# 5

# DESIGNING AN ASSESSMENT OF COMPUTATIONAL THINKING ABILITIES FOR YOUNG CHILDREN

*Emily Relkin and Marina Umaschi Bers*

## What is Computational Thinking (CT)?

In 1963, computer scientist Alan Perlis observed that computer programming required exercising special classes of logical and creative thought processes. Perlis saw value in developing these thinking abilities early in life and recommended that computer programming become part of everyone's education (Grover & Pea, 2013; Perlis, 1963). Two decades later, mathematician, computer scientist and educational innovator Seymour Papert briefly mentioned the term "Computational Thinking" (CT) in his foundational book *Mindstorms: Children, Computers, and Powerful Ideas* (Papert, 1980). Later in his career, Papert became convinced that the acquisition of CT skills empowered children and adults to process and create knowledge of their world (Papert, 1996).

In the early 2000s, Jeanette Wing popularized the term CT in an article in which she defined it as solving problems, designing systems, and understanding behavior by drawing upon the concepts of Computer Science (Wing, 2006). Wing viewed CT as a vital and universal skill set that should be acquired by everyone (Wing, 2008). According to Wing, CT was not only of value in computer science but provided opportunities to succeed in the arts, humanities, and everyday life (Wing, 2008). Hemmendinger, (2010) stated that the goal of CT education should not be to have everyone thinking like a computer scientist but to facilitate real-world problem solving. Past studies have shown that children as young as 4 years old can begin to acquire CT skills (Bers, 2018; Bers, 2008; Leidl, Bers, & Mihm, 2017; Sullivan, Bers, & Mihm, 2017).

There is a paucity of information about the development of CT abilities in young children. The term is still evolving and there is currently no agreed upon age or stage of development at which children should be introduced to CT.

Mioduser and Levy (2010) further studied the effect of acquiring programming skills on the explanations that children give when they observe the behavior of robots. They found that as children gained experience in programming robots, they applied different descriptions to the robotic behaviors they observed. When the robot's behaviors were simple, the children with less experience in programming tended to ascribe human or animal-like motivations to the robot while those who were more experienced gave more mechanistic explanations. When children observed more complex programs, they typically give a combination of both types of explanations. In this study, construction tasks were designed to increase in complexity as the child learned. In addition, the researchers did not help the child beyond prompting, such as, "Why do you think that happened?" The researchers recorded the amount of prompting the children required (Mioduser & Levy, 2010).

Using children's explanations of robotic behaviors as a measure of CT has some assets but also limitations. The K-12 Computer Science Framework suggests that assessments of programming should involve not only the students' ability to write a program but also their ability to explain the significance of it (K-12 Computer Science Framework Steering Committee, 2016). Braitenberg (1984) argued that it is more difficult to understand a robot's behavior through observing a program than it is to understand the behavior by creating a program. Young children's expressive language skills may also limit the nature of the explanations they provide for the behaviors they observe. Mioduser and colleagues critiqued their study by stating, "While deepening how we understand young children's evolving knowledge of autonomous artificial behaviors, it is limited in its small sample and disconnect from classroom situations" (Mioduser et al., 2009, p. 19). They also observed that combining the task of robot construction and explanations in the same subjects introduced a potential confound and recommended that future studies examine these separately.

## *Design Considerations for a CT Assessment Tool*

Based on the work in this area carried out by past investigators, various considerations for the design of a valuable CT assessment tool for young children can be identified:

1. **Age appropriateness:** In order to optimally impact the acquisition of CT abilities, interventions need to target preschool and early elementary school levels. At this age, children are in the process of developing representational and abstract thinking and have sufficient linguistic skills to perform basic programing. An assessment tool for this age group must use age-appropriate language and tasks to assure that abilities and factors such as manual dexterity and attention span and the use of jargon (Sattler, Dumon, & Coalson 2016) are not the limiting factors in measurements.

