

Computational Thinking (CT)

2016-2017 Elementary Inquiry - Notes 20170227

Computational Thinking (CT 6-7) - MinEdBC

- simple algorithms that reflect computational thinking; for sorting, searching, sequence, selection, and repetition; specific statements to complete a simple task; cryptography and code breaking (e.g., cyphers)
- visual representations of problems and data; graphs, charts, network diagrams, info graphics, flow charts, lists, tables, or arrays
- evolution of programming languages; historical perspectives, evolution (e.g., Ada Lovelace, punch cards, Hollerith, Grace Hopper, Alan Turing, Enigma, cyphers)
- visual programming; for example, Kodu, Scratch

CT-PDP

- [Introducing Computational Thinking - GfE](#)

CT-Plugged

- [Swift Playgrounds A new way to learn to code with Swift on iPad - iOS 10](#)
- [Betty's Brain](#)

CT-Unplugged

CS Unplugged

- [Harold the Robot](#) - In this activity children simply give directions to a “robot” (either an adult or another child) and find out which instructions the robot is able to follow, and how their instructions are taken literally.
- [Phylogenetics](#) - Bioinformatics is a discipline that combines computer science and biology. Bioinformatics uses the algorithms and technology of computer science, mathematics, and statistics to solve problems for biology. For example, it has allowed biologists to reconstruct phylogenetic (evolutionary) trees using computer science techniques such as string matching, edit distance, and matrices. This activity lets students participate in the process of reconstructing a phylogenetic tree and introduces them to several core bioinformatics concepts. It's a good exercise for classes in biology

(evolution), bioinformatics (phylogenetics), or mathematics (matrices). It's suitable for ages ten and up, and doesn't require too much prior knowledge: just following instructions, writing, and basic math.

Exploring Computational Thinking (ECT) - Google for Education

- [Exploring Computational Thinking ECT - Google for Education](#)
- [ECT Lesson Plan: CIPHERING a Sentence](#)
- [ECT Lesson Plan: Algorithmic Thinking](#)
- [ECT Lesson Plan: Pattern Machine](#)

Code.org Unplugged

- [CT - code.org Unplugged Monster Catalog pdf](#)
- [site:https://studio.code.org/unplugged](https://studio.code.org/unplugged) - GSearch
- [CS Fundamentals Unplugged](#)
- [CS Fundamentals for Courses 1 - 4 pdf](#)
- [Code Studio Lesson Keys for Courses 1 - 4 pdf](#)
- <https://studio.code.org/unplugged/unplug1.pdf> | unplug1.pdf Intro to the Art of Computer Science; Main Goal: Give the class a clear understanding of what computer science is and how it could be helpful in their lives.
 - Binary Decoder Key
 - Graph paper or Binary Strips
 - Markers
- <https://studio.code.org/unplugged/unplug2.pdf> | unplug2.pdf Computational Thinking; Main Goal: Introduce the "Computational Thinking" model as a way of preparing real-world problems for digital representation.
 - Monster Catalog (1 per group)
 - Blank pieces of paper (3 per person)
- <https://studio.code.org/unplugged/unplug3.pdf> | unplug3.pdf Graph Paper Programming; Main Goal: Help students understand how "coding" works.
 - Sample Drawings/Algorithms Kit
 - Programming Instructions Card
 - Large grid graph paper
 - Markers, pens, or pencils (two or three colors)

- <https://studio.code.org/unplugged/unplug4.pdf> | unplug4.pdf Algorithms; Main Goal: Explain that the same thing can be accomplished many different ways, and sometimes there are “better” ways than others.
 - Tangram Image Pack (one set per pair)
 - Tangram Pieces (one set per pair)
 - Graph paper (five or six sheets per pair)
 - One piece of blank paper
- <https://studio.code.org/unplugged/unplug5.pdf> | unplug5.pdf Functions; Main Goal: Illustrate how repetitive tasks can be stored in a small group that can be “called” several times, instead of wasting space with lots of copies of the same instruction.
 - One foot of string, thread, or fishing line per student
 - 2-4 beads per student
 - 2-4 other accessories (buttons, hoops, spacers) per student
 - One special bead, prism, or student-made sun charm per student
 - One Skills Sheet per group
- <https://studio.code.org/unplugged/unplug6.pdf> | unplug6.pdf Conditionals: (Coding with Cards); Main Goal: This lesson will introduce conditionals, especially as they pertain to loops and if statements.
 - 1 set of Program Pages per class (This has at least one sample program on it)
 - 1 deck of playing cards
- <https://studio.code.org/unplugged/unplug7.pdf> | unplug7.pdf Song Writing; Main Goal: Learn how to define and call functions.
 - Paper and pencils for writing songs
 - Printouts of famous children’s songs with lyrics that change slightly
- <https://studio.code.org/unplugged/unplug8.pdf> | unplug8.pdf Abstraction; Main Goal: Help students to see how often they use abstraction in their everyday life.
 - Paper, pencils, and printed copies of stories for every student
- <https://studio.code.org/unplugged/unplug9.pdf> | unplug9.pdf Relay Programming; Main Goal: Emphasize the importance of checking your work and writing programs in proper sequence.
 - Sample Drawings/Algorithms Kit from lesson 4
 - Programming Instructions Card
 - Large grid graph paper

- Blank notecards or pieces of paper
- Markers, pens, or pencils (two or three colors)
- <https://studio.code.org/unplugged/unplug10.pdf> | unplug10.pdf The Internet; Main Goal: Students will learn how the Internet works, as it relates to URL addresses and web pages.
 - DNS Translation Table
 - Server Address Name Tags
 - Message Packets
 - #10 envelopes
 - Scissors
 - <https://studio.code.org/unplugged/unplug11.pdf> | unplug11.pdf Wrap-Up; Main Goal: Prepare the class for continued learning in computer science. - This is up to the instructor. It's a good idea to have a large variety of items to pique the classroom interest and prompt clever inventions. Some inspiring materials tend to be: - Battery-operated tea lights - Aluminum foil - Paper clips - Markers - Paper (lined, blank, graph, construction) - Scissors - Tape - Fake coins/money - Beads - tring - Popsicle sticks - Pipe cleaners

Notes, Resources and Links

- [2016-2017 Elementary Inquiry Notes 20170227 - GitHub](#)
- [Computational Thinking CT - 2016-2017 Elementary Inquiry - Notes 20170227 - GDoc](#)
- [Computational Thinking - GDrive](#)
- [Computational Thinking - GitHub](#)
- [computational-thinking - GFolder](#)
- [ADST-Content-CompThink - GFolder](#)
-