Early Childhood Education Journal (2023) 51:457–468
https://doi.org/10.1007/s10643-022-01319-0

Classroom Play and Activities to Support Computational Thinking Development in Early Childhood

Joohi Lee[1] · **Candace Joswick**[1] · **Kathryn Pole**[1]

Accepted: 19 January 2022 / Published online: 4 February 2022
© The Author(s), under exclusive licence to Springer Nature B.V. 2022

Abstract
This is a conceptual paper based on existing literature aiming to provide practical information on designing and implementing activities to promote children's computational thinking. Computational thinking is a relatively new term in early childhood education that refers to a specific problem-solving thinking process involving various logical and analytical thinking skills. Four foundational skills have been identified as core thinking skills of computational thinking: decomposition, abstraction, pattern recognition, and algorithm. We explain these four skills in this paper and their practical applications to teaching and learning in early childhood education. Early computational thinking skills are found in common early childhood activities. This paper identifies activities teachers can use in the classroom to explicitly promote children's computational thinking and provides a new perspective on how to adapt classroom activities to integrate computational thinking. In particular, we emphasize the need to vary the demands of the content in the activity and incorporate computational thinking based on children's needs and development to ensure that children progress through the thinking process.

Keywords Computational thinking (CT) · CT skills · CT practice · Algorithm · Mathematics · Literacy

> Computational Thinking is considered a universal competence, which should be added to every child's analytical ability as a vital ingredient of their school learning (Voogt et al., 2015, p. 714).

Introduction

Computational thinking (CT) is not an outcome, but a thinking process (Wing, 2008) which begins with identifying a problem and continues until the problem is successfully solved. It is a specialized process of "solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science" (Wing, 2006, p. 33). Lavigne et al. (2020) defined CT based on Wing's perspective as "a creative way of thinking that empowers individuals to be systematic problem-solvers, enabling them to identify problems, then brainstorm and generate step-by-step solutions that can be communicated and followed by computers *or* humans" (para. 2).

Computational thinking has become a critical core competency in the twenty-first century (Partnership for 21st Century Skills, 2009) to help children become creative and systematic problem solvers in a digital society (International Society for Technology in Education [ISTE], Computer Science Teacher Association [CSTA], & National Science Foundation, 2011). ISTE emphasizes the goal to help all students become computational thinkers who understand and control computing to innovatively solve problems (ISTE, 2021). The importance of CT has been well recognized in upper elementary through secondary education (e.g., Barron et al., 2011) and various efforts have been made to promote student CT by integrating coding or computer science into district curriculums from elementary through high school. In fact, CSTA and Code.org report that 33 states have implemented about 57 computer science (CS) policies as of 2018 and more states are considering implementing CS into their curricula as elective or mandatory courses (Modan, 2019).

Since CT has been considered to be a complex thinking process appropriate for older children for math and computer

✉ Joohi Lee
joohilee@uta.edu

Candace Joswick
candace.joswick@uta.edu

Kathryn Pole
kpole@uta.edu

1 Curriculum and Instruction, University of Texas at Arlington, Arlington, USA

science (Chongo et al., 2020), the foundations in very young children are currently lacking. The term "CT" is relatively new and an emerging concept in early childhood education, and little is known about how to promote children's CT in a classroom setting. When implementing CT activities in early childhood classrooms, it is important to consider the development of the children. It has been well affirmed that four- and five-year-old children learn best through concrete and hands-on play-based activities (Bers, 2018b; NAEYC, 2020; Lee, 2019; Piaget, 1957). Common activities and play can be adapted as ways to build their CT skills. When children engage in hands-on CT activities, they acquire early CT skills necessary to become tool creators in addition to become a tool user capable of solving complex problems creatively and efficiently (Sykora, 2021) by applying various skills.

In particular, four major thinking skills of CT have been identified by ISTE and CSTA: decomposition, abstraction, pattern recognition, and algorithm. Examples of decomposition activities appropriate for young children are puzzles, block play, planning an event, and identifying the separate steps of an activity such as handwashing. Examples of abstraction include "Simon Says" and "Who Am I" games and using prompt questions during story time to help children focus on the sequence of the story. Any sorting and patterning activities are the foundation of pattern recognition skill. Finally, helping children come up with a step-by-step solution to a problem or to complete a task promotes young children's algorithm skills. In this paper, we present activity and play ideas aligned with the four major CT skills and a framework for how to adapt common classroom activities to integrate computational thinking.

Computational Thinking as Critical Thinking Skills

In her seminal work on computational thinking, Wing (2006) not only coined the term "computational thinking" (which had been conceptually discussed by Papert, 1980) but notably said, "[t]o reading, writing, and arithmetic, we should add computational thinking to every child's analytical ability" (p. 33). She conceptualized CT as *fundamental* to computer science but not reserved for computer scientists. Lu and Fletcher (2009) claim that "CT is *not* about getting humans to think like computers, but rather about developing the full set of mental tools necessary to effectively use computing to solve complex human problems" (p. 1). That is, CT is a *set of mental tools* that allow for creative and systematic problem solving, which is necessary for computing but not unique to it (e.g., Hemmendinger, 2010; Yadav et al., 2011).

In other words, a computer is not needed for computational thinking, but computational thinking is required for effective and efficient computing. Both computing skills and coding are increasingly appearing in school curricula and standards (e.g., Barron et al., 2011; ISTE, 2016); CT is needed for both coding and computing. Further, foundational CT-related analytic and problem-solving skills are generally applicable beyond computing. While a discussion of computing in early childhood is beyond the scope of this paper, we note that computing is not only often cost-prohibitive, but also somewhat contentious (see National Association for the Education of Young Children [NAEYC], 2012). For these reasons, our approach to CT in early childhood presented in this paper is "unplugged," meaning that it does not rely on access to computer hardware to implement.

