

This is a section of [doi:10.7551/mitpress/11209.001.0001](https://doi.org/10.7551/mitpress/11209.001.0001)

# Teaching Computational Thinking

## An Integrative Approach for Middle and High School Learning

**By:** Maureen D. Neumann, Lisa Dion

### **Citation:**

*Teaching Computational Thinking: An Integrative Approach for Middle and High School Learning*

**By:** Maureen D. Neumann, Lisa Dion

**DOI:** 10.7551/mitpress/11209.001.0001

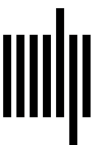
**ISBN (electronic):** 9780262366144

**Publisher:** The MIT Press

**Published:** 2021

### **OA Funding Provided By:**

The open access edition of this book was made possible by generous funding from Arcadia—a charitable fund of Lisbet Rausing and Peter Baldwin.



The MIT Press

---

## Index

*Note: Page numbers in italics indicate figures.*

- \$ (dollar sign), 133
- () (parentheses), 87
- # (pound) symbol, 21, 25
  
- Abstraction, 13. *See also* Algorithmic art;  
Labyrinths; Mazes  
definition of, 9, 15  
as essence of computational thinking, 1–2  
in literature graphs, 43  
state space and, 86  
in temperature graphing, 130
- Acceleration. *See also* Basketball program  
equation for, 108  
Newton's law of universal gravitation,  
107  
Newton's second law of motion, 99–101
- Action-reaction pairs, principle of, 100–101
- Adjacency, 82–83
- Advancing, in mazes, 75–78, 76
- Adventure*, 57
- Aeneas (Trojan hero), 69, 164n7 (chap. 4)
- Aeneid, The* (Virgil), 69
- Agents, maze, 72  
advancing/backtracking, 73, 75–78, 76  
right bearing versus left bearing, 77, 85
- Air hockey activity, 111
- Albers, Josef, 13, 15, 28
- Albers-style art, simulating  
Codesters algorithm, 30–31, 32  
learning activities, 29, 32  
programming skills used in, 29  
Scratch algorithm, 29–30, 30
- Alcuin of York, 86
- Alcuin's puzzle  
description of, 86  
state space, 86–88, 88–89  
state-transition graph, 90, 91  
Trémaux's algorithm applied to, 91, 91–92
  
- Alexander the Great, 59
- Algorithmic art, 13, 15–16  
Albers-style, 28–32, 30, 32  
Andrade-style, 22–28, 23, 26  
combining styles of, 39  
Kandinsky-style, 16–22, 18–20, 23  
student interpretation of, 16  
Vasarely-style, 33–39, 35, 37–38
- Algorithmic processes. *See also* Algorithmic  
art; Maze-threading algorithms  
algorithmic complexity, 64, 164n6 (chap. 4)  
breadth-first search, 78  
definition of, 10  
labyrinth generation, 61–64, 63  
literature graphs, 43  
temperature analysis, 129–131
- Amenemhet III, temple of, 60, 163n2
- Anchises (Greek legend), 69
- Andrade, Edna, 13, 15, 22
- Andrade-style art, simulating, 22–28  
Codesters algorithm, 24–28, 27  
learning activities, 24, 26, 27  
programming skills used in, 23  
Scratch algorithm, 24, 26
- Angry Birds*, 93
- Append function, 138
- Ariadne (Greek legend), 60, 66, 67
- Art, algorithmic. *See* Algorithmic art
- Average temperature data, processing.  
*See* Temperature trend analysis
  
- Backtracking, in mazes, 73, 75–78, 76
- BASIC, 8
- Basketball program, 108–111, 158–159
- Bell, Tim, 14
- Bellman, Richard, 78
- Best-fit line, temperature trends, 131–132,  
139–141, 140

