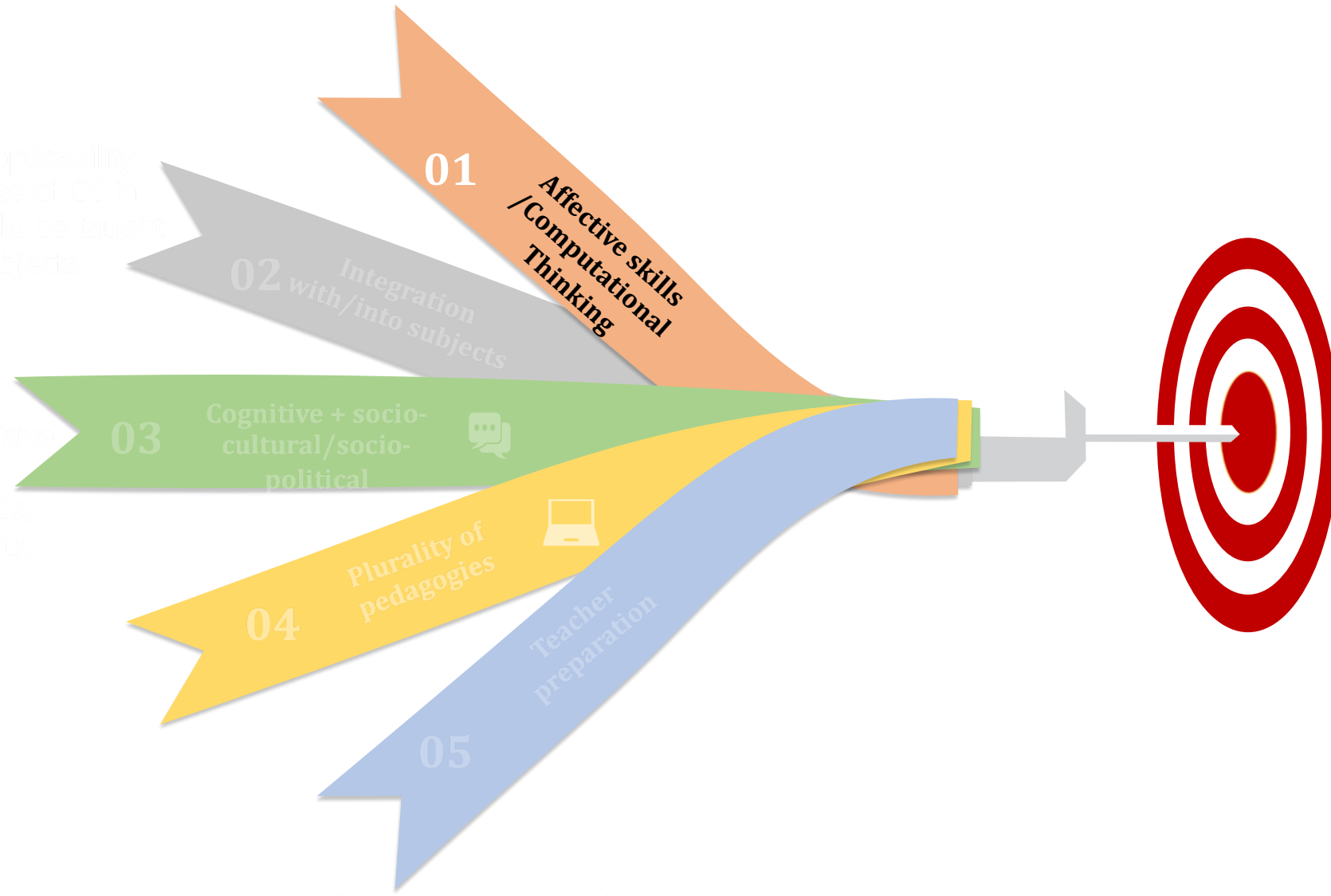
This book strives to be an enduring, practical guide for every K-12 CS teacher anywhere who is either teaching or planning to teach programming at any level. To this end, readers will discover:

- An A-to-Z organization that affords comprehensive insight into teaching introductory programming
- 26 chapters that cover foundational concepts, practices and well-researched pedagogies related to teaching introductory programming as an integral part of K-12 computer science. Cumulatively these chapters address the two allent building blocks of effective teaching of introductory programming – **what content to teach (concepts and practices)** and **how to teach (pedagogy)**
- Concrete ideas and rich grade-appropriate examples inspired by practice and research for classroom use.
- Perspectives and experiences shared by educators and scholars who are actively practicing and/or examining the teaching of computer science and programming in K-12 classrooms.

For students to see applicability and understand the use of CS in the real world, it should be taught in conjunction with subjects

Attend to conceptual learning through cognitive (non-cognitive- leveraging bed-grounds, connections, communities, ...)

No effort to introduce CS can succeed without preparing teachers; so designing; building on their teaching experiences





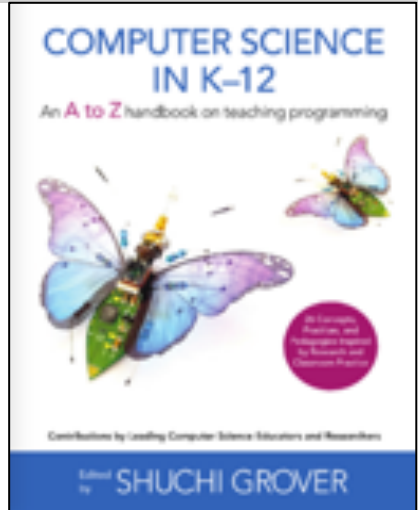
The Concepts and Practices of the K–12 Computer Science Framework

Core Concepts

1. Computing Systems
2. Networks and the Internet
3. Data and Analysis
4. Algorithms and Programming
5. Impacts of Computing

Core Practices

1. Fostering an Inclusive Computing Culture
2. Collaborating Around Computing
3. Recognizing and Defining Computational Problems
4. Developing and Using Abstractions
5. Creating Computational Artifacts
6. Testing and Refining Computational Artifacts
7. Communicating About Computing



Knowledge, Skills, Attitudes, and Beliefs: Learning Goals for Introductory Programming

Rebecca Vivian, Shuchi Grover & Katrina Falkner

CHAPTER 11

Educating the mind without educating the heart is no education at all
- Aristotle

INTRODUCTION

Learning computer programming can be a nuanced experience. It can empower learners as much as it can challenge their self-beliefs and their perceptions of computing. A key goal of 21st century education is to help learners build not just knowledge about new concepts, skills, and practices but also ways of thinking, interests and identities, and beliefs and skills to succeed in this connected, technology-infused world. This is true of teaching computing as well.

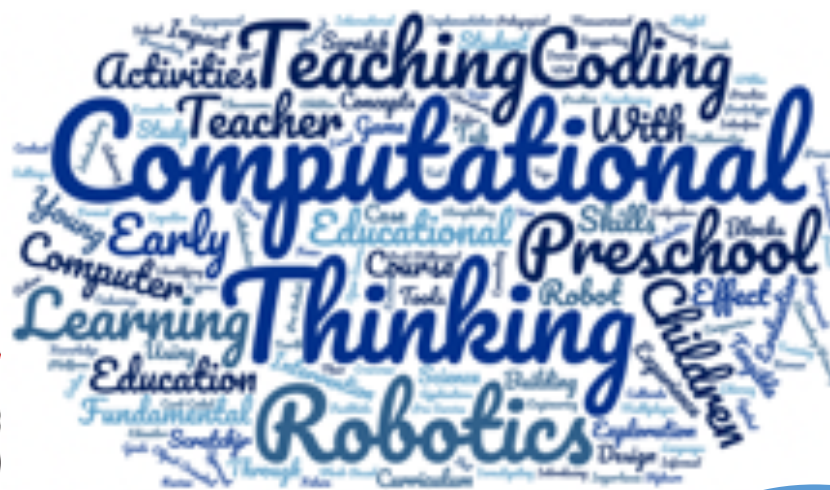
This chapter outlines key knowledge and skills as well as positive attitudes, beliefs, and mindsets related to learning programming. We also unpack practical learning and teaching strategies. The overarching goal is to build teacher capacity and confidence that can support students in flourishing as confident learners with increased interest and agency in the learning process. Ensuring students have the necessary problem-solving and self-regulatory learning skills to design, build, and test programming solutions supports the development of resilient learners who also recognize that taking



Figure 1. Deeper learning calls for attending to cognitive, interpersonal, and intrapersonal skills to prepare K-12 students for college and beyond.

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 in K–12: A Review of the State of the Field

Shuchi Grover¹ and Roy Pea²

Jeannette Wing's influential article on computational thinking 6 years ago argued for adding this new competency to every child's analytical ability as a vital ingredient of STEM learning. What is computational thinking? Why did this article resonate with so many and serve as a rallying cry for educators, education researchers, and policy makers?

of the fastest-growing job markets through 2018. This CS imperative has dovetailed with the science policy attention to STEM learning in the United States since the turn of the 21st century. With CT being viewed as at the core of all modern science, technology, engineering, and mathematics (STEM) disciplines (Henderson, Cortina, Hazzan, & Wing, 2007) it appears that

DOI:10.1145/3265747

Mike Tissenbaum, Josh Sheldon, and Hal Abelson

Viewpoint

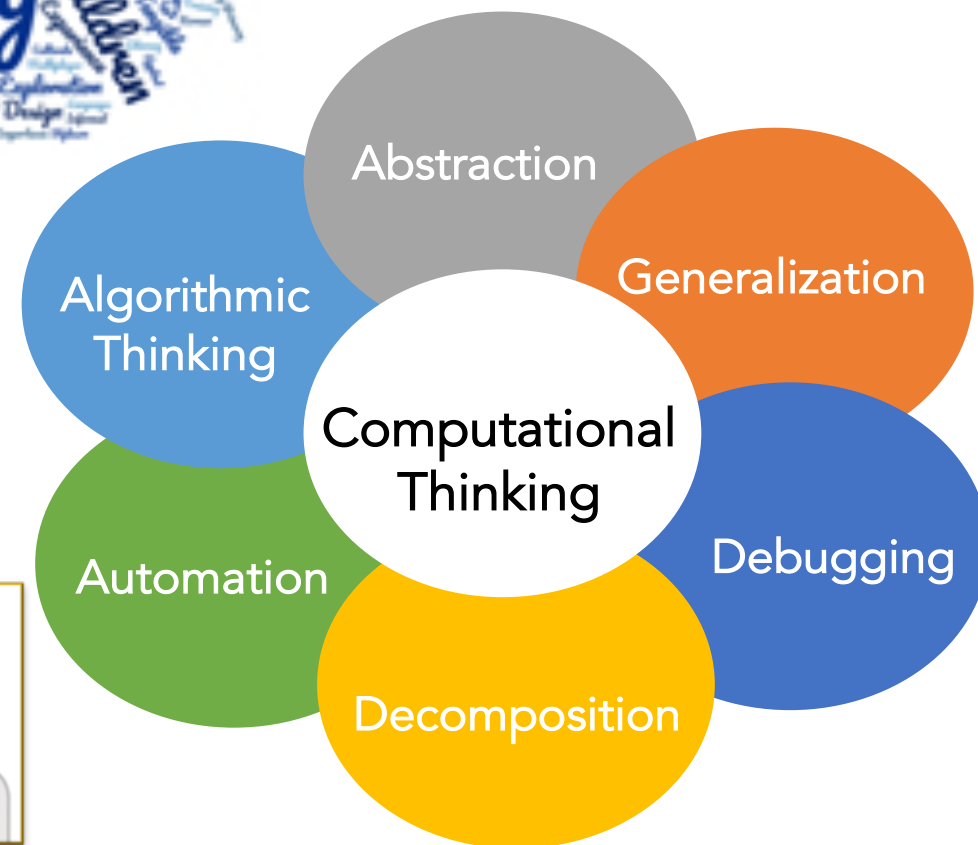
From Computational Thinking to Computational Action

Envisioning computing education that both teaches and empowers.