2. **Authentic interaction:** To place young children at ease and avoid stress that can be associated with formal standardized testing, a CT assessment tool should be structured as authentically as possible, ideally capturing the dynamics of play and familiar teacher-child, parent-child, and/or peer to peer interactions. Authentic assessment should be something that is worthwhile for the child (Dewey, 1938). Shaffer and Resnick (1999) describe authenticity in assessment as relevant to the learning process. They also argue that computers and technologies can lead to new types of authentic assessment and learning if used correctly. The assessment should not be intrusive or disruptive and should include context specific prompts (Ming, Ming, & Bumbacher, 2014).
3. **Ease of administration:** Past researchers studied children's CT over long periods of time requiring three or more hours of time per child (Mioduser & Levy, 2010; Mioduser et al., 2009; Werner et al., 2014). Many educational assessments currently are too lengthy and/or complex for teachers to give effectively. Teachers continue to feel unprepared to conduct high-level assessments and many have a low assessment literacy despite educational efforts (DeLuca & Bellara, 2013). The National Research Council indicates that it is difficult to assess CT and recommends that teachers must be extremely skilled in coding through professional development in order to properly assess children (National Research Council, 2010). Ideally, teachers who are not particularly skilled at coding and assessments should be able to administer CT assessments.
4. **Time constraints:** The duration of testing sessions should be kept relatively short in light of the limited attention spans of children in this age group and the limitations on time available to teachers for individualized assessments (Moyer & Gilmer, 1953). To be of practical value for preschool and early elementary school use, the instrument should require no longer than 15–30 minutes for a single assessment.
5. **Sensitivity:** Following the "low floor, high ceiling" model (Papert, 1980), an ideal assessment tool should be equally useful for assessing novices as well as experts (Sattler, 2014).
6. **Scoring:** To create an assessment tool that does not require an expert to administer or score, the ratings system employed should use simple outcome categories and/or numeric scores that are straightforward to calculate (Koretz, McCaffrey, Klein, Bell, & Stecher, 1992).
7. **Communication of results:** While numeric scores are suitable for rating the level of CT proficiency, descriptive names (e.g., "Early Programmer") based on the score may better convey the meaning. It is important to present results in a fashion that is easy for people unfamiliar with CS concepts but can still give quality technical data (Sattler, Dumont, & Coalson, 2016).

## Development of a Pilot CT Assessment Tool for Young Children

To address the above criteria, we set out to create a pilot CT assessment tool that permits standardization and increased sensitivity, by presenting children with specific structured scenarios rather than simply observing free play with the robot. Structured scenarios permit assessments to be carried out in children who have only rudimentary understanding of the robot's operation and limited manual dexterity. To capture the full range of CT abilities, the scenarios are presented in gradually increasing complexity, from simplest to most complex. The assessment includes a combination of questions and simple coding tasks designed to create a balance between the need for well-developed language skills and non-verbal communication. Scoring and categorizations are sufficiently flexible so that missing one question or failing to complete one task does not disqualify the child from achieving a rating appropriate to their skill level.

We based our assessment on the "Seven Powerful Ideas" of CS that according to Bers (2018) inform curriculum that can promote CT in early childhood and that are developmentally appropriate. These are further described in Table 5.1.

**TABLE 5.1** Seven Powerful Ideas of Computer Science (Based on Bers, 2018)

| *Powerful Idea* | *Definition* | *Example* |
|---|---|---|
| Algorithms | Sequencing/order, logical organization | Child understands that KIBO blocks must be scanned in a specific order |
| Modularity | Breaking up larger task into smaller parts, instructions | Child uses repeat blocks in order to accomplish a goal rather than scanning a large number of blocks |
| Control Structures | Recognizing patterns and repetition, cause and effect | Child recognizes that he or she must use a beginning and an end when making a program and are able to use If Blocks and Repeat Blocks |
| Representation | Symbolic representation, models | Child sees the difference between the blue motion blocks and the orange sound blocks |
| Hardware/ Software | Smart objects are not magical, objects are human engineered | Child describes what the function of KIBOs electronics do. Child understands that one must give the robot a program in order for it to work |
| Design Process | Problem solving, perseverance, editing/ revision | Child has the capability to plan and test an idea in order to improve a project |
| Debugging | Identifying problems, problem solving, perseverance | Child identifies a bug in either hardware or software and is able to fix the problem |

Past researchers empirically identified stages in the development of programming skills and related those to acquisition of CT skills (Jenkins, 2002; Rogalski & Samurçay, 1990). For this purpose, we chose the Developmental Model of Programming proposed by Vizner (2017). One advantage that this model offered for our prototype assessment was it was developed using the KIBO robot platform based on observation of children 4–7 years old with various levels of skill and exposure to the platform. This model consists of four stages of programing and provides some parameters for assessing children's coding proficiency.