- Bézier curves, 33, 34
- Bookend literary characters, 48
- Books, analyzing with graphs. *See* Literature graphs
- Breadth-first search algorithm, 78
- Brennan, Karen, 10
- Bumper Rocks collision program
  - background and sprites, 101–102, 102
  - collisions and collision function, 103–105
  - engine, 106
  - full program, 156–158
  - keyboard functions, 103
  - movement function, 106
  - programming skills used in, 101
  - setup, 102
- Burns, Rebecca C., 11
- Cairo, Alberto, 119
- Cardinality of sets, 81
- Carpenter, Edmund, 59
- Chakravayuha*, 61
- Chapters, graphing by, 48–50, 49
- Charlemagne, 86
- Chartres Cathedral labyrinth, 71, 71
- Christian labyrinths, 57, 69–71, 71
- Circuitous paths, 57. *See also* Labyrinths
- Classroom weather stations, 119
- Climate data analysis, 124–139
  - best-fit line, 131–132, 139–141, 140
  - classroom weather stations, 119
  - data processing with Python, 135–139
  - data processing with spreadsheets, 131–135, 132–136
  - line graph and scatter plot of data, 130, 131
  - NOAA Climate Data Online dataset, downloading, 125–129, 127–129
  - programming skills used in, 130
  - temperature analysis algorithm, 129–131
  - transference of skills to different datasets, 142–143
- Climate Data Online dataset (NOAA), 125–129, 127–129
- Closed systems, 100
- Codesters, 2, 8, 14
  - Albers-style art algorithm, 30–31, 32
  - Andrade-style art algorithm, 24–28, 27
  - basketball program, 108–111, 109, 158–159
  - Bumper Rocks collision program, 101–107, 156–158
  - comments in, 21
  - coordinate grid, 18, 19
  - cost of, 163n1 (chap. 2)
  - hexadecimal colors in, 19, 21, 29, 31
  - indentation in, 35
  - Kandinsky art algorithm, 18–22, 19–20
  - online tutorials for, 16
  - ping-pong program, 112–115, 159–161
  - simulating laws of physics in, 13
  - Vasarely art algorithm, 34–38
  - velocity in, 97, 97
- Coding, learning, 148–150
- Collaboration, 93, 149–151
- Collection of data, 118
- Collision function, 105, 110–111
- Collisions
  - basketball program, 110–111
  - Bumper Rocks program, 101–107, 156–158
- Colors, hexadecimal, 19, 21, 29, 31
- Combinatorial puzzles, solving with
  - Trémaux's algorithm, 85, 86–92, 88–91
- Commands. *See also* Functions
  - import, 136
  - INDEX, 133
  - LINEST, 139
  - pip3 install matplotlib, 136
  - pip3 install numpy, 136
- Comments, Codesters, 21
- Common Core State Standards for Mathematics*, 65
- Communication, computational thinking and, 2
- Complexity, algorithmic, 64, 164n6 (chap. 4)
- Comprehension of code, 10–11
- Computational thinking
  - applications of, 2
  - characteristics of, 1–3
  - divide and conquer, 64
  - integration into classroom, 11–12, 13–14, 145–151
  - need for, 1
  - skills for, 8–11
- Computer programming skills. *See* Programming skills
- Computer science (CS)
  - definition of, 2
  - in intermediate and secondary schools, 7–8
- Computuses, 70
- Concentric circles/squares. *See* Algorithmic art
- Conditionals, 9–10
  - in basketball program, 108
  - in Bumper Rocks collision program, 101