Foundational Computational Thinking Skills

Computational thinking is a whole thinking process of solving problems involving a system for identifying a problem, planning the steps to solve the problem, executing a plan, detecting errors, and fixing the errors to make the plan work. Solving a problem involves various thinking skills. Four major skills are salient when engaging in CT activities, starting from decomposition and going through the pattern recognition and abstraction processes, and finally creating an algorithm to solve a complex problem in an efficient way (Lynch, 2019). These four skills have been commonly accepted when explaining computational thinking skills (e.g., CSTA, ISTE, Code.org, Google, etc.). In this paper, we also use these four major skills as the foundation for early childhood activities. The following figure displays each of these skills with a brief description (Fig. 1).

Decomposition

Decomposition requires an analytic factorizing process of breaking down a complicated problem or system into smaller, more manageable pieces (Valenzuela, 2020). That is, decomposition is a *deconstructing* process:

> The ability to break down a large problem into smaller parts is important for many real-world tasks. To decompose a problem effectively, one must understand its constraints, generate potential solutions, and evaluate the strengths and weaknesses of those solutions. Importantly, these steps are often better taken before one acts; attempts to achieve a complex task without proper planning can lead to unnecessary effort to correct a mistake or even irreversible failure. (Dietz, et al., 2019, p. 1).

Abstraction is the filtering out of unnecessary or unneeded details for solving a problem. "Filtering out" is essentially a process by which information that might otherwise be a distraction from problem solving is ignored. This allows the most important and relevant information to be prioritized. Without abstraction, identifying how to solve the problem and solving it may be more challenging.

Pattern Recognition

Pattern recognition is the skill of sorting out similarities and differences or identifying patterns among and within problems. Problems are easier to solve when patterns are recognized; common patterns and their solutions can be applied to new problems that share the same patterns. Without pattern recognition, each problem is new and novel so that problem solving or information processing does not become more systematic and efficient. Further, without pattern recognition, sorting by characteristics or by similarities and differences is especially challenging. Characteristics—those features or qualities belonging to a group, thing, or person—are used to make sense of much of the world and to organize our thinking.

Algorithm

An *algorithm* is the step-by-step plan for solving problems. Algorithm design is the process of developing a step-by-step solution to a problem or the rules to follow to solve a problem. The steps of an algorithm represent the actions to be taken to solve a problem efficiently and effectively. This plan must be correctly ordered, have a clear start and end, include all necessary steps in-between, and contain all relevant information. Execution of the algorithm is also part of the algorithm skill.

Relationship Between Foundational Computational Thinking Skills

When problems are *decomposed*, *patterns* are identified and unnecessary information can be disregarded. These skills allow for a "making the problem easier" process—a problem is made smaller through the *decomposition process* that breaks it down to more manageable smaller problems. From the set of smaller problems, *patterns* of previously solved problems (and their solutions) might be found and applied. While looking for *patterns* within these smaller problems, it is also easier to recognize unnecessary information, that is, *abstraction*. Through *algorithms*, a blueprint or map or simple steps or rules can be developed and implemented to solve each of the smaller problems. With a set of solutions to the smaller problems, the original problem can be solved. In computing, the set of algorithms, when executed, are the program that solves a complex problem efficiently. Altogether, CT involves the understanding that good plans achieve a goal effectively and efficiently (Dietz et al., 2019).

Conceptual Perspectives

Approaches to Computational Thinking Development

A review of the literature reveals two primary approaches to supporting CT development in the classroom (Martin, 2018; Weintrop et al., 2016). One approach promotes CT as a distinct subject, developed through computer science courses or units of instruction. This approach, which has been adopted in many schools, requires scheduled time in the school day for computer science study, advanced teacher expertise, and resources specific to CT concepts (Merino-Armero et al., 2021). As part of this approach, "coding" has been widely used in education to promote children's CT development. In fact, many states have implemented coding in their curricula, requiring students from kindergarten through grade 12 to learn how to code (ISTE, 2017).

The other primary approach to supporting CT development is to integrate content with CT (e.g., Lee et al., 2019). From this perspective, the set of critical thinking skills or thought processes that comprise CT are elicited through problem-solving activities grounded in existing content and embedded into the overall school curriculum. Such integration of CT and content is commonly done in the science, technology, engineering, and mathematics fields. This approach has been found to be effective in kindergarten through grade 12 Computer Science education (Merino-Armero et al., 2021).

Integrated Approach to CT in Early Childhood

Children's development and learning are closely intertwined (NAEYC, 2009). That is, there is a reciprocal relationship between children's development and their learning (Institute of Medicine and National Research Council of the National Academies, 2015), as growth in one leads to growth in the other. Early childhood is an important time for young children to grow, play, and explore the world they live in through "active, meaningful engagement" (NAEYC, 2020, p. 9). Developmentally, it is a life stage characterized by genuine curiosity and desire for learning. Children are born with innate curiosity (Stephens, 2007), wanting to know about the natural world and the artificial world, the world of emotions and the world of ideas, the world by themselves and the world with others in social contexts. For young children to develop new knowledge, they need hands-on experiences to construct their learning (Bransford et al., 2000). In early childhood education curricula, an integrated approach to learning and development is preferred (NAEYC, 2009). Therefore, we adopt an integrated approach to incorporating CT skill development.