A Reevaluation of Computational Thinking in K–12 Education: Moving Toward Computational Literacies

Yasmin B. Kafai¹ and Chris Proctor²

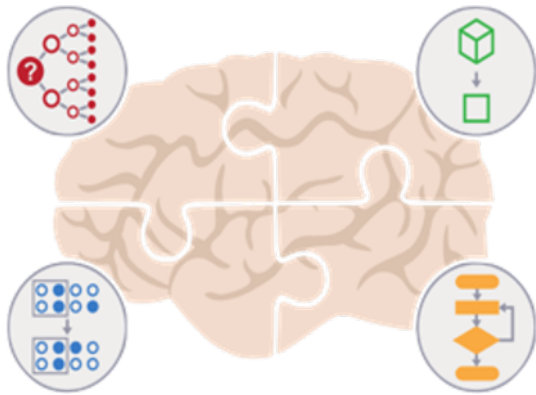
Over the past decade, initiatives around the world have introduced computing into K–12 education under the umbrella of computational thinking. While initial implementations focused on skills and knowledge for college and career readiness,



The practice of CS involves learning of problem-solving skills (in addition to core disciplinary topics comprising CS) that transcend programming languages and help learners engage with and apply computing in multiple subjects, domains, and situations

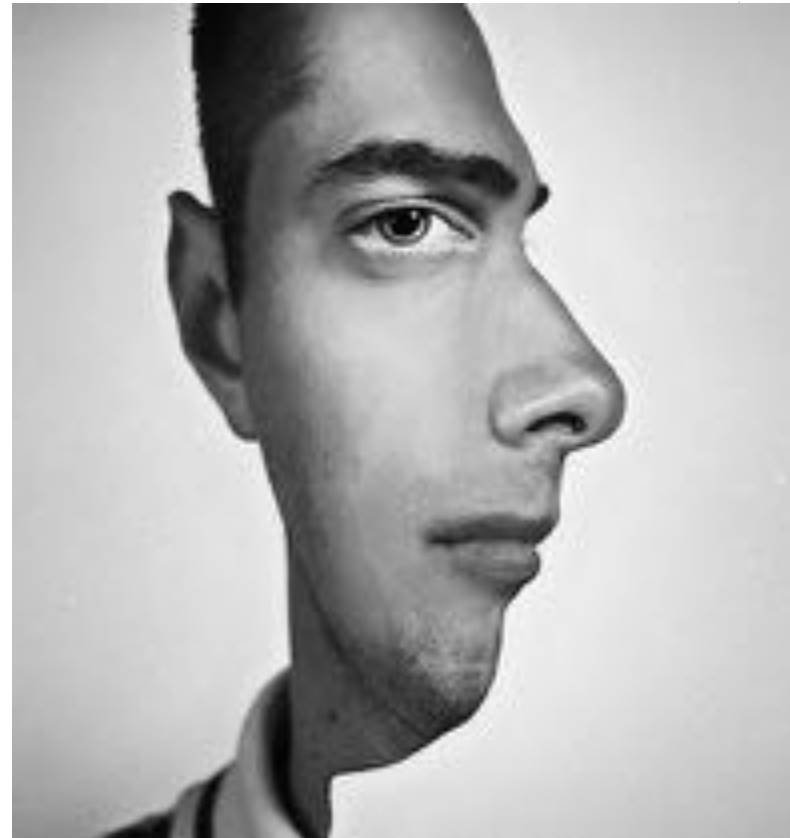
A Tale of Two CTs (and a Revised Timeline for Computational Thinking)

By Shuchi Grover
November 5, 2018



CT for CS

Computational Thinking (or CS Thinking) as a disciplinary thinking / problem-solving skill crucial for deeper learning of computer science

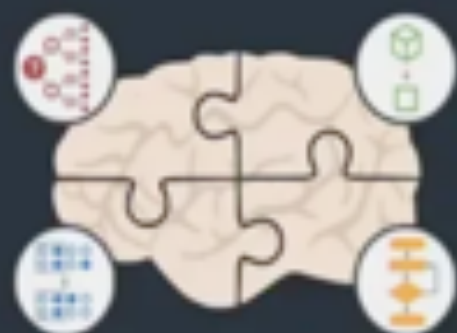


Source: <https://www.moillusions.com/two-face-optical-illusion/>



CT for Integration

Computational Thinking and programming as a tool or means to innovate in or enrich learning in other disciplines



CT for CS

Computer Science Thinking as a **disciplinary thinking / problem-solving skill** crucial for **deeper learning of computer science**

Learning to code & all CT practices

Coding+unplugged activities

Learning CT is one end/goal



Source: <https://www.moflusions.com/two-face-optical-illusion/>



CT for Integration

Computational Thinking and programming as a **tool/vehicle** to innovate in or enrich **learning in other disciplines**

Coding to learn



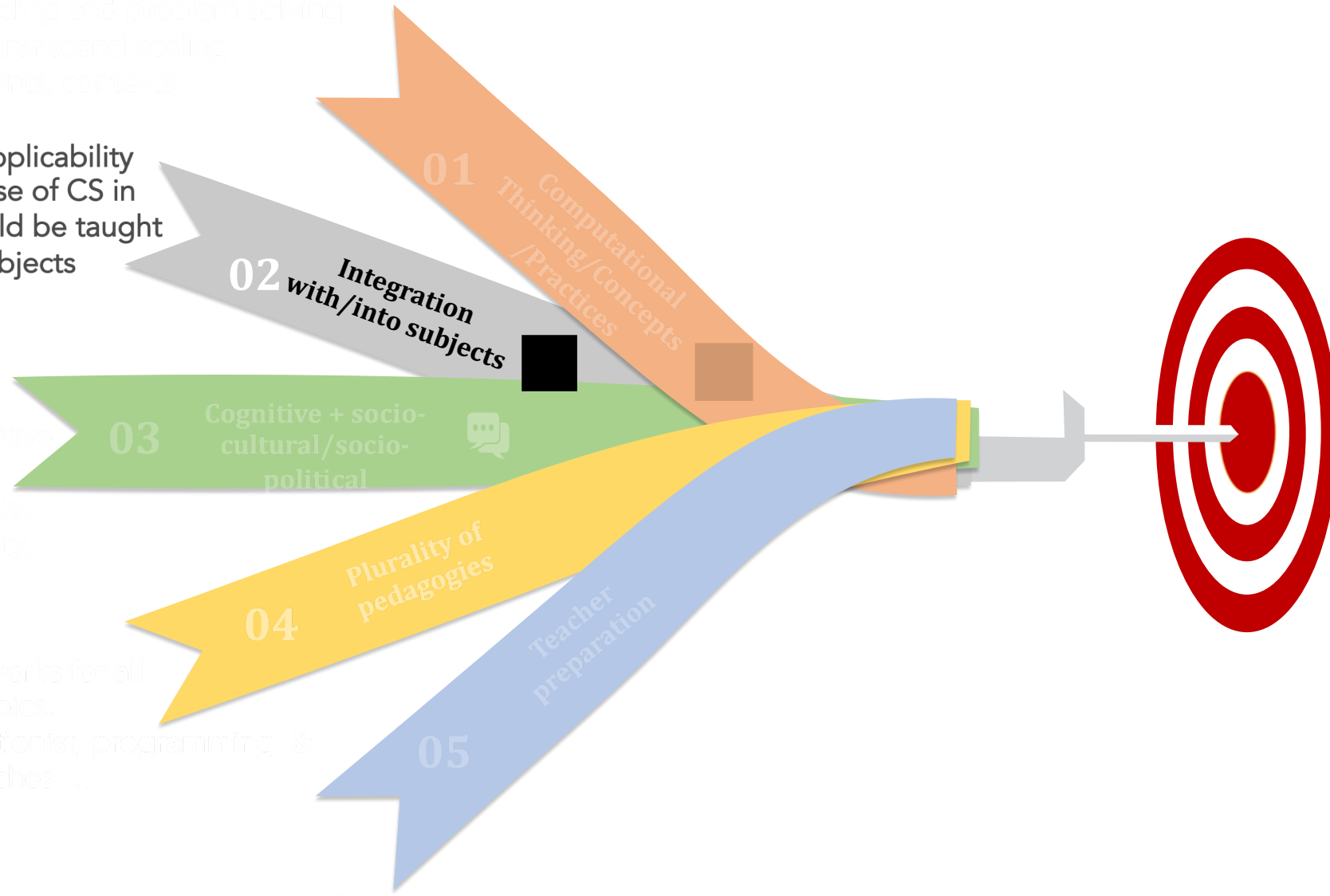
Clarity on what to teach, build enduring understanding and problem solving skills that transcend coding, environments, contexts

For students to see applicability and understand the use of CS in the real-world, it should be taught in conjunction with subjects

Accord to conceptual learning through cognitive / non-cognitive - leveraging bed-grounds, connections, communities, ...

No single approach works for all contexts, learners, topics. Unplugged, constructionist, programming- & game-based, approaches ...