- in literature graphs, 43
- in ping-pong program, 112
- in Scratch vector program, 98
- in temperature graphing, 130
- Connected mazes, 74
- Context, grounding data in, 121–122, 122
- Conway's Game of Life, 53–55, 54
- Coordinate grid, Codesters, 18, 19
- Coordinate pairs, 18
- Corn maze activity, 78
- Counterexamples, 73
- CRAAP test, 143
- Creativity
  - fostering with rich tasks, 3–4, 146–147
  - need for, 1–2
- Cremona (Italy), labyrinth in, 68, 70
- Cretan labyrinths
  - design of, 57–60, 58–59
  - drawing algorithms for, 61–64, 63, 153–156
  - eleven-circuit, 64, 65
  - historical examples, 60, 163n2
  - kernel of, 62
  - learning activities, 60, 61, 64, 69
  - path length estimation, 65–67, 67–68
  - seven-circuit, 62–64, 63, 65
  - three-circuit, 64
- Crowther, Will, 57
- CSUNPLUGGED website, 14
- .csv library, 136–137
- Curriculum integration, 11–12, 13–14, 145–151
  - collaboration, 149–151
  - desired results and learning goals, 146–147
  - interdisciplinary approach to, 12
  - learning to code, 148–150
  - multidisciplinary approach to, 11–12
  - performance tasks, 146
  - productive struggle, 151
  - transdisciplinary approach to, 12
- Curve function, 37
- Curves, Bézier, 33, 34
- Daedalus (Greek mythology), 60, 67
- Data collection, 118
- Data investigation/inquiry, 13. *See also*
  - Climate data analysis
  - elements of, 117–119
  - learning activity, 119
  - manipulated data, 119–124
- Data misrepresentation, 119–124
  - data omissions, 122, 123
  - exaggerated or minimized data, 122–124, 124–126
  - false generalizations, 121
  - insufficient data calculations, 121
  - learning activities, 119, 143
  - out-of-context data, 121–122, 122
  - sampling procedures, 120
  - unjustified conclusions, 124
- Data processing
  - with Python, 135–139
  - with spreadsheets, 131–135, 132–136
- Data usage, 10
  - in Albers-style art, 29
  - in Andrade-style art, 23
  - in literature graphs, 43
  - in temperature graphs, 130
  - in Vasarely-style art, 33
- Decimal numbers, 164n9 (chap. 4)
- Decision points (mazes), 72, 75, 78, 83, 86
- Degree of vertex, 45
- Delta, 29, 32
- Depth-first search, 78, 79
- Dewey, Thomas, 120
- Digraphs, 48
- Diodorus, 60
- Direction, vectors, 94–98
- Disconnected mazes, 74, 75
- Disease propagation, graphs of, 52–55, 54
- Displacement, 94–98, 108
- Divide and conquer, 64
- Dollar sign (\$), 133
- Doob, Penelope Reed, 164n7 (chap. 4)
- Down\_key function, 113
- Drake, Susan M., 11
- Draw\_lines function, 36
- Dynamics, law of, 93, 99–100
- Edges (graph), 41–43, 82–83
- Education, twenty-first century, 4–5
  - adapting and applying to new situations, 7
  - goals of, 3
  - learning from mistakes in, 5–6
  - productive struggle in, 5–6
  - real-life contexts for, 3–4
  - rich tasks for, 3–4
  - team settings in, 6–7
  - technologically prolific environments, 7–8
- Egypt, labyrinths in, 60
- Elastic collision, 156–158