Research suggests that it is critical in early childhood to provide children with certain types of learning experiences to practice critical thinking skills (Ramey & Ramey, 1999). Lavigne et al. (2020) claim that CT is the one of those thinking skills to which children must be intentionally exposed to help them practice CT skills. They suggest implementing unplugged daily activities to promote young children's CT as appropriate in early childhood, such as setting up the table for guests following steps to promote algorithm skills (e.g., identifying the number of guests, covering the table with a tablecloth, gathering and placing plates, cups, and cutlery, etc.), discussing the sequence of steps for making a peanut butter and jelly sandwich, using scenario-based problems, and so forth. In addition, current CT-related literature in early childhood recommends unplugged activities integrating stories or children's literacy practices to tell major events of the story in a sequence (Lavigne & Wolsky, 2021; Lee & Jo, 2019).

In this paper, we also focus on unplugged activities and present the integration of computational thinking skill development in common classroom activities used in early childhood for development and learning; we contend that in early childhood, CT can be engendered through an integrated-with-existing-curriculum approach. These activities are concrete, hands-on, and play-based, and thus appropriate for children's development and interests. They are also unplugged, so they do not rely on any computer-based technology to promote CT. For example, we illustrate how CT is embedded in common classroom activities from handwashing to sorting toys and manipulatives and even planting seeds. Further, we explain how teachers can both adapt additional classroom activities to support children's CT skill development and differentiate those activities to meet the needs of all learners.

Common Early Childhood Classroom Activities Based on Foundational CT Skills

Targeted and focused opportunities to practice the foundational CT skills in a developmentally appropriate manner such as a play-like setting are necessary to promote children's CT. In this section, we present common early childhood classroom activities that engender each of the four foundational CT skills, organized by skill.

Classroom Activities Supporting Decomposition Skills

Practicing decomposition ultimately helps children solve complex problems efficiently by breaking the whole problem into smaller pieces. Some common early childhood activities to help children practice decomposition skills are described below, including puzzles and blocks, planning an event, and washing hands.

Puzzles and Blocks

Puzzles and blocks are common activities that support young children's mathematical thinking and spatial sense (Lee et al., 2009). They are also great resources for developing decomposition skills to strengthen children's CT development. The nature of puzzles enables children to see whole pictures and requires them to place decomposed pieces to make the whole by placing puzzle pieces into the right places. A puzzle itself is based on composition and decomposition principles. That is, a whole puzzle set breaks down to small pieces (decomposition) and small pieces add up to a whole puzzle set (composition). Puzzle complexity ranges from the number of total pieces to the configuration of each piece fitting uniquely into the frame to pieces fitting against each other to fill the frame.

In early childhood, children often play with blocks, such as when they build towers, tracks, and homes for their toys. When integrating picture cards of completed block buildings with block play activities, children see both the individual block pieces (the decomposed pieces) and a completed building in a picture (a composed set). Children can use picture cards to replicate buildings, finding the same pieces and putting them together as they see in the photo (Lee et al., 2015). This activity helps them experience the composition and decomposition processes.

Planning an Event

Planning any sort of event involves many steps, which are helpfully captured in a list. The list may include tasks like "write invitations" and "get snacks and party favors." Early childhood classrooms may plan holiday observances, classroom visit days for friends and family, or field trips into the community. Young children can help decompose these scenarios to create lists of manageable tasks. Splitting responsibility for tasks among groups of children can promote communication and collaboration as well as feelings of belongingness. For instance, a group of children may be responsible for preparing an area of the classroom that will welcome parents and friends to the room.

Literacy and reading can be incorporated into these activities as well. Writing invitations and thank you notes is a common classroom literacy practice that can be broken into smaller tasks and divided among children. The young children's book *Dragons Love Tacos* (Rubin & Salmieri, 2012) can serve as inspiration for a classroom taco party. The teacher reads the text and the class learns that taco shells, chicken or beef, cheese, tomatoes, and lettuce are needed for the tacos, but the dragons cannot have spicy salsa! For the party, children can decompose tasks to list entertainment (music), decorations (perhaps made by hand during art time), and food. Guests can come to the party and build their own tacos, enjoying the dragon-inspired decorations created by the children and displayed around the room. This activity promotes not only the *decomposition* CT skills, but also literacy and social and emotional learning.

Handwashing

When children are tasked to wash their hands, they must *decompose* the whole task into steps. A teacher may have whole-group discussions with children about why they wash their hands and facilitate discussions on how they should wash their hands. This discussion can be tailored to help children practice decomposition thinking skills: the steps should come from the children's discussions. A teacher may use guided questions to cover all five steps of washing hands recommended by the U.S. Centers for Disease Control (2021): wet, lather, scrub, rinse, and dry. Once children come up with the steps, they can create their own "how-to" posters with written and illustrated steps for handwashing. This functions as a resource that helps children visualize the decomposed tasks, and they can also use the poster to remind them what to do next as they wash their hands. Displaying children's posters in various locations in and outside of the class (e.g., on the classroom wall, in the restroom, at the entrance of the classroom) provides children with visual representations of decomposed tasks as well as the correct steps for washing their hands.