No effort to introduce CS can succeed without preparing teachers, so designing, building on their teaching experiences

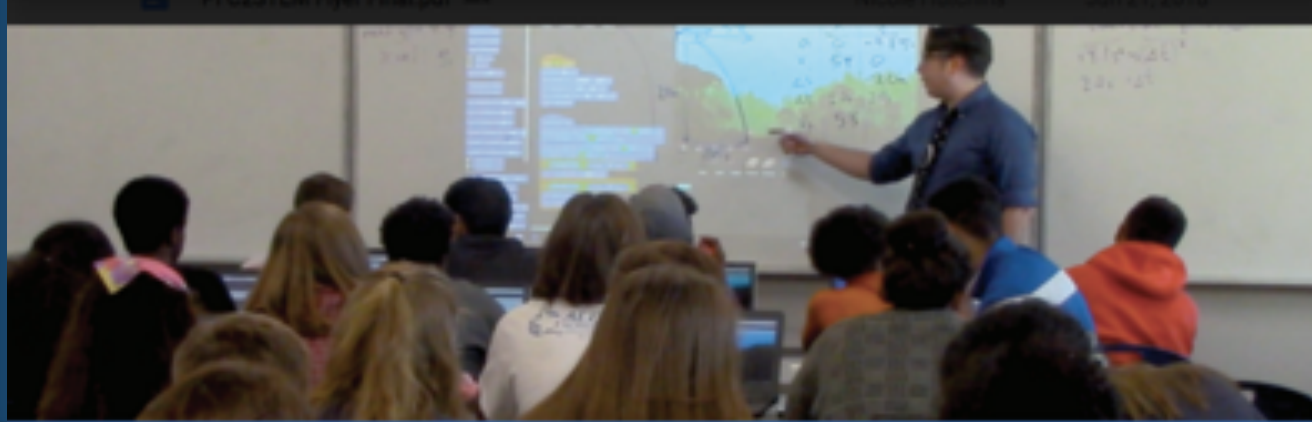


Computational Thinking
from K-12 Disciplinary
Perspectives - 2 Workshop
series (NSF)



ICLS2020 Symposium:
Integrating STEM and Computing
in PK-12: Operationalizing
Computational Thinking for
STEM Learning and Assessment
(Grover et al., 2020)

Special Issue of Journal of
Science and Technology
Education focused on
integration of CT In STEM
Disciplines
(Lee, Grover, Martin, Pillai,
& Malyn-Smith, 2020)



PISA 2021 Mathematics
Framework Expert Group
(inclusion of some CT items)

VELA: Exploratory Activities
for Computational Concepts
in Intro Programming

Synergistic Learning of
Physics/Biology & CT
through computational
model building in C2STEM

Integrating CT into Science
& Math activities for pre-K
learners in formal/informal
settings

CSFrontiers: Engaging
Female High School
Students in New Frontiers
of Computing

Computational Thinking
from K-12 Disciplinary
Perspectives - 2 Workshop
series (NSF)

Special Issue of Journal of
Science and Technology
Education focused on
integration of CT In STEM
Disciplines
(Lee, Grover, Martin, Pillai,
& Malyn-Smith, 2020)

VELA: Exploratory Activities
for Computational Concepts
in Intro Programming

Synergistic Learning of
Physics & CT through
computational model
building in C2STEM

Integrating CT into
Science & Math activities
for preK learners in
formal/informal settings

CSFrontiers: Engaging
Female High School
Students in New Frontiers
of Computing



ICLS2020 Symposium:
Integrating STEM and Computing
in PK-12: Operationalizing
Computational Thinking for
STEM Learning and Assessment
(Grover et al., 2020)

PISA 2021 Mathematics
Framework Expert Group
(inclusion of some CT items)

Computational Thinking
from K-12 Disciplinary
Perspectives - 2 Workshop
series (NSF)



ICLS2020 Symposium:
Integrating STEM and Computing
in PK-12: Operationalizing
Computational Thinking for
STEM Learning and Assessment
(Grover et al., 2020)

PISA 2021 Mathematics
Framework Expert Group
(inclusion of some CT items)

Special Issue of Journal of
Science and Technology
Education focused on
integration of CT In STEM
Disciplines
(Lee, Grover, Martin, Pillai,
& Malyn-Smith, 2020)

VELA: Exploratory Activities
for Computational Concepts
in Intro Programming

Synergistic Learning of
Physics & CT through
computational model
building in C2STEM

Integrating CT into Science
& Math activities for pre-K
learners in formal/informal
settings

CSFrontiers: Engaging
High School Girls in New
Frontiers of Computing

Computational Thinking
from K-12 Disciplinary
Perspectives - 2 Workshop
series (NSF)



ICLS2020 Symposium:
Integrating STEM and Computing
in PK-12: Operationalizing
Computational Thinking for
STEM Learning and Assessment
(Grover et al., 2020)

PISA 2021 Mathematics
Framework Expert Group
(inclusion of some CT items)

Special Issue of Journal of
Science and Technology
Education focused on
integration of CT In STEM
Disciplines
(Lee, Grover, Martin, Pillai,
& Malyn-Smith, 2020)

VELA: Exploratory
Activities for
Computational Concepts
in Intro Programming

Synergistic Learning of
Physics & CT through
computational model
building in C2STEM

Integrating CT into Science
& Math activities for pre-K
learners in formal/informal
settings

CSFrontiers: Engaging
Female High School
Students in New Frontiers
of Computing

COMPUTER SCIENCE IN K-12

An **A to Z** handbook on teaching programming



Contributions by Leading Computer Science Educators and Researchers

Edited
by SHUCHI GROVER

<https://csa2z.com/>

Integrating Programming Into Other Subjects

Shuchi Grover & Aman Yadav

CHAPTER
9

INTRODUCTION

The goal of teaching programming in K–12 school years is to introduce learners to a fundamental skill for the times we live in. Algorithms are touted to be the “fourth R” alongside Reading, writing, and arithmetic. However, a computer science classroom is not the only space to learn coding. Many subjects offer students engaging ways to learn programming in the context of those disciplines. In fact, some believe that teaching programming within a science or a social studies class enhances the learning experience because of the concrete contexts in which it can be demonstrated. Furthermore, problem solving along with coding offers the opportunity to develop a key 21st-century skill, **computational thinking (CT)**—a composite set of thinking and problem-solving skills closely related to, and learned through, coding. This chapter discusses techniques for integrating CT and coding in lessons, and presents vignettes from elementary and secondary classrooms that teachers can draw inspiration from, to bring CS to their students.

My basic idea is that programming is the most powerful medium of developing the sophisticated and rigorous thinking needed for mathematics, for grammar, for physics, for statistics, for all the hard subjects...
— Seymour Papert

Integrating Programming Into Other Subjects

Shuchi Grover & Aman Yadav

CHAPTER 9

INTRODUCTION

The goal of teaching programming in K–12 school years is to introduce learners to a fundamental skill for the times we live in. Algorithms are touted to be the “fourth R” alongside Reading, Writing, and Arithmetic. However, a computer science classroom is not the only space to learn coding. Many subjects offer students engaging ways to learn programming in the context of those disciplines. In fact, some believe that teaching programming within a science or a social studies class enhances the learning experience because of the concrete contexts in which it can be demonstrated. Furthermore, problem solving along with coding offers the opportunity to develop a key 21st-century skill, **computational thinking (CT)**—a composite set of thinking and problem-solving skills closely related to, and learned through, coding. This chapter discusses techniques for integrating CT and coding in lessons, and presents vignettes from elementary and secondary classrooms that teachers can draw inspiration from, to bring CS to their students.

My best idea is that programming is the most powerful medium of developing the algorithmic and logical thinking needed for mathematics, for physics, for statistics, for all the hard subjects...
—Seymour Papert

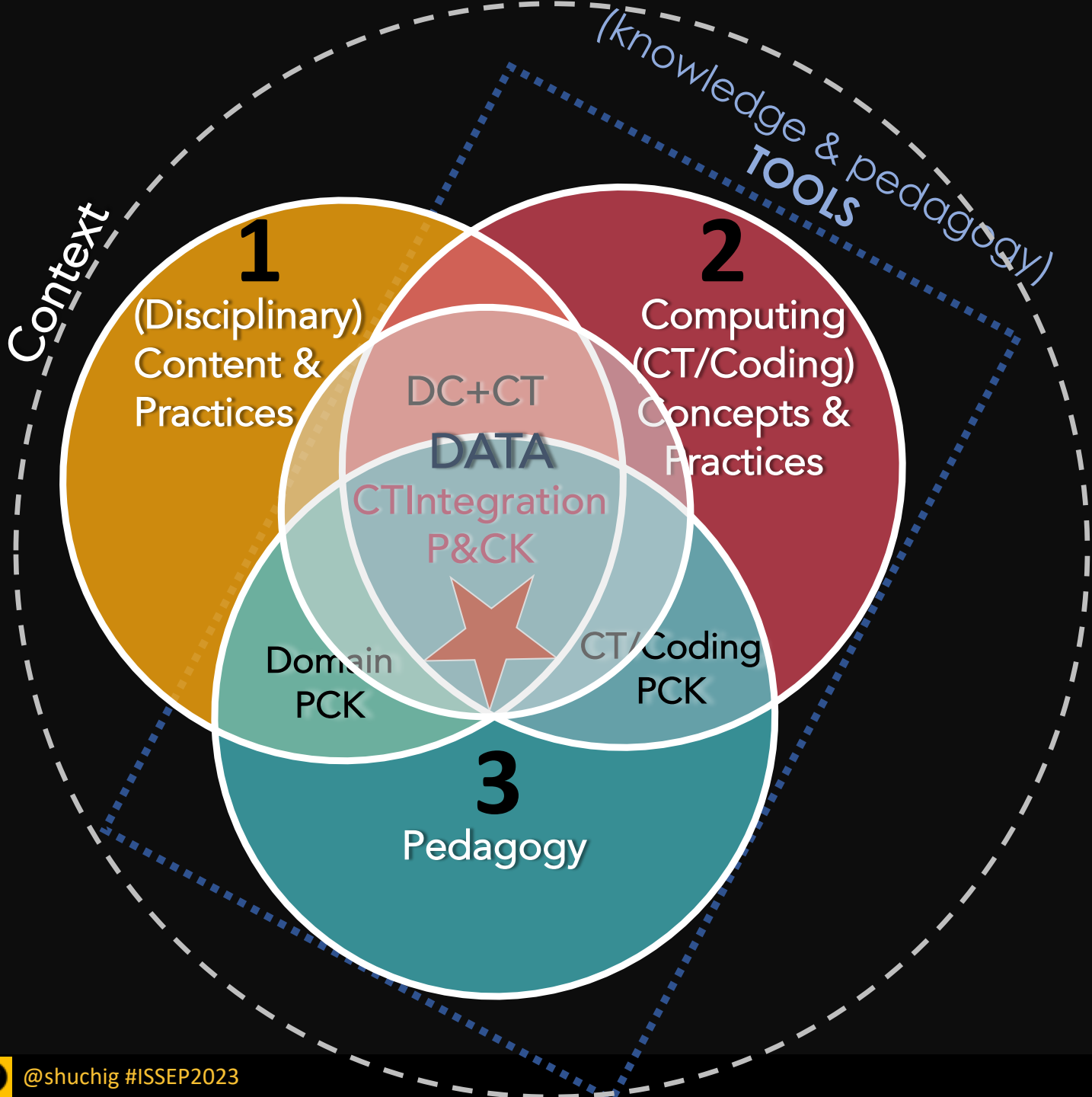
WHY INTEGRATE PROGRAMMING IN OTHER SUBJECTS?