The four levels of proficiency specified in Vizner's Developmental Model of Programming are as follows:

1. **Proto-Programmer:** Child has little to no understanding of what a program is; child does not create his or her own code and may press the "On" button repeatedly without first programming the robot.
2. **Early Programmer:** Child is capable of creating programs with the Begin and End blocks. The child may try to use as many blocks as possible and may scan blocks that are not part of a meaningful program sequence.
3. **Programmer:** Can use 3–6 instructions without using complex blocks such as Repeat. This child may debug a program using trial and error but needs assistance from others when creating programs that are complex.
4. **Fluent Programmer:** Solves 6+ instructional tasks and uses advanced debugging techniques.

## A Prototype Early Childhood CT Assessment: TACTIC-KIBO

Our pilot CT assessment tool called TACTIC-KIBO (**T**ufts **A**ssessment of **C**omputational **T**hinking **i**n **C**hildren—**KIBO** robot version) is specific to the KIBO robotics platform which was designed to teach coding to children ages 4–7. The KIBO robotics platform was chosen because it is time-tested and developmentally appropriate. However, it is our hope that this approach may be applicable to many other programming and robotic technologies. The KIBO robotic platform is used widely in 54 countries throughout the world as a means of teaching CT and coding skills to young children. KIBO was developed under a grant from the National Science Foundation (NSF Grant No. DRL-1118897) as a developmentally appropriate tool for teaching the basics of computer programing to neurotypical children aged 4–7 (Sullivan, Elkin, & Bers, 2015).

To use the KIBO robot, children must align and scan barcodes on programming blocks representing steps in a program that guides the robot's actions (See Figure 5.1a). The child then presses a green flashing button to make the robot perform the program. Every program must have a "Begin" and "End" block. Other examples of programming blocks are "Spin," "Sing," "Turn Right," and "Shake." By virtue of resembling building blocks, programming blocks are in a

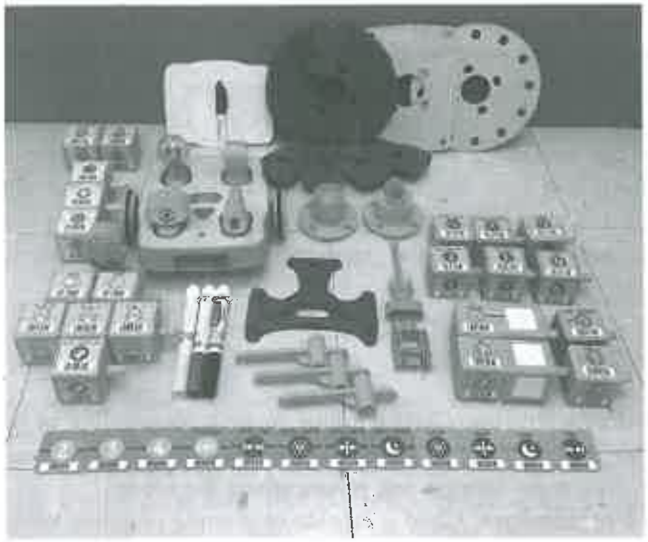

**FIGURE 5.1A** Photograph of Full KIBO Robotics Kit

format that is familiar to young children and arguably more manageable than computer screens that have only two dimensions.

The KIBO robotics kit is an exemplary educational robotics platform designed with a "low floor and high ceiling" (Papert & Harel, 1991) The programs that KIBO performs can be as simple as a three-block program and as complex as a program featuring conditional blocks, repeat blocks and nested statements. The more complex blocks (repeats and ifs) require "parameter" stickers which include "if near" "repeat until dark." (See Figure 5.1b) KIBO was designed to make young children producers of technology, not consumers. Children can have fun and express themselves as they make personally meaningful projects with KIBO.