Classroom Activities Supporting Abstraction Skills

Abstraction is an advanced thinking skill that is often considered to be acquired during the elementary years. However, it is necessary to build foundational abstraction skills from early childhood. Abstraction enables children to ignore irrelevant information or details by focusing on important information to complete a task. The abstraction process helps children think about problems more easily by reducing unnecessary information or steps (Cansu & Cansu, 2019). Activities that support children's abstraction skills include "Simon Says" and "Who Am I" games. Story time also provides various opportunities to help children practice abstraction skills (e.g., having children retell the main events of a story).

"Simon Says" and "Who Am I?" Games

Playing "Simon Says" is a good way to help children practice focusing on important information and ignoring irrelevant information. When children hear "Simon Says," they follow the command Simon has given (e.g., "Simon says, jump 2 times!"). If a command is given without the "Simon says" preface, they should ignore the command. To incorporate mathematics into the game, "Simon" may ask children to tap their knees a certain number of times, for example. Adding a numeric value to the actions allows children to practice counting and incorporating movement with counting, which helps build important subitizing skills and awareness of the comparative size of numbers.

In "Who Am I?" or "I Spy" games, children pretend to be detectives looking for a person or object that has particular attributes. These games can also incorporate mathematics. For instance, the teacher may describe a hidden shape with three straight sides and three angles and ask children to identify what the shape is. Children can also be asked to find a shape in the classroom with the same characteristics before revealing the hidden shape. For example, the teacher may request that the children find a shape "with three straight sides," tasking them to find triangles around the room. Children practice paying attention to the important information to complete the tasks (e.g., finding shapes with certain attributes).

Interesting teacher-made scenarios help children focus on important information. For example, children may play "Who Am I?" by looking at the shape, size, and imprint of footprints to identify the origin of the prints. During this activity, as children refine their observations, they eliminate the choices that do not match the given attributes. For example, if the footprint is of a bare foot, then everyone wearing shoes can be eliminated. As children work through these scenarios, they develop abstraction skills.

Story Time

Early childhood classroom story time is filled with teacher-scaffolded abstraction. When reading, a teacher uses prompting questions to help children focus on the main story of the book. For example, along with reading *The Very Hungry Caterpillar* (Carle, 1969), a teacher may ask, "What did the caterpillar eat on Monday?" "Tuesday?" "Wednesday?" and so forth. These probes enable children to focus on the main events of the story. Prompts play an important role during reading to help children practice their abstraction thinking skills.

Illustrations that draw children's attention to a book as they grapple with the interactions between written text and illustrations, which rarely tell quite the same story (Wolfenbarger & Sipe, 2007), also help build ability in abstraction. As young children make meaning from interactions, they become active meaning-making agents by sorting, combining, and reconciling the information coming from words and illustrations (Sipe, 1998). Teachers can facilitate this development by leading deep conversations about books and by using tools such as anchor charts with pictures of main events that help children focus on important information from the book. The events presented in the chart can be pre-selected by the teacher or co-constructed with children. As children become familiar with anchor charts, they can create their own charts of the books read in the classroom and organize them by various traits (e.g., genre, settings, characters). This will promote children's abstraction thinking skills as they select important information from books shared in the classroom.

Classroom Activities Supporting Pattern Recognition

Pattern recognition is a skill that involves sorting out similarities and differences or identifying patterns among and within decomposed small pieces of the larger problem (Lee & Jo, 2019). Pattern recognition is critical for helping children process information by organizing it to solve a complex problem more effectively based on their previous experiences of recognizing patterns. For example, when children know how to tie a shoelace, the task of tying a ribbon is comparatively easy. The following presents early childhood activities that support children's pattern recognition based on sorting and patterning activities.

Sorting and Patterning

Many early childhood classroom activities include sorting and patterning. For instance, when given plastic chain links or blocks or connecting cubes of multiple colors, children can sort them into groups of same colors or create chains of repeating color combinations. Color is not the only sorting or patterning characteristic to use, of course. Mathematical sorting often happens when children are asked to sort shapes by their attributes or characteristics, such as size, form, and function. Sorting items helps children attend to the characteristics that are important (e.g., the number of sides of shapes) and those that are not (at least not for that particular sort, e.g., the size of the shape). Patterning, like that done with a series of colored blocks or a series of shapes, can be repeated multiple times. Doing so requires children to attend to precision, making sure their repetitions match those previously made. Mathematical patterning in early childhood predicts later mathematical achievement even more strongly than counting (Rittle-Johnson et al., 2016). Mathematical patterning may involve sequencing numbers, counting aloud every other number, or repeating arrangements and sequences of items, like shapes.

In early childhood classrooms, cleaning up is a common activity performed between events like playing with toys indoors and snack time. Clean-up time provides children with opportunities to practice important CT skills focusing on patterns by applying sorting and patterning skills. When it is time to clean up, children sort play materials and place them in their assigned places. For example, children place the same types of blocks in a specific container or shelf: LEGO® blocks go to the LEGO® block container and unit blocks go to the unit block container. This is a simple everyday activity that helps children organize and process information by recognizing patterns.

Daily Routines

Displaying and reviewing daily routines helps make children aware of patterns of daily routine sequences (e.g., what to do). Children's days revolve around patterns and sequences—arriving at school, participating in story time, washing hands, having a snack. When children know the pattern of their daily classroom or home routines, they know what to expect next. Teachers can support young children's development of pattern recognition through the patterns of fixed daily routines, displaying visual routine agendas and reviewing them with children. In addition to patterns found in daily routines, calendar time is full of patterns, such as days of the week, and many stories feature patterns, such as those found in cumulative tales (e.g., *The House that Jack Built (*e.g. retold by S. Taback), *The Enormous Turnip* (e.g. retold by B. McBeath), and *The Mitten (*e.g. retold by J. Brett).