Seymour Papert’s pioneering work on using programming in mathematics classrooms in the 1980s laid the foundation for integrating programming into other subjects. Just as meaningful and authentic use of technology tools help enrich learning in the classroom, coding enriches the learning of the host domain. But the benefits of integrating coding in other subjects are bidirectional—integration also helps students learn computing “in context.” This is especially true of STEM subjects.

Computing and STEM share a deeply symbiotic relationship, and as such, mathematics and science classrooms provide perhaps the most intuitive and easy non-CS contexts for integrating computing. . . . STEM contexts can enrich computational learning while also providing valuable opportunities to embed coding and CT in established and accessible (as well as required) STEM courses.—Grover and Pea (2018).

Although there has been much focus on teachers’ understanding of computational thinking and programming constructs as well as their attitude toward computational thinking ideas, Yadav and colleagues have concluded through their research that CT can also change teachers’ disciplinary teaching practice. Foregrounding CT provides teachers with tools and practices they can use to deepen disciplinary learning by providing students with an opportunity to engage in “thinking skills” that are not always explicitly focused on, especially at the elementary level. For example, in the mathematics example discussed later in this chapter, as teachers ask students to draw a triangle in Scratch, they are drawn into thinking about interior and exterior angles. This is because rotating the Scratch sprite requires the use of the exterior angle, even though one might intuitively use the interior angle. It provides an opportunity for the students to dive deeper into geometrical ideas that



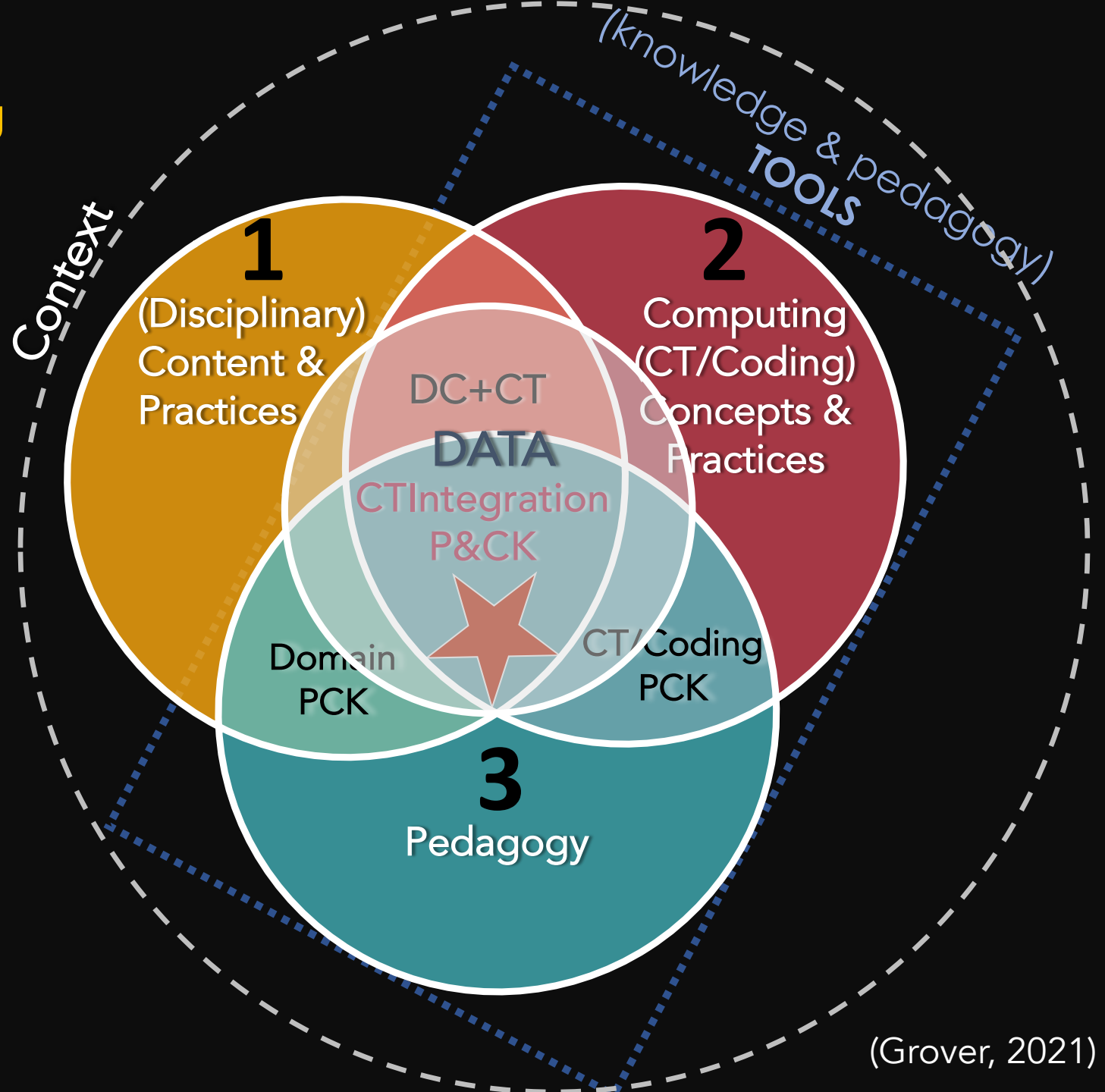


CTIntegration:
 A FRAMEWORK FOR
 DESIGNING FOR
 AND ANALYZING
 THE
 INTERCONNECTED
 ELEMENTS OF CT
 INTEGRATION INTO
 OTHER DOMAINS

Grover, S. (2021). 'CTIntegration': A Conceptual Framework Guiding Design and Analysis of Integration of Computing and Computational Thinking Into School Subjects.
<https://doi.org/10.35542/osf.io/eg8n5>

DATA as the linchpin for integrating STEM (& non-STEM) and Computing

- Data from (or relevant to) the domain and communities
- Answer questions from the domain
 - Critically examine social & scientific phenomena
 - Evidence-based reasoning
 - Recognizing patterns
 - Make predictions
 - Support argumentation
- Data practices
- Data representations & abstractions
- Data analysis tools (spreadsheets, databases, ...)



Clarity on what to teach; build enduring understanding and problem solving skills that transcend coding environments, contexts

Attend to conceptual learning through cognitive & non-cognitive-- leveraging backgrounds, connections, community,...

03

Cognitive + socio-cultural/socio-political



01

Computational Thinking/Concepts /Practices

02

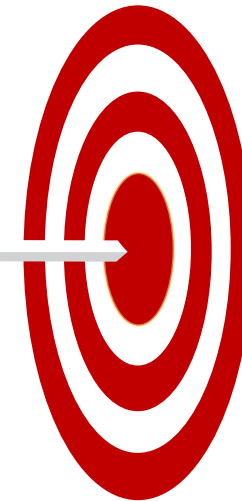
Integration with/into subjects

04

Plurality of pedagogies

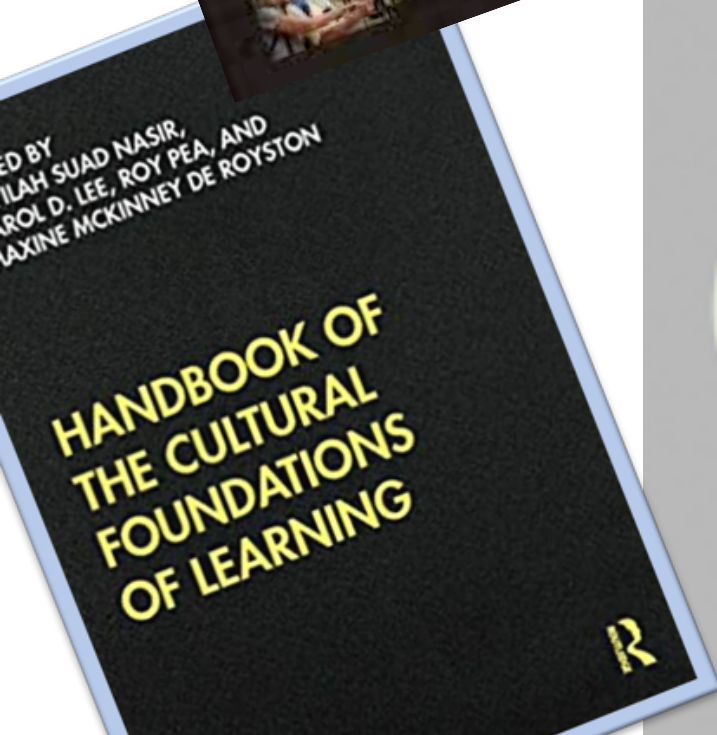
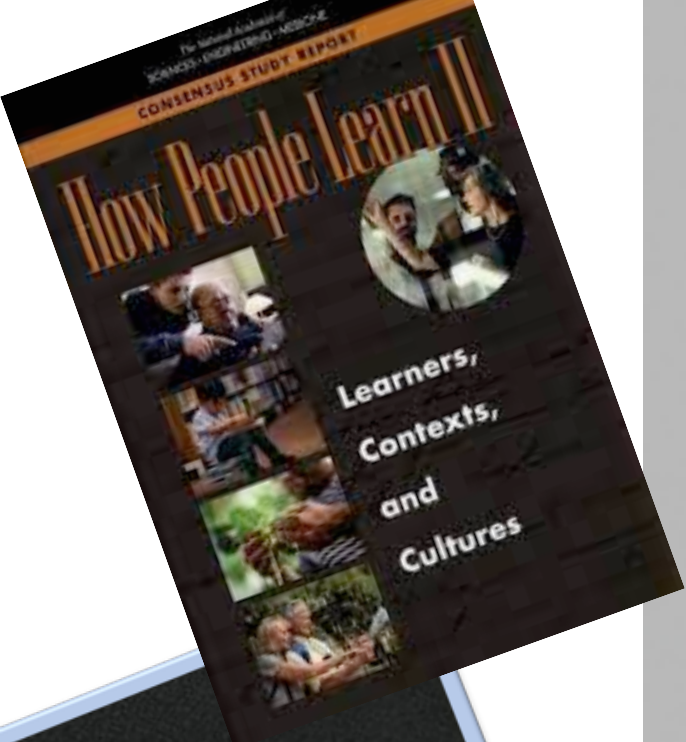
05