- Emotions, impact on learning, 5–6
- Empty sets, 81
- England, labyrinths in, 59, 67
- Equations
  - acceleration, 108
  - displacement, 96, 108
  - elastic collision, 103
  - path length estimation, 66–67
  - speed, 94
  - velocity, 108
- Even, Shimon, 75
- Events
  - in Bumper Rocks collision program, 101
  - definition of, 9
  - in ping-pong program, 112
- Evidence graphs, 41–42, 50–51, 51–52
- Exaggerated data, 122–124, 124–126
- Exponential literary characters, 48
- Facebook, 41
- False generalization, 121
- Field marks, 62
- Fontana, Giovanni, 71
- Food insecurity, need for solutions on, 117
- Force, 94, 99
- Ford, Lester R., Jr., 78
- France, labyrinths in, 67, 71
- Frieze patterns, 57, 58
- Frustration, 5–6, 151
- Fuchs, Lynn S., 7
- Full-arc literary characters, 48
- Functions
  - in basketball program, 108, 109–111
  - in Bumper Rocks collision program, 101, 103–106
  - definition of, 36
  - in ping-pong program, 112–114
  - in temperature graphing program, 137–139
  - in Vasarely-style art algorithm, 36–37
- Generalization of data, errors in, 121
- Gibson, J. Paul., 83
- Girls Who Code, 8
- Glide command, 95
- Global warming. *See also* Climate data analysis
  - data misrepresentation in, 121
  - need for solutions on, 117
  - transdisciplinary integrative approach to, 12
- Google
  - depth-first search, 79
  - Sheets, 131–135, 132–136
- Graphical user interfaces (GUIs), 11
- Graphs
  - abstracting from mazes, 79–85, 82, 84
  - graph theory, 13, 79–83
  - literature. *See* literature graphs
  - social network, 41–42, 52–55, 54
  - state-transition, 90, 92
  - temperature. *See* temperature trend analysis
- Gravitation, 107, 110
- Gravity, 107
- Group work, 6–7, 93
- Gun violence, need for solutions on, 117
- Hadley, John, 86
- Hampton Court, maze at
  - abstracting graphs from, 83–85, 84
  - features of, 57–58, 72, 72
  - Trémaux’s algorithm applied to, 83–86, 85
- Harry Potter and the Sorcerer’s Stone* graphs
  - algorithm design, 43
  - cleaning, 46–47
  - creating, 43–46, 46
  - digraphs, 48
  - evidence graphs, 50–51, 51–52
  - graphing by chapter, 48–50, 49
  - programming skills used in, 43
  - subgraphs, 46–47
- Herodotus, 60, 163n2
- Hexadecimal colors, 19, 21, 29, 31
- Histories, The* (Herodotus), 60
- Hour of Code, 8, 16
- House primary winners, fundraising by, 122–123, 124
- How Charts Lie* (Cairo), 119
- Idle, 135
- Immigration, need for solutions on, 117
- Import command, 136
- Incident, 82
- Indentation, code, 35
- INDEX command, 133
- India, labyrinths in, 59
- Inertia, 94, 99
- Infectious diseases, need for solutions on, 117
- Information computing and technology (ICT), 7–8
- Information fluency, need for, 1

- Innovation, inspiring, 1, 11, 151
- Integers, 81
- Interdisciplinary integrative approach, 12
- Investigation, data. *See* Data investigation/inquiry
- Italy, labyrinths in, 59, 68, 70, 71
- Iteration, in labyrinths, 13, 62–64
- Jericho labyrinth, 69–71
- Junctions (maze), 57–58, 71–72
  - abstracting graphs from, 83, 84, 84
  - in breadth-first search algorithm, 78
  - degree of, 72
  - in maze at Hampton Court, 72
  - old versus new, 75, 75
- Kandinsky, Wassily, 13, 15
- Kandinsky-style art, simulating
  - algorithm design, 17
  - circles and concentric circles in, 22, 23
  - Codesters algorithm, 18–22, 19, 20
  - learning activities, 17, 21, 22
  - programming skills used in, 16
  - Scratch algorithm, 17, 18
- Kern, Hermann, 57, 59, 60
- Kernels, labyrinth, 62
- Keyboard functions
  - Bumper Rocks program, 103
  - ping-pong program, 113–114
- Kinematics, 93, 108. *See also* Physics, laws of
- Knossos, Palace of, 57, 59, 60
- Labyrinths, 13. *See also* Mazes
- Christian, 57, 69–71, 71
  - circuitous nature of, 57
  - Cretan, 57, 58, 59–69, 59, 63, 65, 67–68, 153–156
  - definition of, 57
  - frieze patterns compared to, 57, 58
  - historical examples of, 57
  - learning activities, 60, 71
  - Roman, 57, 67–69, 70
  - spirals compared to, 57, 58
- Laws of physics. *See* Physics, laws of
- Learning activities
  - air hockey, 111
  - Albers-style art, 29, 32
  - Andrade-style art, 24, 26, 27
  - Codesters/Scratch languages, 16
  - data investigation/inquiry, 119, 135, 143
  - Kandinsky-style art, 17, 21, 22
  - labyrinths, 60, 61, 64, 69, 71
  - laws of physics, 94, 100, 101, 108, 110, 111
  - literature graphs, 50, 51
  - mazes, 72, 73, 74, 78, 83, 86
  - social network graphs, 55
  - Spider's Web Morning Greeting, 42
  - Vasarely-style art, 36, 37
  - video game creation, 107
- Left bearing agents, 77–78, 85
- Lego Robotics, 8
- Length, lists, 87
- Length estimation, labyrinths, 65–67, 67–68
- Lethal states, 88
- Libraries
  - .csv, 136–137
  - math, 36
  - Matplotlib, 135–136, 138
  - Numpy, 136
  - random, 21, 36
- Life-size maze activities, 78
- LINEST command, 139
- Lists
  - in Albers-style algorithm, 31
  - decimal numbers as, 164n9 (chap. 4)
  - definition of, 87
  - in Kandinsky-style algorithm, 21
  - in temperature-graphing program, 137–138
- Literacy, technological, 2, 7–8
- Literature graphs
  - algorithm design for, 43
  - benefits of, 41–42
  - cleaning, 46–47
  - creating, 43–46, 46
  - digraphs, 48
  - evidence graphs, 50–51, 51–52
  - graphing by chapter, 48–50, 49
  - learning activities, 50, 51
  - literary analysis with, 43–51
  - programming skills used in, 43
  - resources for, 50
  - structure of, 41–42
  - subgraphs, 46–47
- Lives of the Caesars, The* (Suetonius), 69
- Loops
  - in Albers-style art, 29, 31
  - in Andrade-style art, 23–28
  - in Bumper Rocks collision program, 101, 106
  - definition of, 9
  - in Kandinsky-style art, 16–22
  - in Scratch vector program, 98
  - in Vasarely-style art, 37