Classroom Activities Supporting Algorithm Design Skills

Algorithm design is the process of developing a step-by-step solution to a problem or the rules to follow to solve a problem (Lee et al., 2021). Algorithm is the final stage of decomposition, pattern recognition, and abstraction in which a set of rules is created to solve a problem. Algorithm is closely related with decomposition in that it involves a set of decomposed steps; however, the purpose of algorithm design is to create the most efficient procedure by arranging those decomposed steps (Lee, 2020). Activities such as making a taco, setting a table for guests, tying shoelaces, and treasure hunts support children's skills for algorithm design.

Making Tacos and Setting and Serving a Table

In the planning a party activity previously mentioned, tacos would be served. Extending this activity to encompass *algorithm design* can include spending literacy time writing a "how-to" that instructs party guests on how to assemble their own tacos. Young children enjoy assembling their own tacos, learning what happens when they don't follow the algorithm (it's hard to skip the shell!), and advising their guests on how to make the best taco. Setting a table for snack or mealtime or the taco party and then serving the food also supports children's algorithm design. They need to think through the order in which they will serve items and how they will set the table. For instance, the cups will be easy to distribute if they are empty and then filled at the tables. Napkins should be placed before the utensils, which hold those napkins down and then don't themselves touch the table. To add a mathematical emphasis to these common early childhood classroom activities, ask students to identify how many of each setting they'll need so that they count out the number of people at their table and recount that same number when getting the settings.

Tying Shoelaces and Treasure Hunts

Tying shoelaces and treasure hunt activities are good ways for children to practice algorithm skills. Tying a shoelace is a complex task for young children, as they must follow specific and exact steps to successfully tie their shoelaces. As there are several methods of tying shoelaces, children may select the most efficient way to complete the task following the step-by-step procedure. For example, one way is to first take each side of the lace in each hand; second, cross them over and end with the first knot; and third, use the "bunny ears" method, looping each side of the lace then knotting again.

Another example of an algorithm task is a treasure hunt. The goal for this task is for children to create or follow a set of directions to locate a treasure. This can be a group task in which children create a treasure map based on the information given to them. A teacher may provide information about where the treasure is hidden using picture cards. This will allow them to create the map and locate the treasure. The whole process helps children practice algorithm skills as they create an accurate map with important landmarks to follow.

Process Incorporating Computational Thinking in Early Childhood Classroom Activities

Nearly all young children's activities can be feasibly modified and differentiated to support CT development. We explain how to do so in three parts: (a) activity selection considerations, (b) activity modification and implementation planning, and (c) activity differentiation to meet all learners' needs. We first discuss each of these parts using "planting seeds" activities, then illustrate our entire process from selection, modification and implementation planning, and differentiation options.

Activity Selection Considerations

We recommend selecting an activity with existing or possible characteristics that specifically address at least one of the four foundational CT skills. For example, recall that puzzles and blocks are activities that have *decomposition* characteristics: components or pieces that make up the whole. The first row of Table 1 displays characteristics to consider when selecting activities that support children's development of the four foundational CT skills.

Table 2 illustrates how to use our method for incorporating CT in early childhood classroom activities, we present our process using a common classroom activity not previously discussed. Typically, in early childhood classrooms, when studying plants and the life cycle, children will plant seeds in soil, water the plants and set them in sunlight, and chart their growth over time.

Table 1 Considerations for supporting foundational computational thinking skills in early childhood classroom activities

	Foundational CT skill			
	Decomposition	Abstraction	Pattern recognition	Algorithm design
Activity characteristics	Activity has components or pieces	Activity has information that is irrelevant or not useful	Activity has similarities within or to other activities	Activity has rules or a step-by-step process
Guiding questions for activities	"What are all the things we'll need?" "What are all the things we'll have to think about?"	"What are the important details?"	"How is this similar to something else we've done?"	"What are the rules we must follow?" "What are the steps to complete this problem?"
Suggestions for activity modification or differentiation	Increase or decrease the number of components or elements in an activity Represent components or elements in a written list or in visual form	Increase or decrease the number of relevant pieces of information Increase or decrease the number of irrelevant or not useful pieces of information	Increase or decrease the length of the pattern or sequence Increase or decrease the similarities or differences between activities	Increase or decrease the number of steps or rules needed Change the possible order of steps for a successful outcome

Table 2 Planting seeds activity

Planting seeds			
In this common early childhood classroom activity, children learn about living things and the life cycle, specifically focusing on plants. Children plant seeds and observe them as they grow			
Decomposition	Abstraction	Pattern recognition	Algorithm design
Items needed: • Cups • Seeds and seed packets • Water • Soil • Sunlight	Adding irrelevant items (e.g., weeds) helps children's abstract thinking by sorting out these items from needed items	Similarities to the needs (e.g., water) of other living things like pets Connections to knowledge and previous experience with plants	Set of directions (may be from seed packet): • Partially fill cup with soil • Insert seed(s) spaced apart as directed • Cover with soil • Water • Place in sunlight (as directed)
Example guiding instructional questions			
"What do we need to plant our seeds?"	"Is there anything you see that we don't need?" or "What do we need to plant the seeds?"	"What do all living things need to grow?"	"What are the steps to plant our seed?" "What do we do after filling the cup with dirt?"
Mathematics	Literacy	Modification and differentiation Ideas	
Count out items needed Measure the growth of the plants	Create a list of items needed Write out how to plant a seed Read and discuss children's books about plants and living things	Pre-fill cups with dirt Have students determine how much of each item is needed for each child to plant a seed Predict how much plants will grow and compare to actual growth	