Teacher preparation



No effort to introduce CS can succeed without preparing teachers; so designing; building on their teaching experiences





CULTURAL BACKGROUND
Society's expectations and beliefs



TEACHING CONTEXT
Teachers' conceptions of learning and teaching
School culture
Instructional method
Assessment method
Rules
Curriculum

CONCEPTIONS OF LEARNING



INDIVIDUAL CHARACTERISTICS
Age
Cognitive development
Motivation
Self-concept



HOME CONTEXT
Familial cultural beliefs
Familial conceptions of learning

CONTEXT

CULTURE

Attending to the
**SOCIOCULTURAL
&
SOCIOPOLITICAL**
For
EQUITY

CONVERSATION

CONNECTION

Learner-Centered and Culturally Relevant Pedagogy

Tia C. Madkins, Jakita O. Thomas, Jessica Solyom, Joanna Goode, and Frieda McAlear

CHAPTER 12

INTRODUCTION: CULTURALLY RELEVANT PEDAGOGY

Underrepresented minority students (for example, Black, Latinx, Native American/Alaskan, Hawaiian / Pacific Islander in the United States) have historically experienced racial bias and structural inequities both inside and outside of school settings. Educational inequities appear at all levels, from low funding for schools with high proportions of underrepresented students of color to diminished teacher and counselor expectations, tracking students into remedial and special needs programs, and over-referring students to school disciplinary officials. For underrepresented students of color, these practices are an extension of colonial and assimilative educational practices, have led to the development of school-perpetuated (historical) trauma, and contribute to experiencing an education environment that feels irrelevant, hostile, and unwelcoming.

Culturally relevant pedagogy (CRP) was first proposed by Ladson-Billings as well as Allen and Boykin in the 1990s. CRP is founded on the idea that learning grounded in a familiar cultural context can potentially increase equitable outcomes. This framework outlines three tenets for academic success: (1) implementing academic rigor, (2) honoring students' cultural and linguistic backgrounds, and (3) helping students to understand, recognize, and critique social inequities. This mode of teaching also emphasizes an authentically caring rapport between teacher and student and connecting curriculum to students' home cultures and everyday lived experiences.



CULTURALLY RESPONSIVE-SUSTAINING COMPUTER SCIENCE EDUCATION: A FRAMEWORK



KAPOR CENTER

Promote “computational discourse” and discussions around computing concepts

Encourage **conversations** about contexts, cultures, connections as they relate to computing (even if it involves confronting uncomfortable truths).

Make programming accessible by **connecting** learning to students’ interests, social identities, perspectives, & everyday lives

(e.g., inviting a student who skateboards to help introduce a culturally situated design tool about the culture & math of skateboarding)

Contextualize CS/programming examples and activities in students’ lives and communities

(e.g., creating the best transportation route for afterschool activities in their community as a way of learning about efficient algorithms).

Draw from students’ **cultural** assets and knowledge to use as building blocks for examining programming topics

(e.g., using popular music to discuss programming paradigms such as loops and linked lists, or using family recipes to discuss how the same algorithm can be represented in different ways)

Through attention to context, culture and connections, we demonstrate what matters when you teach computing (or AI, or whatever the next new tech is)...

It's always the **humans** in the equation that matter above all else.

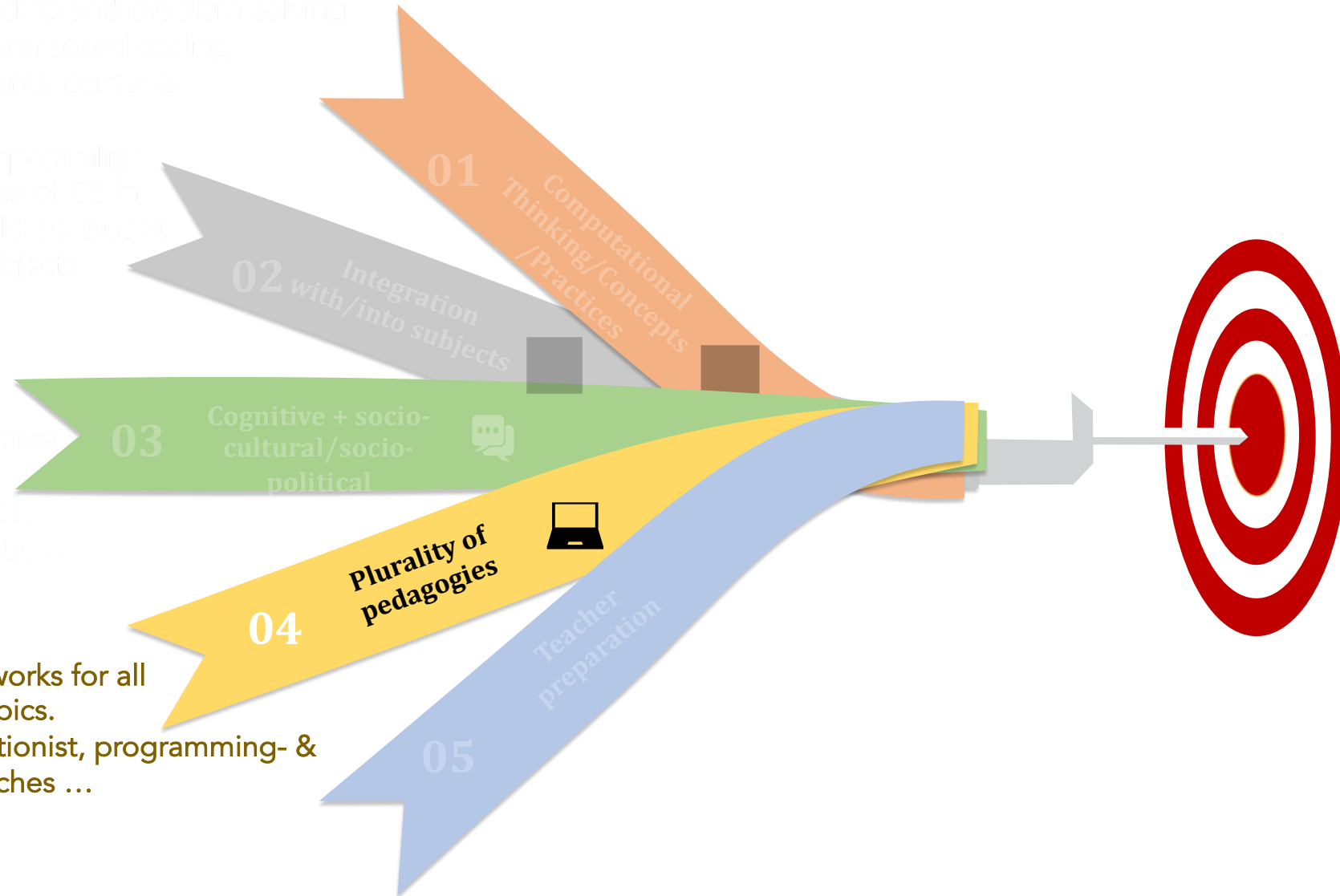
Clarity on what to teach, build enduring understanding and problem solving skills that transcend coding, environments, contexts

For students to see applicability and understand the use of CS in the real world, it should be taught in conjunction with subjects

Attend to conceptual learning through cognitive, non-cognitive, leveraging bed-grounds, connectors, communities...

No single approach works for all contexts, learners, topics. Unplugged, constructionist, programming- & game-based, approaches ...

No effort to introduce CS can succeed without preparing teachers, so designing, building on their teaching experiences



Universal Design for Learning: Reaching All Students

CHAPTER 21

Maya Israel and Todd Lash

INTRODUCTION

Beginning in the very early grades, with a focus on providing meaningful, individualized learning to all learners means that today's classrooms are linguistically diverse than they have ever been. This diversity provides an opportunity to consider how to provide relevant programming experiences.

A primary method of proactively planning through the Universal Design for Learning Applied Special Technology (CAST). The engagement of all learners. This chapter approaches to teaching programming to UDL has been implemented in practice.

► UDL: Maximizing Student Instructional Barriers

The UDL framework offers a practical way to account for the wide range of learner differences. The following concepts underlie this framework.

1. **Learner variability is the norm.**

Learner-Centered and Culturally Relevant Pedagogy

CHAPTER 12

Tia C. Madkins, Jakita O. Thomas, Jessica Solym, Joanna Goode, and Frieda McAlear

INTRODUCTION: CULTURALLY RELEVANT PEDAGOGY

Underrepresented minority students (for example, Black, Latinx, Native American/Alaskan, Hawaiian / Pacific Islander in the United States) have historically experienced racial bias and structural inequities both inside and outside of school settings. Educational inequities appear at all levels, from low funding for schools with high proportions of underrepresented students of color to diminished teacher and counselor expectations, tracking students into remedial and special needs programs, and over-referring students to school disciplinary officials. For underrepresented students of color, these practices are an extension of colonial and assimilationist educational practices, have led to the development of school-prepared (historical) trauma, and contribute to experiencing an education environment that feels irrelevant, hostile, and unwelcoming.

Culturally relevant pedagogy (CRP) was first proposed by Ladson-Billings as well as Allen and Boykin in the 1990s. CRP is founded on the idea that learning grounded in a familiar cultural context can potentially increase equitable outcomes. This framework outlines three tenets for academic success: (1) implementing academic rigor, (2) honoring students' cultural and linguistic backgrounds, and (3) helping students to understand, recognize, and critique social inequities. This mode of teaching also emphasizes an authentically caring rapport between teacher and student and connecting curriculum to students' home cultures and everyday lived experiences.

What we have learned about learning and computer science is that all of the seemingly cultural preferences and interests are profoundly impacted by historical legacies, structural inequities, denied learning opportunities, and belief systems that justify their inequity. - Jane Margolis, *Stuck in the Shallow End*

Hard Fun With Hands-On Constructionist Project-Based Learning

CHAPTER 8

Deborah A. Fields and Yasmin B. Kafai

INTRODUCTION

"Hard fun" is a conclusion that right things may connect with or forget) ethic and

"Hard fun" is a philosophy of MIT. Behind this we have "fun" when we understand about learning as the process and, by d

CHAPTER 3 Creative Coding

Miles Berry

INTRODUCTION: WHAT IS CREATIVE CODING?