- Lucas, Édouard, 75  
*Lusus Trojae*, 69
- Magnitude, vectors, 94–98  
 “Mahabharata” (Sanskrit epic), 61
- Manipulated data  
 data omissions, 122, 123  
 exaggerated or minimized data, 122–124, 124–126  
 false generalization, 121  
 insufficient data calculations, 121  
 learning activities, 119, 143  
 out-of-context data, 121–122, 122  
 sampling procedures, 120  
 unjustified conclusions, 124
- Math library, 36
- Matplotlib library, 135–136, 138
- Mathews, William Henry, 57
- Mazes, 13. *See also* Junctions (maze); Labyrinths; Maze-threading algorithms  
 abstracting graphs from, 79–85, 82, 84  
 agents, 72–73, 75–78, 76, 85  
 computational process, 72  
 connected/disconnected, 74, 75  
 decision points, 72, 75, 78, 83, 86  
 definition of, 57  
 features of, 72–73, 72  
 historical examples of, 57, 71  
 junctions, 57–58, 71–72, 72, 75, 75, 77–78, 83, 85  
 learning activities, 72, 73, 74, 78, 82, 85  
 maze at Hampton Court, 57–58, 72, 83–85, 84–85  
 minimazes, 78, 81  
 multicursal nature of, 13, 58, 71
- Maze-threading algorithms  
 as computational process, 72  
 random walk, 73  
 Tarry’s, 78  
 Trémaux’s, 75–78, 76, 79–81  
 wall following, 73–74
- McTighe, Jay, 146
- Mechanics, 93, 94. *See also* Physics, laws of
- Members of sets, 81
- Microsoft Office, 11
- Minimazes, 78, 81
- Minimized data, 122–124, 124–126
- Minorities, in STEM, 7–8
- Minos, King of Crete, 60
- Minotaur (Greek legend), 60, 67–68
- Mistakes, learning from, 1, 5–6
- Modularization, 9, 33
- Month\_string variable, 137
- Moore, Edward F., 78
- Moretti, Franco, 13, 43
- Motion, laws of, 93  
 acceleration, 99–101, 107–108  
 basketball program, 108–111, 109, 158–159  
 Bumper Rocks collision program, 101–107, 102, 156–158  
 displacement, 94–98, 108  
 learning activities, 94, 100, 101, 107, 108, 110, 111  
 Newton’s first law, 94, 95  
 Newton’s law of universal gravitation, 107  
 Newton’s second law, 99–100  
 Newton’s third law, 100–101  
 ping-pong program, 112–115, 159–161  
 reflection, 111–115  
 velocity, 94–98, 97, 108
- Movement function, 106
- Multicursal mazes. *See* Mazes
- Multidisciplinary integrative approach, 11–12
- Multiplication principle, 87
- Multisets, 81
- Murder and nonnegligent manslaughter rates, 122, 123
- Mystery novels, evidence graphs for, 50–51, 51–52
- National Oceanic and Atmospheric Administration (NOAA) climate data, 125–129, 127–129
- Natural numbers, 81, 164n8 (chap. 4)
- Neumann, Maureen D., 78
- Newton, Isaac, 13, 93  
 first law of motion, 94, 95  
 law of universal gravitation, 107  
 second law of motion, 99–100  
 third law of motion, 100–101
- Next function, 137
- North America, labyrinths in, 59
- Novels, analysis of. *See* Literature graphs
- Numbers  
 decimal, 164n9 (chap. 4)  
 integers, 81  
 natural, 81, 164n8 (chap. 4)  
 rational, 81
- Numpy library, 136
- Office suite, 11
- Omission of data, 122, 123
- Op Art movement, 15, 22, 39. *See also* Algorithmic art