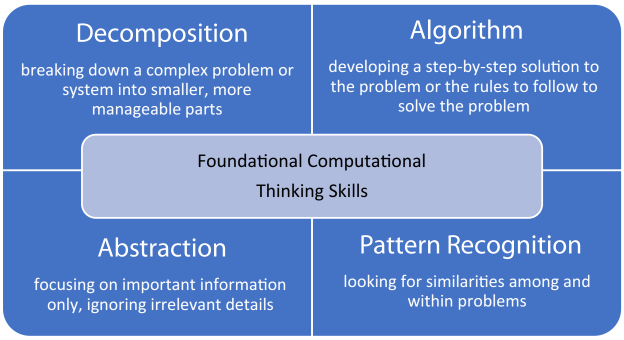

Fig. 1 Computational thinking foundational skills. *Note* Adapted from ISTE (2016)

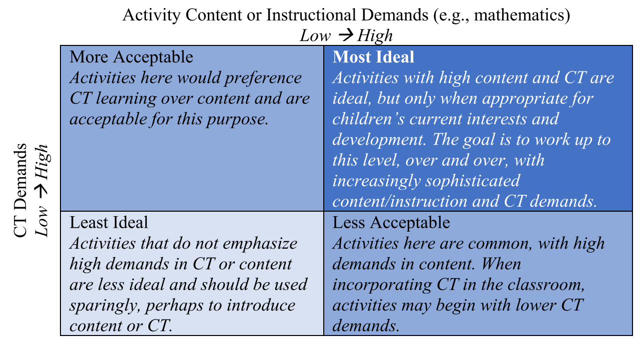

Fig. 2 Variations for implementing classroom activities. Adapted from Joswick et al. (2019)

We select this activity (See Table 2) because it has all four major CT skills-*decomposition*, *pattern recognition*, and *algorithm design.* When planting seeds, children decompose the items needed to plant seeds and to help them grow (e.g., cups, seeds, soil). As they do so, they can recognize similar pattens in identifying what living things need to grow. For example, nutrients and water are needed for all! Finally, children follow each step in a set of directions to plant their seeds. The first row of Table 2 details the foundational CT skills in a seed-planting activity for young children.

Activity Modification and Implementation Planning

Selected classroom activities can be modified to support CT skills. Consider an activity like story time. We noted previously that story time is filled with teacher-scaffolded abstraction, an intentional modification to the activity. That is, story time could simply be a teacher-read text activity. To support foundational CT skills, like *abstraction*, the teacher can ask questions along with the reading to help children focus on relevant information from the text, like "What are the important details?" Listed in the second row of Table 1 are guiding instructional questions teachers can use with their chosen activity to support each foundational CT skill.

For example, "planting seeds" activity support mathematics and literacy content in addition to CT skills. Ways to support mathematics include asking children to count out the items needed, count the number of seeds, or measure the growth of the plants. For literacy, children can generate a list of the items needed for planting. Ideas for guiding questions to ask during instruction are in the second row of Table 2; content variations are listed in the third row.

Teachers can implement activities repeatedly, with varying goals for learning each implementation (See Fig. 2). For instance, the first time an activity is carried out, the focus could be on the mathematics or literacy content. Once the content is more familiar to the child, the next time could feature an emphasis on the CT.

Consider sorting activities. To familiarize children with sorting, an early implementation of the activity may be to sort two colors of blocks into two groups by color. This has a low demand on CT skills and content or instructional knowledge and therefore falls into the "Least Ideal" category seen in Fig. 2. However, this task can be more complex as children become familiar with sorting. An implementation with high CT and content demands would be to sort a collection of six pattern block types by shape characteristics (e.g., number of sides). Sorting activities in between may feature sorting a limited number of shapes by type (e.g., squares and triangles) or sorting various materials or different sizes and types by color (varying between low and high CT demands and low and high content or instructional demands).

Another consideration is that some early childhood classroom activities can support more than one foundational CT skill. Cycles of implementation may focus more or less on each of the integrated foundational CT skills. Figure 2 displays a continuum matrix of demands considering the content of activities (e.g., mathematics) and the incorporation of CT.

Activity Differentiation

We recommend choosing to vary the demands of content or instruction and CT of an activity so that it meets the progression of the course (e.g., starting with less ideal activities and working up to most ideal, incorporating iterations of activities that are less and more acceptable along the way) and the needs of students (e.g., those with advanced content knowledge would benefit from incorporating progressively increased CT demands). That is, the modification and implementation principles we have previously discussed can be used to differentiate activities to meet the varying needs of young children.

In the third row of Table 1 above, we present general guidelines for varying each foundational CT skill in an activity. For example, to differentiate the *algorithm design* demands, the number of steps or rules needed can be increased or decreased. Think of the "Treasure Hunt" activity: the number of directions between the start and the treasure can be as few or as many as appropriate for each child. Further, the difficulty of the steps can be varied. A direction like "take three steps from the bookshelf" is less demanding than a direction that requires a measurement (e.g., "move three feet to the left of the bookshelf").

Ideas for modifications and differentiation of the seed-planting activity may include pre-filling the cups with dirt, predicting the growth of the plants, comparing the growth of different types of plants given different environments (e.g., more or less water or more or less light), and so forth. Table 2 summarizes the activity and offers further ideas for differentiation.