The KIBO robot also has four openings on its upper surface for sensors that can detect light, sound, and proximity, as well as modules that can flash a light or record/play sounds. In addition, KIBO contains attachable art and building platforms compatible with LEGO bricks that allows children to use various materials to decorate their robot. There is also an attachable expression module that allows children to attach drawing using a dry erase board and marker extensions so that they can draw as KIBO moves, and a free throw extension to aid children in learning concepts of physics and math. This robot has been shown to help children learn technological literacy as well as other curriculum such as math, science, art, and language (Bers, 2018; Sullivan et al., 2015; Sullivan et al., 2017).

KIBO robot was selected as the initial robotics platform for CT assessment in children for several reasons. It is a time-tested, award-winning robotics platform designed for preschool to elementary school age children. It has an existing user base that spans multiple continents and languages, making it available to a large number of children and teachers. Its use of tangible programming blocks is not only advantageous for children but can help to create a more user-friendly environment for teachers or other evaluators to interact with the children they are assessing. KIBO uses programming principles that are analogous to those used in other robotic platforms for young children as well as platforms developed for older children and adults. It is hoped this will facilitate creation of versions of the assessment instrument that are applicable to other coding platforms in the future.

**FIGURE 5.1B** KIBO Robot Sensors

Bers' Seven Powerful Ideas were adapted as the domains for our prototype CT assessment tool and combined with Vizner's Developmental Model of Programming, which were adapted for use as a hierarchical scoring system. The design foundations of the TACTIC assessment tool are summarized in Figure 5.2.

The TACTIC assessment tool involves pre-programmed KIBO robot activities of escalating complexity which serve as the framework for asking questions and posing tasks for the child to complete. Figure 5.3 shows the block sequences and commands for all levels.

The examiner programs the KIBO in front of the child using the specified program and add-ons. This assessment can be done on a tablet or computer for automatic scoring or by paper for manual scoring. The examiner takes the child through the series of questions and tasks for each level. The test continues until the children reach a level that they get three or more questions or tasks incorrect, or until all four levels have been completed.

To help characterize TACTIC-KIBO in terms of its age appropriateness and face validity, a pilot study was carried out with participants from the Boston area. Fifteen kindergarten and early elementary school children between the ages of 5 and 7 with

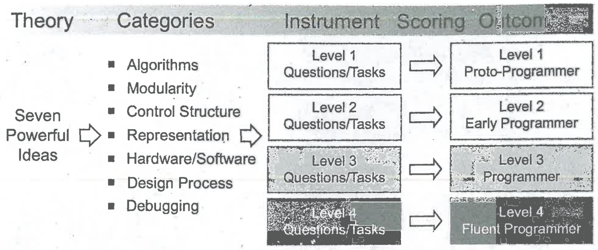

**FIGURE 5.2** The Design Foundations of the TACTIC-KIBO CT Assessment Tool

| Action | Blocks and parameter cards | Add-ons |
|---|---|---|
| LEVEL 1 BEGIN - FORWARD - END |  | |
| LEVEL 2 BEGIN - FORWARD - WHITE LIGHT ON - END |  |  |
| LEVEL 3 BEGIN - REPEAT TWO TIMES - FORWARD - WHITE LIGHT ON - END REPEAT - END |  |  |
| LEVEL 4 BEGIN - REPEAT UNTIL NEAR - FORWARD - WHITE LIGHT ON - END REPEAT - END |  |  |
| FINAL PROGRAM BEGIN - REPEAT IF NEAR - FORWARD - WHITE LIGHT ON - END IF - END |  |  |

**FIGURE 5.3** The Set of Robotic Activities Used in the Pilot TACTIC-KIBO CT Assessment

past exposure to the KIBO robot were recruited and consented using an Institutional Review Board approved protocol. The children were videotaped during the TACTIC-KIBO assessment and again as they engaged in structured interactive play sessions (IPS) with the KIBO. This IPS sessions allowed an independent assessment of CT skills to be assessed by expert raters.

The IPS was created as a means of measuring the validity of TACTIC-KIBO based on its correlation to expert assessments. The IPS had three parts, the first being a confrontational naming game used to test the child's knowledge of KIBO hardware. Then, a free-play construction session in which the child was encouraged to program a project of their own choice. Finally, the IPS included a construction challenge in which the child was asked to augment a program using higher level skills. The challenge was designed to bring the children to their ZPDs as described by Vygotsky.