- Pair programming, 8
- Palace of Knossos, 57, 59, 60
- Papert, Seymour, 8
- Parentheses, delineation of lists with, 87
- Path length estimation, 67–68, 71
- Performance tasks, 146
- Perseverance, 1, 4, 6, 151
- Peterson, Roger Tory, 62
- Physics, laws of, 13, 93
  - acceleration, 99–101, 107–111
  - action-reaction pairs, 100–101
  - in basketball program, 108–111, 109, 158–159
  - in Bumper Rocks collision program, 101–107, 102, 156–158
  - displacement, 94–98, 108
  - dynamics, 99–100
  - gravitation, 107, 110
  - inertia, 94, 95
  - learning activities, 94, 100, 101, 107, 108, 110, 111
  - in ping-pong program, 112–115, 159–161
  - reflection, 111–115
  - vectors, 94–98, 97
  - velocity, 94–98, 108
- Ping-pong program, 112–115, 159–161
- pip3 install matplotlib command, 136
- pip3 install numpy command, 136
- Planar graphs, 82
- Plays, analysis of. *See* Literature graphs
- Pliny, 60
- Plot function, 139
- Police killings, data on, 121–122, 122
- Polyfit function, 139, 143
- Pound (#) symbol, 21, 25
- Primary data sources, weather, 13, 125, 135.
  - See also* Climate data analysis
- Problem solving, 1, 4–5
- Processing of data. *See* Data processing
- Productive struggle, 5–6, 151
- Program listings
  - Albers-style Codesters art, 30–31
  - Andrade-style Codesters art, 24–25
  - basketball program, 108–109, 158–159
  - Bumper Rocks program, 102–103, 156–158
  - Codesters velocity program, 97, 97
  - Cretan labyrinth, 153–156
  - Kandinsky-style Codesters art, 20–21
  - ping-pong program, 112–114, 159–161
  - temperature-graphing program, 136–139, 141, 161–162
  - Vasarely-style Codesters art, 34–38
  - velocity in Codesters, 97
- Programming skills
  - in Albers-style art, 29
  - in Andrade-style art, 23
  - in basketball program, 108
  - in Bumper Rocks program, 101
  - description of, 9–11
  - importance of, 1
  - in Kandinsky-style art, 16
  - learning resources for, 148–150
  - in literature graphs, 43
  - in ping-pong program, 112
  - for Scratch vector program, 98
  - for temperature-graphing program, 130
  - transference of, 10, 142–143
  - in Vasarely-style art, 33
- Propositions to Sharpen the Young* (Alcuin), 86
- Python, 2, 14. *See also* Functions; Libraries
  - comments, 21
  - python (.py) files, 136
  - temperature-graphing program, 135–139, 161–162
- Python (.py) files, 136
- Questions, research, 118
- Quota sampling, 120
- Random library, 21, 36
- Random number generator, 25, 31
- Random walk algorithm, 73–74, 74
- Rational numbers, 81
- Raw data, context of, 121–122, 122
- Recursion, 13. *See also* Labyrinths; Mazes
- Reflection, 111–115
- Resnick, Mitchel, 10
- Rich tasks, 3–4, 146–147
- Right bearing agents, 78, 85
- Roman labyrinths, 57, 67–69, 70
- Rowling, J. K., 13, 43, 48
- S\_key function
  - basketball program, 103
  - ping-pong program, 113
- Sampling procedures, 120
- Scaffolds, 64
- Scalability, 10
  - in Andrade-style art, 23
  - in temperature graphing, 130
- School funding, need for solutions on, 117
- Schuster, Carl, 59
- Scientific thinking, 2
- Scores, basketball program, 110–111