Conclusion

Though computational thinking is a comparatively new and emerging concept in early childhood education, researchers have found support for the need to build children's foundational CT skills during early childhood education (e.g., Bers, 2018a, 2018b; Lavigne et al., 2020; Strawhacker & Bers, 2019). CT is composed of four main and interconnected skills: decomposition, abstraction, pattern recognition, and algorithm design (ISTE, 2016). In this conceptual paper, we presented examples of early childhood classroom activities to support children's CT development based on these four CT skills. A unique contribution of our work is to provide early childhood educators with practical guidelines for identifying, preparing, and implementing classroom activities to support children's CT development.

When identifying classroom activities to promote children's CT, we suggest identifying classroom activities that involve at least one or two foundational CT skills. It may be necessary to vary the level of demand of the CT (and varying the content demands of the activity) based on the needs of the children in the classroom. Further, varying the demands and repeating an activity with more or less emphasis on the content (e.g., mathematics or literacy) or the CT allows children to become familiar with the particular concepts on either content or CT. Teachers may gradually build children's mastery of both content and CT by modifying the level of demand of the content and the CT skills. When implementing this approach, it is important to monitor the process of children's learning and vary the levels of demands based on children's developmental level or mastery of concepts or contents from less or least acceptable activities to more or most ideal activities (see Fig. 2). Varying the level of demand scaffolds children's understanding of the content and/or the CT at higher levels.

Considering the importance of a play-like classroom learning setting and early exposure to developmentally appropriate yet challenging learning, we contend that early CT can be developed in young children through integrating CT skills with existing early childhood curricula like mathematics or literacy by modifying the levels of content and/or CT demands. Early exposure to CT prepares young children to become computational thinkers for computing, information processing, and coding. This enables them to become efficient and creative problem solvers who can function competently in the twenty-first century.

References

Barron, B., Cayton-Hodges, G., Bofferding, L., Copple, C., Darling-Hammond, L., & Levine, M. (2011). *Take a giant step: A blueprint for teaching children in a digital age.* The Joan Ganz Cooney Center at Sesame Workshop. https://edpolicy.stanford.edu/sites/default/files/publications/take-giant-step-blueprint-teaching-young-children-digital-age.pdf

Bers, M. U. (2018a). Coding and computational thinking in early childhood: The impact of Scratch Jr. in Europe. *European Journal of STEM Education, 3*(3), 8. https://doi.org/10.20897/ejsteme/3868

Bers, M. U. (2018b). *Coding, playgrounds and literacy in early childhood education: The development of KIBO robotics and Scratch Jr*. https://sites.tufts.edu/devtech/files/2018/05/EDUCON.pdf

Bransford, J. D., Brown, A. L., & Cocking, R. R. (2000). *How people learn: Brain, mind, experience, and school: Expanded edition* (1st ed.). National Academy Press.

Brett, J. (1989). *The mitten: A Ukrainian folktale, adapted*. Putnam.

Cansu, S. K., & Cansu, F. K. (2019). An overview of computational thinking. *International Journal of Computer Science Education in Schools, 3*(10), 17–30. https://doi.org/10.21585/ijcses.v3i1.53

Carle, E. (1969). *The very hungry caterpillar*. World Publishing Company.

Chongo, S., Osman, K., & Nayan, N. A. (2020). Level of computational thinking skills among secondary science students. *Science Education International, 31*(2), 159–163. https://doi.org/10.33828/sei.v31.i2

Computer Science Teachers Association. (2017). *CSTA K-12 computer science standards* (Revised 2017). https://k12cs.org/wp-content/uploads/2016/09/K%E2%80%9312-Computer-Science-Framework.pdf

Dietz, G., Landay, J., & Gweon, H. (2019). Building blocks of computational thinking: Young children's developing capacities for problem decomposition. *Cognitive Science,* 1647–1653.

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

International Society for Technology in Education. (2016). *ISTE National Educational Technology Standards (NETS).* https://www.iste.org/iste-standards

International Society for Technology in Education. (2017). Computational thinking. competencies. http://www.iste.org/standards/computational-thinking

International Society for Technology in Education. (2021). *Computational thinking competencies.* https://www.iste.org/standards/iste-standards-for-computational-thinking

Joswick, C., Clements, D. H., Sarama, J., Day-Hess, C., & Banse, H. (2019). Double impact: Mathematics and executive function.