A simple curriculum was created to guide the IPS with the goal of collecting sufficient information to permit an assessment of CT ability based on review of the IPS by independent expert raters. A scoring sheet for the IPS was developed to help standardize scoring by the outside raters. Experts rated 25% of the students using the same four-level classification system used for the CT assessment tool but based exclusively on the behaviors observed during the IPSs. The expert ratings of the IPS and the ratings based on the CT assessment tool were used to establish the inter-rater reliability. Demographic data (date of birth, gender, hours of experience with KIBO, experience with other robotics platforms, experience with programmable robots) were collected from the parent/legal guardian after the consent was signed.

## Pilot Study Outcomes

The primary outcome measure of the TACTIC-KIBO pilot study was the correlation between the TACTIC-KIBO ratings and IPS ratings by the primary examiner. Fourteen out of 15 children completed TACTIC-KIBO as well as IPS. Among the children who completed both assessments, there was a highly significant correlation between the total TACTIC scores and the expert rating of the IPS ($r= 0.895$, $p< 0.001$). Discrepant ratings occurred exclusively among four children rated as Level 4 Fluent Programmers by TACTIC who were judged to be Level 3 by IPS. The average administration time for TACTIC-KIBO was 16 minutes compared to 19 minutes for the IPS. The IPS, however, required an additional 15–30 minutes for scoring, while TACTIC-KIBO was scored as it was administered.

Using pre-specified criteria for administration and scoring, TACTIC-KIBO identified four levels of CT abilities ranging from novice to fluent in kindergarten and first-grade children. TACTIC-KIBO scores correlated significantly with expert assessments based on observation of KIBO IPSs (See Figure 5.4). According to past research and recommendations, administration time and ease of scoring were suitable for use in classroom and research settings (Moyer & Gilmer, 1953; Ruff & Lawson, 1990). These results suggest that TACTIC-KIBO is a promising means of assessing CT abilities in young children. Limitations to this study include the sample size, biases in subject selection, biases in test administration and ratings, context, and issues relating to the lack of availability of a true gold standard for measuring CT in young children.

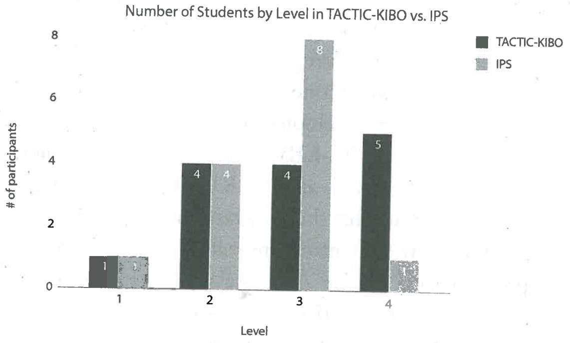

**FIGURE 5.4** Comparison between TACTIC-KIBO Level Scores and Expert Rating Scores from KIBO Interactive Play Sessions

We observed that TACTIC-KIBO was engaging and enjoyable for the majority of children tested. This included students who were not yet fully literate, and for whom English was a second language. TACTIC-KIBO duration proved suitable for 5–7-year old children despite the relatively short attention spans of children in this age group.

A caveat in the interpretation of the data on mean number of correct responses per level is that the method of administration used in this pilot allowed children to advance to the next level of questions only if they successfully responded in three or more categories on the preceding level. The assumption in the analysis of correct responses is that children would not be able to respond correctly on higher levels if they did not meet the three-correct response criterion on the lowest levels. It is possible that this approach underestimated the total number of correct responses children may have provided in a given category.

## *Future Directions*

TACTIC-KIBO requires further testing and validation before it can be recommended for use in schools and research studies. This should include testing a broader range of ages (4–7 years), experienced and inexperienced robot programmers, all genders, and children of different socioeconomic backgrounds and nationalities.

In the next phase of development, we anticipate testing will include administration by elementary school teachers. This will require the creation of a training program for educators and a certification process to assure that they meet acceptable standards of inter-rater reliability. A training curriculum along with annotated videos of TACTIC-KIBO being administered can be used for these purposes. A software application to help in the administration and scoring of TACTIC-KIBO has been developed. This will allow for easier administration by employing automated prompts, feedback, and permitting remote data collection.