"Creative coding" is a pedagogy centered on the expression of pupils' own original

Zestful Learning

Bryan Twarek

CHAPTER 26

INTRODUCTION

Zest is the feeling of spirited satisfaction associated with the intrinsic joy of learning. Both the process and the outcome of learning involves curiosity, collaboration, fun, and pride. Zestful learning (synonymous with joyful learning) positively influences students' attitudes, engagement, self-efficacy, and consequently, all learning outcomes. When students are motivated, engaged, and accompanied by zest, they are better able to process information and make connections critical to the learning at hand. San Francisco Unified School District (SFUSD) computer science (CS) specialist Irene Nolan notes, "If you construct a classroom environment that is joyful, it creates intrinsic rewards and leads to positive feelings about CS. Students think about CS as something that they want to continue learning and exploring."

A zesty class most closely resembles a writer's workshop in elementary school than a college introductory CS classroom. Students spend the majority of time creating and working, rather than listening to lectures. They are deeply engaged in content and ask lots of questions. They create things they care about and receive targeted feedback as guidance; they are proud to showcase and celebrate their work. It is clear that students want to be there, they lose track of time, and they are reluctant to pause their work when the class ends.



The happiest comes from the joy of doing well done, the rest of creating things new. - Simone de Beauvoir

Guided Exploration Through Unplugged Activities

CHAPTER 7

Paul Curzon & Shuchi Grover

WHAT IS GUIDED EXPLORATION?

The tendency in introductory programming classrooms today is often to have students jump into coding right away thanks to the "low floor" feature of block-based environments like Scratch, Snap!, and others, which make it possible to create simple working programs quite literally in a snap! Exploration and discovery learning has been shown to offer many benefits, such as affording students more agency in the learning process and stimulating metacognitive activities, such as hypothesis testing. However, researchers have also noted limitations of unstructured exploration or discovery learning in several contexts. This includes and especially introductory programming. For example, if learners are given too much freedom, they may fail to encounter programming concepts that they must grasp to implement exploratory learning behavioral activities in practice typically adds structure and is framed as a positive step in the questions—they are encouraged

One approach to guided exploration—away from the common programming activities is programming concepts. They

Let me and I forget, but we and I remember, together we and I learn. - Benjamin Franklin

CHAPTER 23 Worked Examples & Other Scaffolding Strategies

Jane Waite & Shuchi Grover

INTRODUCTION: THE WHY AND WHAT OF SCAFFOLDING

► Why Scaffolding?

Learning to program is known to be non-trivial. It is a complex cognitive activity that is often difficult for novice learners regardless of age and programming environment. One of the reasons programming is considered difficult is due to the high cognitive load imposed by the concepts that learners need to master as they learn to code. They have to make sense of the task at hand, conceptualize a design, implement and evaluate a programmed solution.

Although we strive for hands-on engagement with programming, activities that rely on open-ended programming with minimal guidance may result in a lack of explicit engagement with, and understanding of, programming concepts and patterns. In order to develop a deep sense for computational problem-solving, young learners should be fully shepherded through meaningful learning activities with high-quality examples of coded solutions before they begin writing programs.

This chapter presents teaching approaches and sequences of approaches that teachers can use to reduce students' cognitive load while learning to program. All these teaching approaches scaffold learning, thereby making lessons more cognitively manageable. The approaches are code reading, worked examples, live coding, Use-Modify-Create, and FRIMM, or sensible combinations thereof.

Scaffolding situations are those in which the learner gets assistance or support to perform a task beyond his or her own reach if pursued independently when "unassisted". - Roy Pea (paraphrasing Wood, 1996)

There is no failure. Only feedback. - Nelson Mandela

CHAPTER 6 Feedback Through Formative Check-ins

Shuchi Grover, Vicky Sedgwick, & Kelly Powers

INTRODUCTION: WHAT AND WHY OF FORMATIVE FEEDBACK?

Formative feedback refers to formal and informal assessment moves or procedures that teachers employ in an effort to make inferences about what their students know and can do during their routine classroom learning. This is seen as assessment for learning (as opposed to assessment of learning, which is the more summative view of assessment). The overarching objective of the formative assessment process is not to assign a performance grade to a student but rather to supply reliable evidence to the teacher and student that could be used to enhance students' learning.

Computer science teachers can informally assess students in several ways, for example, a show of hands in response to a question, students' expressions of frustration, disengagement, or joy during a coding task, and informal conversations with students as they code and debug their programs. However, education literature makes the case for formal methods of feedback collection as well. Groundbreaking classroom research in the late 1990s by Paul Black and Dylan Williams showed that formative assessment in the classroom improves student learning.

Formative assessment is a process that involves both teachers and learners, and is characterized by the following:

1. When teachers implement formative assessment as a process in collaboration with

Integrating Programming Into Other Subjects

CHAPTER 9

Shuchi Grover & Aman Yadav

INTRODUCTION

The goal of teaching programming in K-12 school years is to introduce learners to a fundamental skill for the times we live in. Algorithms are tested alongside reading, writing, and arithmetic. However, a computer is not the only space to learn coding. Many subjects offer students an opportunity to learn coding in the context of those disciplines. In fact, some believe programming within a science or a social studies class enhances the learning because of the concrete contexts in which it can be demonstrated. Solving along with coding offers the opportunity to develop a key 21st-century skill: computational thinking (CT)—a composite set of thinking and problem-solving skills closely related to, and learned through, coding. This chapter discusses integrating CT and coding in lessons, and presents vignettes from elementary classrooms that teachers can draw inspiration from, to bring CS

WHY INTEGRATE PROGRAMMING IN OTHER SUBJECTS?

Seymour Papert's pioneering work on using programming in middle school in the 1980s laid the foundation for integrating programming into other meaningful and authentic use of technology tools help enrich learning coding enriches the learning of the host domain. But the benefits of

My Auntie told me that programming is the future.

CHAPTER 24 X-ing Boundaries With Physical Computing

Sue Sentance and Katharine Childs

We are born makers. We move what we're learning from our heads to our hands, through our hearts. - Brent Green

WHAT IS PHYSICAL COMPUTING?

Physical computing (also called tangible computing) refers to the use of both software and hardware to build interactive physical systems that sense and respond to the real world. It includes building tangible interactive objects or systems, designing with creativity and imagination, and engaging physically as well as mentally. From a learning perspective, physical computing intersects a range of activities often associated with design technology, electronics, robotics, and computer science. But perhaps more importantly, physical computing provides a means to explore the use of technology in a wide range of subjects.

Physical computing can be used for digital making projects that cross curriculum subjects. In digital making projects, children have a project as an end goal and need to use design, programming, and "making" skills to create something functional. Some examples are given later on in this chapter. These kind of activities are well suited to clubs or nonformal settings where children can work on projects over a number of weeks, but can also be used in the classroom using ideas presented in this chapter. Physical computing also provides a link to other subject areas, such as science, music, and physical education.

Physical computing is associated with the learning of programming, something that children (and adults!) can find difficult. Nonphysical computing is screen-based, so the

Peer Collaboration and Pair Programming

CHAPTER 16

Shannon Campe & Jill Denner

INTRODUCTION

Put simply, it is when two or more people work together toward a shared goal. Extensive research shows the benefits of collaborative learning for students' problem-solving and cognitive development. Interactions that involve working with a peer support, challenge, and require students to explain the subject matter, reflect on another person's contribution, and lead to deeper learning. Collaboration is increasingly important in introductory programming classrooms. Computer science (CS) is increasingly being taught as a subject, and more teachers are expected to teach programming as a subject. Collaborative computing is described in the US K-12 CS curricula. Collaborative computing is described in the US K-12 CS curricula of performing a computational task by working in pairs and "requires individuals to navigate and incorporate diverse perspectives, disparate skills, and distinct personalities." Similar to the Framework for Science Teachers Association (CSTA) standards recommend that students seek and incorporate feedback from team members and users to meet user needs. A strategy that can be used with primary and secondary students to facilitate and practice collaboration is **pair programming**.

Pair programming is a pedagogy where two people work side by

Alone we can do so little, together we can do so much. - Helen Keller

CHAPTER 17 Questions and Inquiry

Shuchi Grover and Steven Floyd

THE WHAT AND WHY OF QUESTIONS IN A THINKING CLASSROOM

Questions are central to learning and problem solving. Not only are they integral to the processes by which teachers guide their students in developing their understanding of a topic, they are deeply intertwined with the process through which students construct their learning while exploring concepts. Teachers and teacher-questions can also guide the process of student reflection and self-explanation. Self-explanations have been shown to be valuable to student learning.

Programming is essentially about problem solving. George Polya's

Seven problem-solving heuristics or techniques to build understanding a

series of questions: critically, to understand the problem, and to use the

to look back at the process and reflect for reflection and self-reflection.

Questions at the beginning:

- Do you understand of the words used in stating the problem?
- What are you asked to find or do?
- Do you know the problem in your own words?
- Do you think of a solution or algorithm that might help you understand

Code reading

The task scaffolds/constrains the skill being taught

The resource (code sample read) scaffolds/constrains the concepts being covered

Live coding

The teacher scaffolds/constrains the skill being taught

The teacher scaffolds/constrains the concepts being covered

Worked examples

The resource scaffolds/constrains the skill being taught

The resource scaffolds/constrains the concepts being covered

Use Modify Create

The task, as a sequence of stages, scaffolds the learning by constraining the skills being taught at each stage

Any resources used scaffold/constrain the concepts being covered

PRIMM

The task, as a sequence of stages scaffolds, the learning by constraining the skills being taught at each stage