- Scratch, 2, 13–14  
 Albers-style art algorithm, 29–30, 30  
 Andrade-style art algorithm, 26  
 critical code reading capacities, 10  
 Kandinsky-style art algorithm, 17, 18  
 in middle and high school classrooms, 8  
 online tutorials for, 16  
 simulating laws of physics in, 13  
 vector program, 98, 99
- Sequencing  
 in Andrade-style art, 23  
 definition of, 9  
 in Kandinsky-style art, 16  
 in literature graphs, 43
- Set\_major\_locator function, 138
- Set\_x\_speed function, 114
- Set\_ylim function, 138
- Sets, data, 81–82
- Seven-circuit Cretan labyrinths, 62–64, 63, 65
- Shapes, in art. *See* Algorithmic art
- Shoot function, 110–111
- Show function, 139
- Singmaster, David, 86
- Skateboard learning activities  
 Newton's first law of motion, 94  
 Newton's second law of motion, 100  
 Newton's third law of motion, 101
- Snapchat, 41
- Snapp, Robert R., 78
- Social interaction, knowledge constructed through, 5
- Social network graphs, 41–42, 52–55, 54
- Sousa, David A., 5
- Spain, labyrinths in, 59
- Species extinctions, need for solutions on, 117
- Speed, calculation of, 94
- Spider's Web Morning Greeting, 42
- Spirals, 57, 58
- Spreadsheets, processing temperature data with, 131–135, 132–136
- Sprites, Bumper Rocks program, 101–102
- State space  
 of Alcuin's puzzle, 86–92, 88–89  
 definition of, 86
- State-transition graphs, 90, 91
- Status ordering, 6–7
- STEM (science, technology, engineering, and mathematics), women and minorities in, 7–8
- String variables, 21
- Struggle, productive, 5–6, 151
- Subgraphs, 46–47
- Subplot function, 138
- Subroutines, 9, 93  
 in basketball program, 108–111  
 in Bumper Rocks program, 101–107  
 in ping-pong program, 112–115  
 in Vasarely-style art, 33–38
- Suetonius, 69
- Tarry, Gaston, 78
- Tarry's algorithm, 78
- Tasks, rich, 3–4, 146–147
- Teamwork, 1, 6–7
- Technological literacy, 2, 7–8
- TeleGraph learning activity, 55
- Telephone challenge, 163n4
- Temperature analysis algorithm, 129–131
- Temperature-graphing program (Python)  
 best-fit line, 139–141, 140  
 displaying graph in, 139  
 full program, 161–162  
 graphing data in, 138  
 Python setup, 136  
 reading from data files in, 137
- Temperature trend analysis, 124–139  
 best-fit line, 131–132, 139–141, 140  
 classroom weather stations, 119  
 data processing with Python, 135–139  
 data processing with spreadsheets, 131–135, 132–136  
 line graph and scatter plot of data, 130, 131  
 NOAA Climate Data Online dataset, downloading, 125–129, 127–129  
 programming skills used in, 130  
 temperature analysis algorithm, 129–131  
 transference of skills to different datasets, 142–143
- Theseus (Greek legend), 60, 66, 67–68
- Three-circuit Cretan labyrinths, 64
- Transdisciplinary integrative approach, 12
- Transference of skills, 10, 142–143
- Trémaux, Charles Pierre, 75
- Trémaux's algorithm, 75–78, 76  
 applying to graphs, 83, 85  
 solving combinatorial puzzles with, 85–92, 86–91  
 threading mazes with, 79–81
- Truman, Harry S., 120, 120
- Twenty-first century, learning for  
 adapting and applying to new situations, 7  
 learning from mistakes, 5–6  
 problem solving, 4–5