In its current form, TACTIC-KIBO requires a child be taken out of class to be tested individually. Although individual assessments out of the classroom are currently a common practice in preschool and elementary school classrooms, it may be more beneficial and less disruptive to adapt this study design so that it could be given to an entire class at one time. A version of TACTIC suitable for in-classroom is currently under development.

In a recent pilot study, TACTIC-KIBO was applied to assessing CT in young children longitudinally before and after a KIBO robotics coding enrichment program. Results indicate that after the program, most children increased in their level by at least a level. In the future, repeated administrations could help to establish the reproducibility of the CT assessment. If repeated administration results in stable scores, TACTIC-KIBO may prove useful for longitudinal assessment. However, if repeated administration is associated with a learning effect that causes scores to drift between testing session, it may be necessary to create alternative forms with equivalent questions.

TACTIC-KIBO was designed around the KIBO robotic platform. However, there are many other robotic toys and programming games existing and under development for young children as well as adolescents. The question of the generalizability of methods used to assess CT in this study may be best addressed by developing alternative forms of TACTIC that are applicable to other platforms. Creating alternative forms for other platforms would be useful in developing intuitions about lines of questioning that are more or less universal across platforms.

To date research has focused on CT assessments for young children who are neurotypical. It is important to understand typical development of CT abilities to obtain normative data on CT assessment. Once sufficient normative data are obtained, it is important to extend the application to a more diverse population, including children with Autism Spectrum Disorder (ASD). There is evidence that children with severe ASD are engaged by the KIBO robot and that they may become more communicative with their teachers as a consequence of playing with and programming the robot (Albo-Canals et al., 2018). In light of this, it would be interesting to study the acquisition of CT abilities in such children and compare their development with that of neurotypical children.

Our initial experience with TACTIC-KIBO leads us to conclude that assessment of CT abilities in young children is a viable and promising area of future research. It is critical to continue to develop appropriate CT instruments for use in classroom and research settings in order to move CT educational initiatives forward. Through assessment, researchers can understand the development and acquisition of CT abilities, which in turn help in the creation of better technology platforms for young children. By bringing validated assessment tools to the classroom, early childhood educators can be empowered to implement developmentally appropriate CS curricula that help children to learn to code and reason in ways that can have lifelong benefits.

## Acknowledgments

The KIBO robotics kit was created under a grant from the National Science Foundation (NSF Grant No. DRL-1118897). A very special thank you to the members of the DevTech Research Group at Tufts University, Dr. Martha Pott, and Dr. Brian Gravel, as well as the teachers and children that made this research possible.

## References

Albo-Canals, J., Martelo, A. B., Relkin, E., Hannon, D., Heerink, M., Heinemann, M., … Bers, M. U. (2018). A Pilot Study of the KIBO Robot in Children with Severe ASD. *International Journal of Social Robotics*, *10*(3), 371–383. http://dx.doi.org/10.1007/s12369-018-0479-2

Bers, M. (2008). *Blocks to robots: Learning with technology in the early childhood classroom*. New York, NY: Teachers College Press.

Bers, M. U. (2018). *Coding as a playground: Programming and computational thinking in the early childhood classroom*. New York, NY: Routledge.

Braitenberg, V. (1984). *Vehicles: Experiments in synthetic psychology*. Cambridge, Massachusetts: The MIT Press.

DeLuca, C., & Bellara, A. (2013). The current state of assessment education: Aligning policy, standards, and teacher education curriculum. *Journal of Teacher Education, 64*(4), 356–372. http://dx.doi.org/10.1177/0022487113488144

Dewey, J. (1938). *Experience and education*. New York, NY: Collier Books.

Gadanidis, G. (2017). Five affordances of computational thinking to support elementary mathematics education. *Journal of Computers in Mathematics and Science Teaching, 36*(2), 143–151.

Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher, 42*(1), 38–43. http://dx.doi.org/10.3102/0013189X12463051

Hemmendinger, D. (2010). A please for modesty. *ACM Inroads*, 1(2), 4–7. http:/dx.doi.org/ 10.1145/1805724.1805725

Jenkins, T. (2002). *On the difficulty of learning to program. Proceedings of the 3rd Annual. Conference of the LTSN Centre for Information and Computer Sciences*, 53–58. Leeds. Retrieved from http://www.psy.gla.ac.uk/~steve/localed/jenkins.html

K-12 Computer Science Framework Steering Committee. (2016). *K–12 computer science framework*. Retrieved from https://k12cs.org

Koretz, D., McCaffrey, D. F., Klein, S. P., Bell, R. M., & Stecher, B. M. (1992). *The Reliability of scores from the 1992 Vermont portfolio assessment program: Interim report*. Santa Monica, CA: RAND Corporation.

Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J. … . Werner, L. (2011). *Computational thinking for youth in practice. ACM Inroads, 2*, 32–37.

Leidl, K., Bers, M. U., & Mihm, C. (2017). Programming with ScratchJr: A review of the first year of user analytics. *In the proceedings of the International Conference on Computational Thinking Education*, Wanchai, Hong Kong, 116–125.

Lockwood, J., & Mooney, A. (2018). Computational thinking in education: Where does it fit? A systematic literary review. *International Journal of Computer Science Education in Schools, 2*(1) 41–60. https://doi.org/10.21585/ijcses.v2i1.26

Massachusetts Digital Literacy and Computer Science (DLCS) Curriculum Framework. (2016). *Massachusetts Department of Elementary and Secondary Education*. Retrieved from http://www.doe.mass.edu/frameworks/dlcs.pdf

Ming, V., Ming, N., & Bumbacher, E. (2014) Aligning learning with life outcomes through naturalistic assessment. *Socos LLC white paper*. Retrieved from http://about.socoslearning.com/socoswhitepaper.pdf

Mioduser, D., & Levy, S. T. (2010). Making sense by building sense: Kindergarten children's construction and understanding of adaptive robot behaviors. *International Journal of Computers for Mathematical Learning, 15*(2), 99–127. https://doi.org/10.1007/s10758-010-9163-9

Mioduser, D., Levy, S. T., & Talis, V. (2009). Episodes to scripts to rules: Concrete-abstractions in kindergarten children's explanations of a robot's behavior. *International Journal of Technology and Design Education, 19*(1), 15–36. https://doi.org/10.1007/s10798-007-9040-6

Moyer, K., & Gilmer, B. V. H. (1953). The concept of attention spans in children. *Elementary School Journal, 54*(1), 464–466.

National Research Council. (2010). Report of a workshop on the scope and nature of computational thinking. *The National Academies Press*. Washington, DC: Joseph Henry Press.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York, NY: Basic Books.

Papert, S. (1996). An exploration in the space of mathematics educations. *International Journal of Computers for Mathematical Learning, 1*(1), 95–123.

Papert, S., & Harel, I. (1991). *Situating constructionism. Constructionism*. New York, NY: Ablex Publishing.

Pei, C. (Yu), Weintrop, D., & Wilensky, U. (2018). Cultivating computational thinking practices and mathematical habits of mind in lattice land. *Mathematical Thinking and Learning, 20*(1), 75–89. https://doi.org/10.1080/10986065.2018.1403543

Perlis, A. J. (1963). The computer in the university. In M. Greenberger (Ed.). *Computers and the world of the future* (pp. 180–219). Cambridge, MA: MIT Press.

Portelance, D. J., & Bers, M. U. (2015). Code and tell: Assessing young children's learning of computational thinking using peer video interviews with ScratchJr. *In Proceedings of the 14th International Conference on Interaction Design and Children (IDC '15)*. Boston, MA: ACM.

Rogalski, J., & Samurçay, R. (1990) Acquisition of programming knowledge and skills. In J-M. Hoc, T. R. G.Green, D. J.Gilmore and R. Samurçay (Eds.) *Psychology of programming* (pp. 157–174). London: Academic Press.

Ruff, H. A., & Lawson, K. R. (1990). Development of sustained, focused attention in young children during free play. *Developmental Psychology, 26*(1), 85–93.

Sattler, J. M. (2014). *Foundations of behavioral, social, and clinical assessment of children*. La Mesa, CA: J.M. Sattler Publishers.

Sattler, J. M.Dumont, R., & Coalson, D. (2016) *Assessment of Children WISC-V and WPPSI-VI*. La Mesa, CA: J.M. Sattler Publishers.