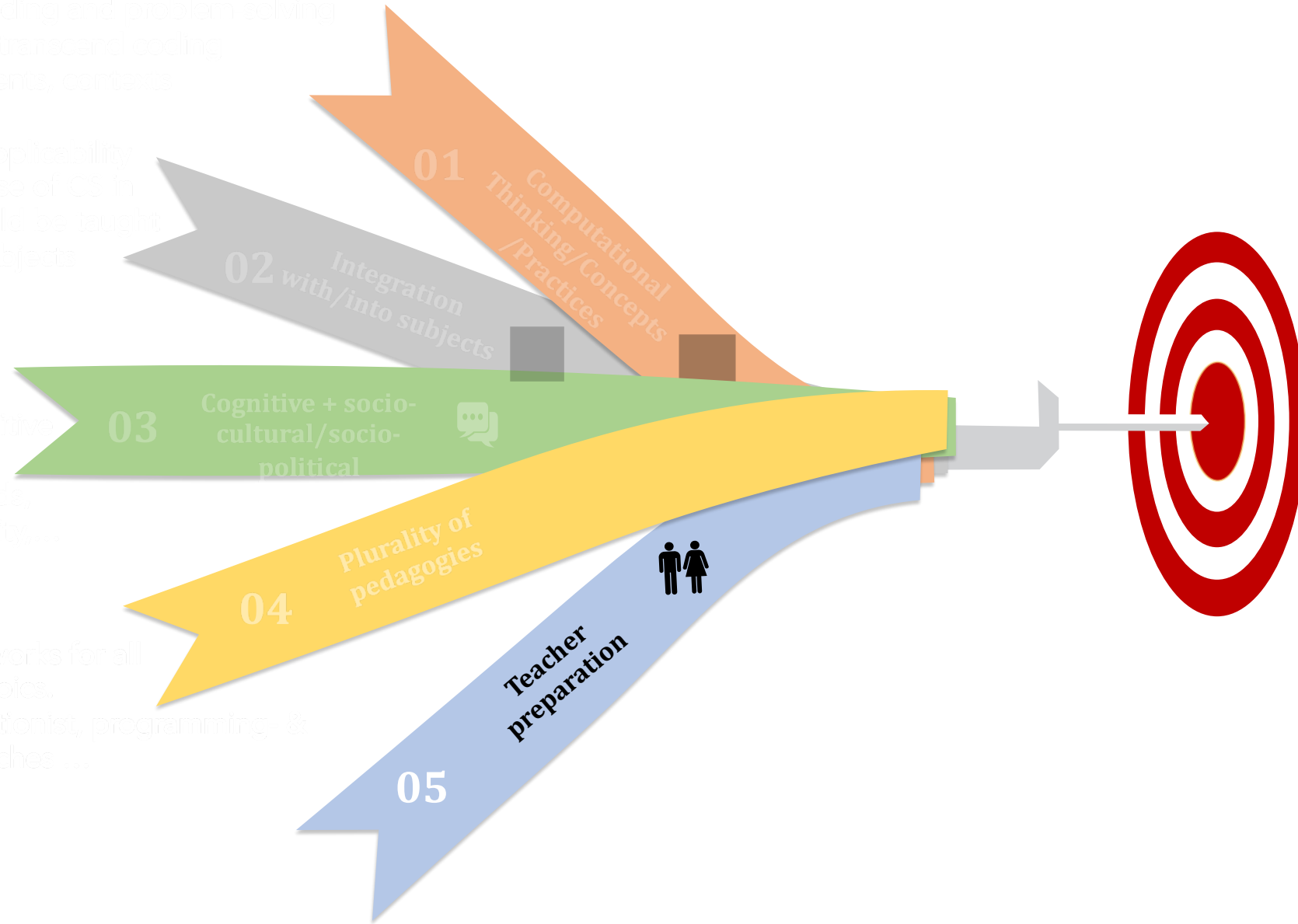
Any resources used scaffold/constrain the concepts being covered

Clarity on what to teach, build enduring understanding and problem solving skills that transcend coding, environments, contexts

Let students to see applicability and understand the use of CS in the real world, it should be taught in conjunction with subjects

Attend to conceptual learning through cognitive / non-cognitive - leveraging bed-grounds, connections, communities, ...

No single approach works for all contexts, learners, topics. Unplugged, constructionist, programming- & game-based, approaches ...



CS Education for K-12 students = CS Education for Teachers



Teachers as key stakeholders

Nothing can be accomplished without adequately preparing teachers and addressing their needs



Co-design

The benefits of co-designing curriculum extend to teacher preparation



Build Teachers' PCK & Build on on Teacher Prior Expertise

Knowing the what of teaching CS is incomplete without knowing the how; build on pre-existing peda. & content expertise of teachers.

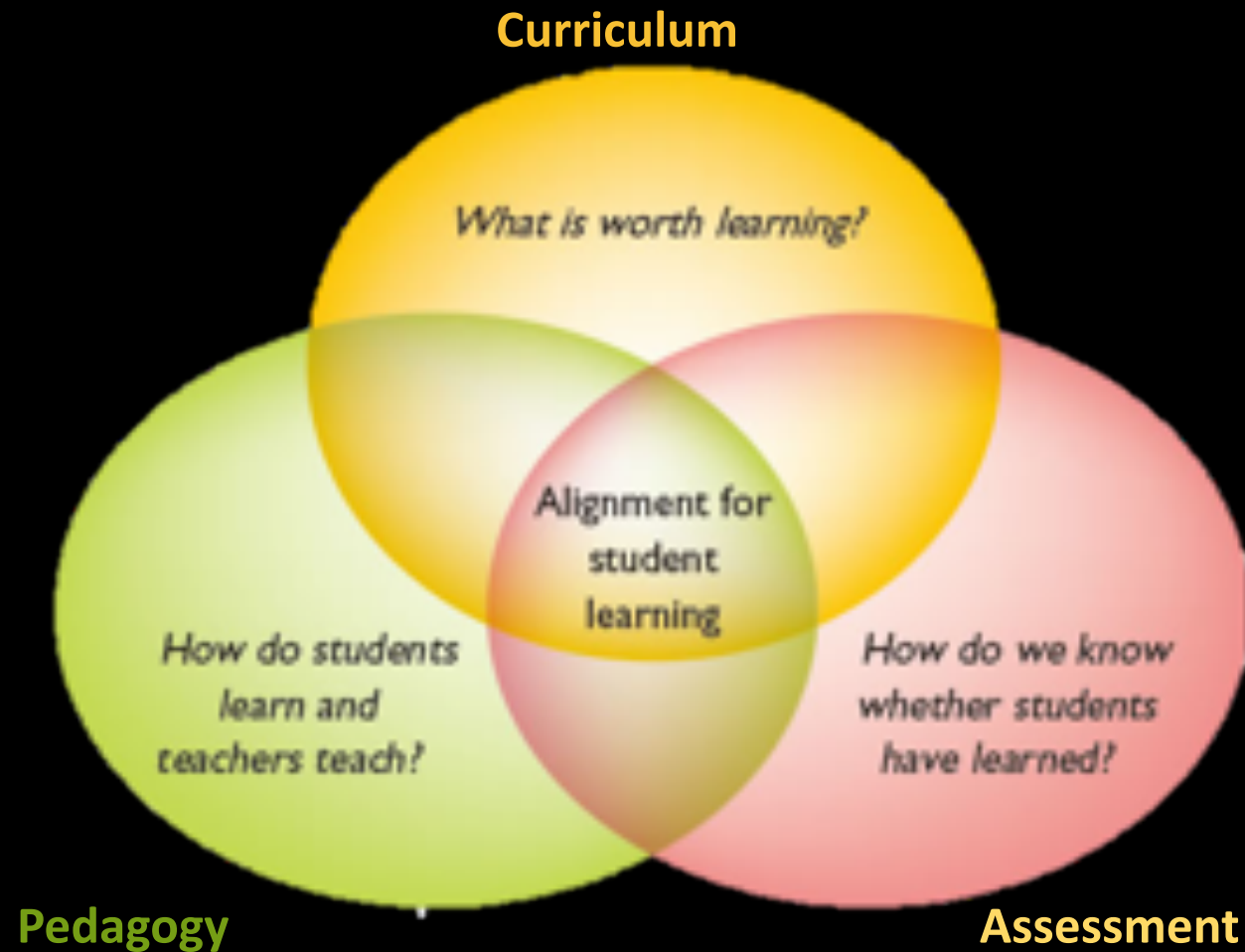


Teacher Communities & Resources

Sustaining CoPs is hard but very worthwhile, for sharing experiences and resources



Curriculum → Pedagogy → Assessment



CHAPTER
6

Feedback Through Formative Check-ins

Shuchi Grover, Vicky Sedgwick, & Kelly Powers

There is no failure. Only feedback.
- Robert Allen

INTRODUCTION: WHAT AND WHY OF FORMATIVE FEEDBACK?

Formative feedback refers to formal and informal assessment moves or procedures that teachers employ in an effort to make inferences about what their students know and can do during their routine classroom learning. This is seen as assessment *for* learning (as opposed to assessment *of* learning, which is the more summative view of assessment). The overarching objective of the formative assessment process is not to assign a performance grade to a student but rather to supply reliable evidence to the teacher and student that could be used to enhance students' learning.

Computer science teachers can informally assess students in several ways, for example, a show of hands in response to a question; students' expressions of frustration, disengagement, or joy during a coding task; and informal conversations with students as they code and debug their programs. However, education literature makes the case for formal methods of feedback collection as well. Groundbreaking classroom research in the late 1990s by Paul Black and Dylan William showed that formative assessment in the classroom improves student learning.

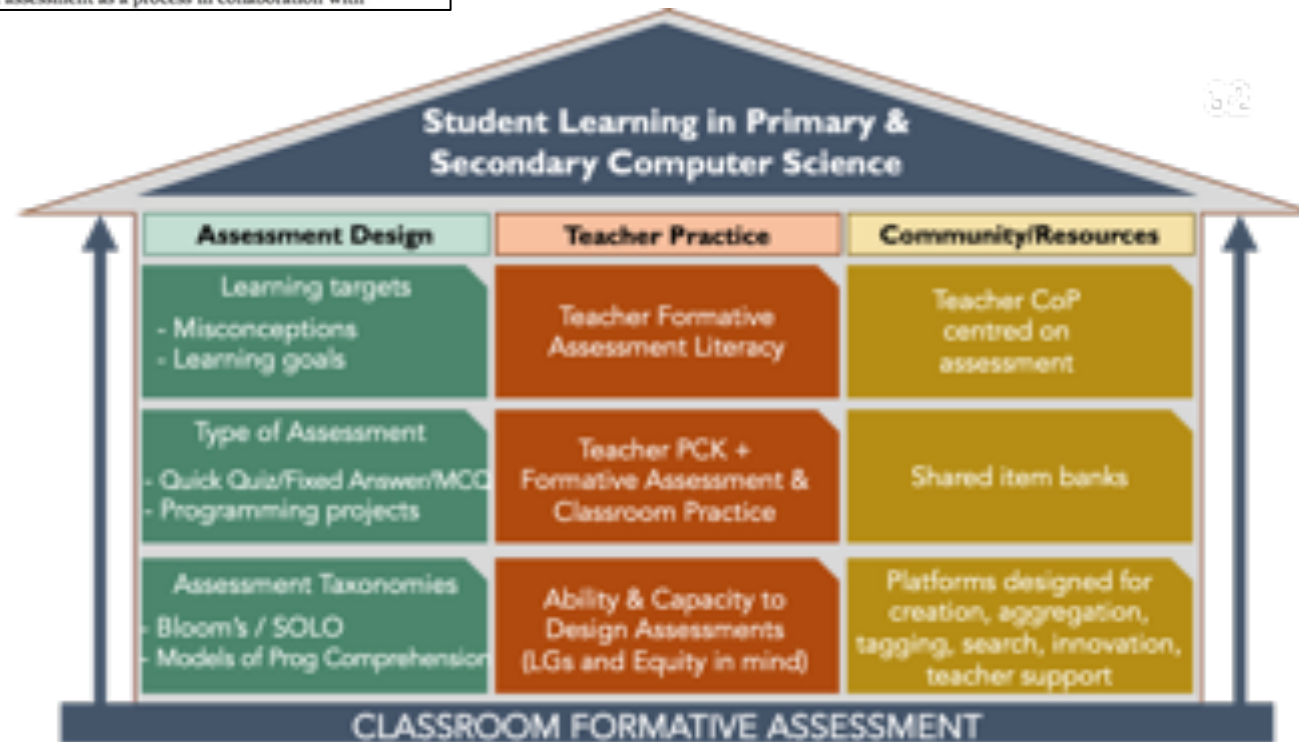
Formative assessment is a process that involves both teachers and learners, and is characterized by the following:

1. When teachers implement formative assessment as a process in collaboration with

Formative Classroom Assessment for Teachers (FCAT)



<https://csassess.org/fcat/>





Why K-12 AI Ed? (aka AI & the coming tsunami!)

State of AI education in K-12

Lessons from K-12 CS Ed for K-12 AI Ed

Challenges, open issues, & recommendations

Challenges & Tensions



Rapidly Changing Landscape



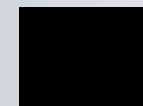
Misperceptions About AI in the Broader Discourse



A Crowded Curriculum & AI's curricular relationship to CS and other core subjects



Building Teacher Capacity



How Deep Can/Should We Go? (What will "AI for All" mean?)



Need for a robust evidence base

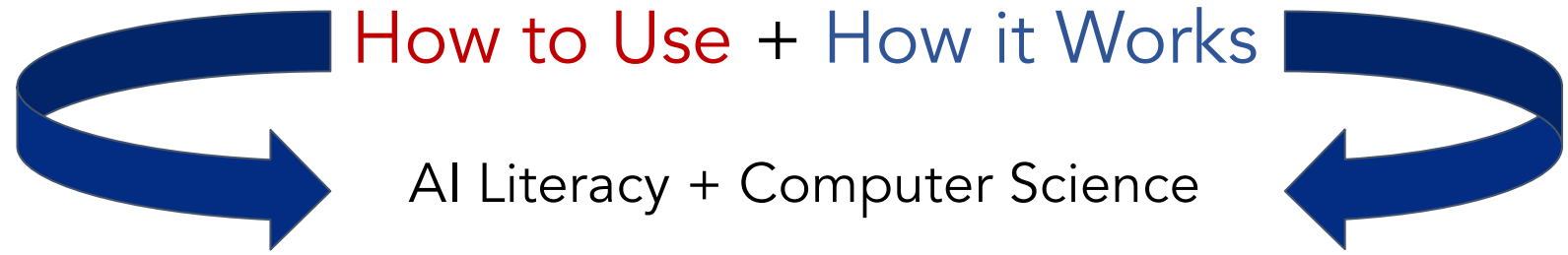


Lack of clarity around terminology & learning goals



AI Literacy

The knowledge, skills, and attitudes associated with how artificial intelligence works, including its principles, concepts, and limitations, as well as how to use artificial intelligence, such as its applications, implications, and ethical considerations.



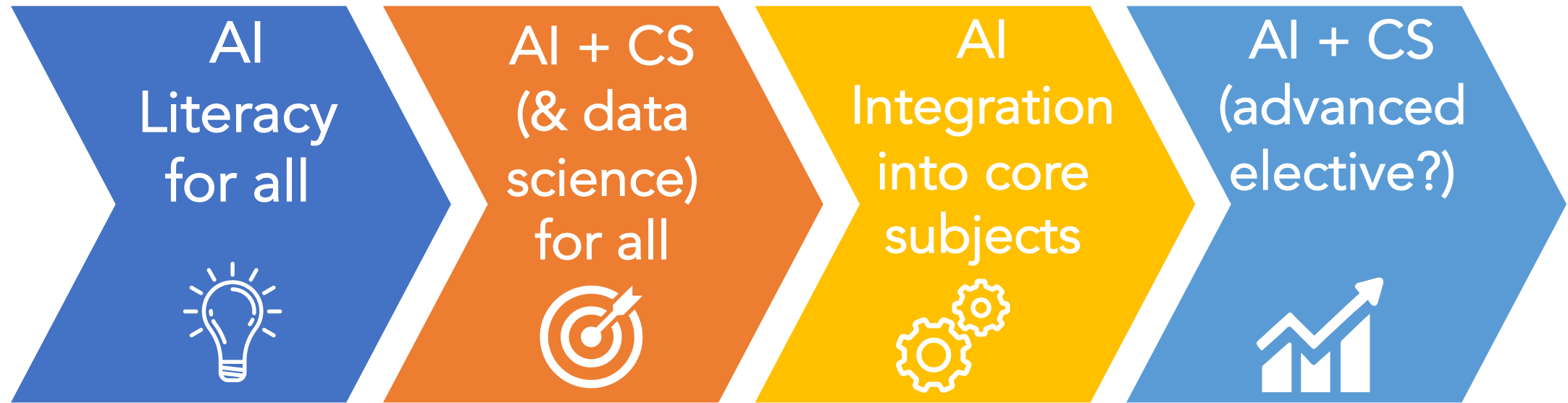
With **AI Literacy**, I CAN:

- Personalize my learning
- Assess safety and privacy with apps and data sharing
- Properly cite AI usage when writing or creating
- Prepare for the future of work
- Act as an informed citizen

With **AI/CS integration**, I CAN:

- Collect data responsibly
- Break down complex problems
- Analyze data critically
- Identify patterns and trends
- Evaluate the effectiveness of solutions
- Ethically evaluate AI systems to minimize bias

K-12 AI Ed Progression

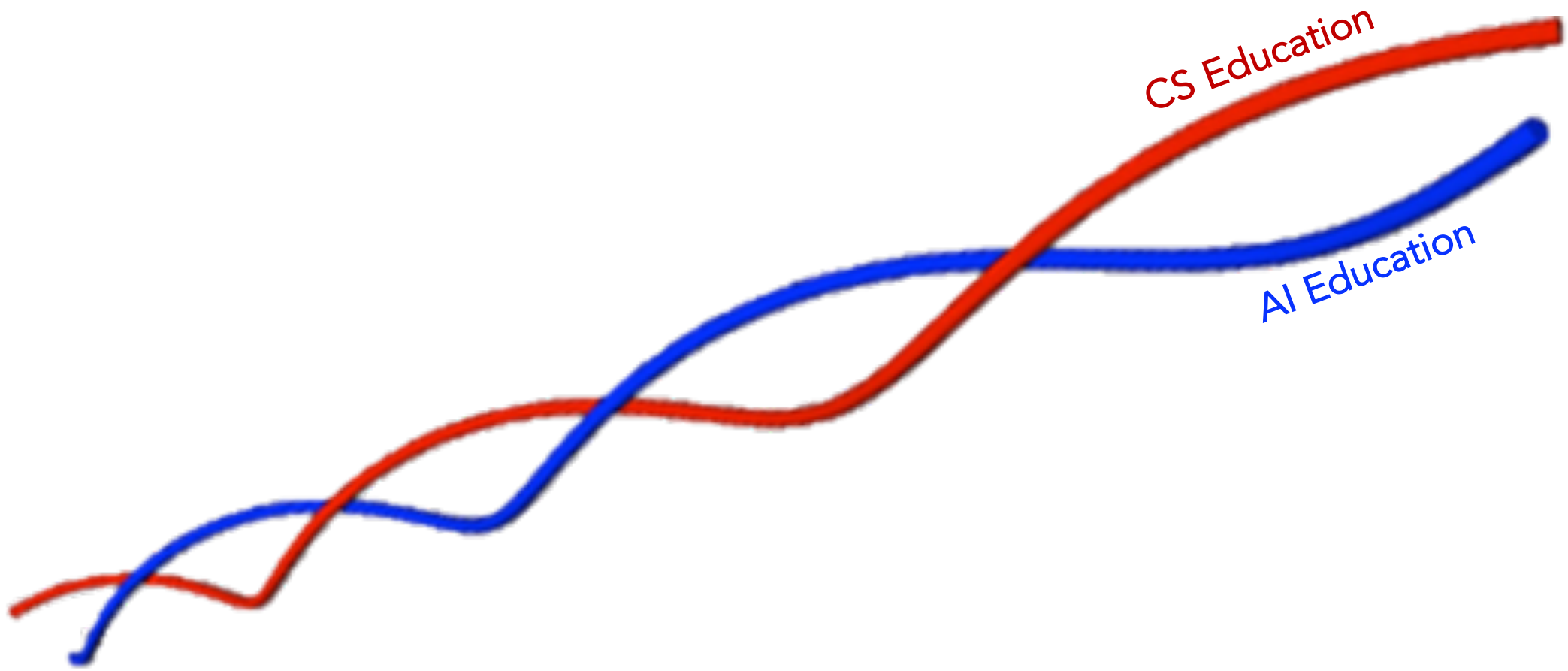


AI LEARNING FRAMEWORK



Integration is the only answer!

- Multi- & inter-disciplinary projects that integrate CS & AI, or CS, AI, & Data Science
- Easily situated in contexts drawn from domains/subjects e.g. science, math, health, earth/climate science, politics, sports, entertainment, culture, digital literacy/cybersecurity,...
- Examination of ethics/biases intertwined throughout



CS Education

AI Education

K-12 CS Standards Revision

- Fall 2023 Begin research
- Spring 2024 Select advisory board(s) and writing team
- **Fall 2024** **Kick off standards writing**
- Fall 2025 Finish draft standards, collect feedback
- Winter 2026 Finalize content of standards
- Spring 2026 Develop exemplary lesson resources, teacher PD
- **Summer 2026** **Publish revised standards**
- Winter 2027 Complete supplementary resource development
- 2026-27 Disseminate Standards and support implementation

BRAVE NEW WORLD



THANK
YOU



@shuchig / shuchigrover.com