- productive struggle, 5–6
- proficiency in technologically prolific environments, 7–8
- real-life contexts, 3–4
- rich tasks, 3–4, 146–147
- team settings, 6–7
- Twitter, 41
- Tyson, Neil deGrasse, 7
- Unicursal labyrinths. *See* Labyrinths
- Universal gravitation, law of, 107
- Unjustified conclusions, 124
- Unordered collections, 81
- Unplugged tasks
  - algorithmic art, 16, 17, 29, 33
  - evidence graphs, 51
  - incorporating computational thinking with, 151
  - laws of physics, 107, 108, 111
- Up\_key function, 113
- Variables
  - in basketball program, 108
  - in Bumper Rocks collision program, 101
  - in ping-pong program, 112
  - in Scratch vector program, 98
  - string, 21
  - in temperature-graphing program, 137–138, 141
- Vasarely, Victor, 13, 15, 33
- Vasarely-style art, simulating
  - algorithm design, 33
  - Codesters algorithm, 34–38
  - learning activities, 36, 37
  - programming skills used in, 33
  - unplugged tasks, 33
- Vectors, 94–98, 97, 99
- Velocity
  - Codesters velocity program, 97, 97
  - equation for, 108
  - vectors, 94–98
- Vertices, 82–83
  - definition of, 41–42
  - degree of, 45
  - for literature graphs, 43, 44, 48
  - for social network graphs, 42–43
- Video games, laws of physics in, 13, 93.
  - See also* Motion, laws of
  - acceleration, 99–101, 107–111
  - action-reaction pairs, 100–101
  - basketball program, 108–111, 109, 158–159
  - Bumper Rocks collision program, 101–107, 102, 156–158
  - displacement, 94–98, 108
  - dynamics, 99–100
  - inertia, 94, 95
  - learning activities, 94, 100, 101, 107, 108, 110, 111
  - ping-pong program, 112–115, 159–161
  - reflection, 111–115
  - vectors, 94–98, 97, 99
  - velocity, 94–98, 108
- Villa Cadolini (Cremona), labyrinth in, 68, 70
- Virgil, 69
- Voogt, Joke, 11
- Vygotsky, Lev S., 5
- W\_key function
  - basketball program, 103
  - ping-pong program, 113
- Wall following, 73–74, 74
- Water balloon drop, 108
- Wealth distribution inequality, need for solutions on, 117
- Weather data analysis. *See* Climate data analysis
- Weather stations, classroom, 119
- Websites, CSUNPLUGGED, 14
- Wiggins, Grant, 146
- Wing, Jeannette, 1
- Women, in STEM, 7–8
- x\_speed, 104, 114
- y\_speed, 104
- Youth-Led Participatory Action Research (YPAR) website, 124
- Zigzagging line art. *See* Andrade-style art, simulating











© 2021 Massachusetts Institute of Technology

All rights reserved. No part of this book may be reproduced in any form by any electronic or mechanical means (including photocopying, recording, or information storage and retrieval) without permission in writing from the publisher.

The MIT Press would like to thank the anonymous peer reviewers who provided comments on drafts of this book. The generous work of academic experts is essential for establishing the authority and quality of our publications. We acknowledge with gratitude the contributions of these otherwise uncredited readers.

This book was set in Stone Serif and Stone Sans by Westchester Publishing Services.

Library of Congress Cataloging-in-Publication Data

Names: Neumann, Maureen D., author. | Dion, Lisa (Computer scientist), author.

Title: Teaching computational thinking : an integrative approach for middle and high school learning / Maureen D. Neumann and Lisa Dion with Robert Snapp.

Description: Cambridge, Massachusetts : The MIT Press, 2021. | Includes bibliographical references.

Identifiers: LCCN 2021000766 | ISBN 9780262045056 (paperback)

Subjects: LCSH: Computer science—Study and teaching (Secondary) | Critical thinking—Study and teaching (Secondary)

Classification: LCC QA76.27 .N474 2021 | DDC 004.071—dc23

LC record available at <https://lcn.loc.gov/2021000766>