Shaffer, D. W., & Resnick, M. (1999). "Thick" authenticity: New media and authentic learning. *Journal of Interactive Learning Research, 10*(2), 195–215.

Sullivan, A., Bers, M. U., & Mihm, C. (2017). Imagining, playing, & coding with KIBO: Using KIBO robotics to foster computational thinking in young children. *In the proceedings of the International Conference on Computational Thinking Education*, Wanchai, Hong Kong.

Sullivan, A., Elkin, M., & Bers, M. U. (2015). KIBO robot demo: engaging young children in programming and engineering. *Proceedings of the 14th International Conference on Interaction Design and Children (IDC'15)*. Boston, MA: ACM. http://doi.org/10.1145/2771839.2771868

Vizner, M. Z. (2017). *Big robots for little kids: Investigating the role of sale in early childhood robotics kits* (Master's thesis). Available from ProQuest Dissertations and Theses database (UMI No. 10622097).

Vygotsky, L. S. (1978). *Mind in society: The development of higher psychological processes*. Cambridge, MA: Harvard University Press.

Vygotsky, L. S. (1987). Thinking and speech. (N. Minick, Trans.). In R. W. Rieber & A. S. Carton (Eds.), *The collected works of L. S. Vygotsky* (Vol. 1., pp. 39–285). *Problems of general psychology*. New York: Plenum Press. (Original work published 1934).

Vygotsky, L. S. (1998). Infancy (M. Hall, Trans.). In R. W. Rieber (Ed.), *The collected works of L. S. Vygotsky: Child psychology* (Vol. 5., pp. 207–241). New York: Plenum Press. (Original work written 1933–1934).

Wang, D., Wang, T., & Liu, Z. (2014). A tangible programming tool for children to cultivate computational thinking. *The Scientific World Journal*, 2014(3). https://doi.org/10.1155/2014/428080

Werner, L., Denner, J., & Campe, S. (2014). Using computer game programming to teach computational thinking skills. In K. Schrier. (Ed.), *Learning, education and games: Curricular and design considerations* (Vol. 1) (pp. 37–53). Pittsburgh, PA: ETC Press. Retrieved from https://dl.acm.org/citation.cfm?id=2811150

Wing, J. (2006). Computational thinking. *Communications of Advancing Computing Machinery*, *49*(3), 33–36. New York, NY: Association for Computing Machinery. http://doi.org/10.1145/1118178.1118215

Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical transactions of the royal society of London. Mathematical, physical and engineering sciences*, 366 (1881), 3717–3725. http://doi.org/10.1098/rsta.2008

---

# 6

# ENGAGING YOUNG CHILDREN IN ENGINEERING DESIGN

## Encouraging Them to Think, Create, Try and T Again

*Pamela S. Lottero-Perdue*

Molly was focused. She was thinking and her hands were moving, pla wooden and foam building blocks deliberately on the table in front of her. W her hands stopped moving, she seemed satisfied, looked at me, and said: "The Molly had just finished her first try at an engineering design challenge to crea fence to contain Henrietta, a Hexbug Nano® robot about the size of a to brush head. Earlier, Molly and her kindergarten classmates giggled when they Henrietta vibrate and watched the robot move haphazardly on the classro linoleum floor. After explaining to me why she made the fence the way that did, Molly was eager to test it out. We turned Henrietta on, lowered her in the fence, and within about 10 seconds, Henrietta escaped. Molly caught H rietta as the little robot left the fenced-in area. "What happened?" I as prompting Molly to analyze what happened. Her analysis was excellent rooted in her earlier understanding of science concepts. I then asked: "So do have any ideas about how you can make your fence a little better for Henriet Molly-the-kindergarten-engineer shared her ideas—which were based on fi where the fence had failed the first time—and without hesitation began reconstruct the fence.

Molly (a pseudonym) was one of 53 kindergartners who participated research study about their engagement in an engineering design challenge opportunities to integrate scientific and mathematical knowledge as they engine (see Lottero-Perdue, Sandifer, & Grabia, 2017 for the science and enginee lesson). As the sole data collector on the project and co-teacher for the science engineering instruction that went along with it, I was able to get an in-depth loo individual children's thinking as they engineered. Normally, my observation young children, a term I will use in this chapter for children in grades kindergarten through grade two (PreK-2), doing engineering have been of stud