Teaching Children Mathematics, 25(7), 416–426. https://doi.org/10.5951/teacchilmath.25.7.0416
Lavigne, H., Presser, A. L., Rosenfeld, D., Wolsky, M., & Andrews. J. (2020). Creating a preschool computational thinking learning blueprint to guide the development of learning resources for young children. *Connected Science Learning, 2* (2). https://www.nsta.org/connected-science-learning/connected-science-learning-april-june-2020/creating-preschool
Lavigne, H. & Wolsky, M. (2021). Using stories to support computational thinking. Edutopia. https://www.edutopia.org/article/using-stories-support-computational-thinking
Lee, J. (2020). Coding in early childhood education. *Contemporary Issues in Early Childhood, 21*(3), 266–269. https://doi.org/10.1177/1463949119846541
Lee, J., Collins, D., & Winkelman, L. (2015). Connecting 2D and 3d: Drafting blueprints, building, and playing, *Young Children, 70* (1), 32–35. https://www.proquest.com/docview/1657332761?pq-origsite=gscholar&fromopenview=true
Lee, I., Grover, S., Martin, F., Pillai, S., & Malyn-Smith, J. (2019). Computational thinking from a disciplinary perspective: Integrating computational thinking in K-12 science, technology, engineering, and mathematics education. *Journal of Science Education and Technology, 29*, 1–8. https://doi.org/10.1007/s10956-019-09803-w.pdf
Lee, J., & Jo, J. (2019). Implementing unplugged coding activities in an early childhood classroom setting. *Early Childhood Education Journal, 47*, 709–716. https://doi.org/10.1007/s10643-019-00967-z
Lee, J., Joswick, C., Pole, K., & Jocius, R. (2021). Algorithm design for young children. *Contemporary Issues in Early Childhood* https://doi.org/10.1177/14639491211033663
Lee, J., Lee, J. O., & Collins, D. (2009). Enhancing children's spatial sense: Tangrams. *Childhood Education, 86*(2), 92–94. https://doi.org/10.1080/00094056.2010.10523120
Lu, J., & Fletcher, G. (2009). Thinking about computational thinking. *SIGCSE '09*. https://doi.org/10.1145/1539024.1508959
Lynch, M. (2019). *Why we must teach our teachers computational thinking?* https://www.thetechedvocate.org/why-we-must-teach-our-teachers-computational-thinking/
Martin, F. (2018). Rethinking computational thinking. *CSTA—The Advocate,* (Feb. 17, 2018). http://advocate.csteachers.org/2018/02/17/rethinking-computational-thinking/
Merino-Armero, J., González-Calero, J., & Cózar-Gutiérrez, R. (2021). Computational thinking in K-12 education. An insight through meta-analysis. *Journal of Research on Technology Education*. https://doi.org/10.1080/15391523.2020.1870250
Modan, N. (2019). *33 states adopted 57 computer science ed policies since 2018*. https://www.k12dive.com/news/33-states-adopted-57-computer-science-ed-policies-since-2018/562530/
National Association for the Education of Young Children. (2009). *Key messages of the position statement*. https://www.naeyc.org/sites/default/files/globally-shared/downloads/PDFs/resources/position-statements/KeyMessages.pdf
National Association for the Education of Young Children. (2012). *Position statement of the National Association for the Education of Young Children and the Fred Rogers Center for Early Learning and Children's Media at Saint Vincent College*. https://www.naeyc.org/sites/default/files/globally-shared/downloads/PDFs/resources/topics/12_KeyMessages_Technology.pdf
National Association for the Education of Young Children. (2020). *Developmentally appropriate practice*. https://www.naeyc.org/sites/default/files/globally-shared/downloads/PDFs/resources/position-statements/dap-statement_0.pdf
Piaget, J. (1957). *Construction of reality in the child*. Routledge & Kegan Paul.
Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas.* Basic Books. https://www.google.com/books/edition/Mindstorms/nDjRDwAAQBAJ?hl=en&gbpv=1&dq=Papert,+S.+(1980).++Mindstorms:+Children,+Computers,+and+Powerful+Ideas.+Basic+Books.&pg=PT12&printsec=frontcover
Partnership for 21st Century Skills. (2009). *P21 Definition framework*. https://files.eric.ed.gov/fulltext/ED519462.pdf
Ramey, C. T., & Ramey, S. L. (1999). *Chapter 8. Beginning school for young children at risk*. Retrieved from https://files.eric.ed.gov/fulltext/ED438026.pdf#page=231
Rittle-Johnson, B., Fyfe, E. R., Hofer, K. G., & Farran, D. C. (2016). Early math trajectories: Low-income children's mathematics knowledge from ages 4 to 11. *Child Development, 88*, 1727–1742. https://doi.org/10.1111/cdev.12662
Rubin, A., & Salmieri, D. (2012). *Dragons love tacos*. Penguin Random-House.
Sipe, L. R. (1998). How picture books work: A semiotically framed theory of text–picture relationships. *Children's Literature in Education, 29*, 97–108. https://link.springer.com/article/10.1023%2FA%3A1022459009182
Stephens, K. (2007). *Curiosity and wonder: Cue into children's inborn motivation to learn*. Retrived from https://www.easternflorida.edu/community-resources/child-development-centers/parent-resourcelibrary/documents/curiosity-and-wonder.pdf
Strawhacker, A., & Bers, M. U. (2019). What they learn when they learn coding: Investigating cognitive domains and computer programming knowledge of young children. *Educational Technology Research and Development, 61*(3), 541–575. https://link.springer.com/article/10.1007%2Fs11423-018-9622-x
Sykora, C. (2021). *Computational thinking for all*. Retrived from https://www.iste.org/explore/computationalthinking/computational-thinking-all
Taback, S. (2004). *This is the house that Jack built*. Puffin.
U. S. Center for Disease Control and Prevention. (2021). *Cleaning, disinfection, and hand hygiene in schools—A toolkit for school administrators*. https://www.cdc.gov/coronavirus/2019-ncov/community/schools-childcare/clean-disinfect-hygiene.html
Valenzuela, J. (2020). *How to develop computational thinkers.* https://www.iste.org/explore/how-develop-computational-thinkers
Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies, 20*(4), 715–728. https://doi.org/10.1007/s10639-01509412-6
Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology, 25*(1), 127–147. https://doi.org/10.1007/510956-051-9581-5
Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33–35. https://www.cs.cmu.edu/afs/cs/Web/People/15110-s13/Wing06-ct.pdf
Wing, J. M. (2008). Computational thinking and thinking about computing: Philosophical transactions of the Royal Society. *Physical and Engineering Sciences, 366*(18813), 717–725. https://doi.org/10.1098/rsta.2008.0118
Wolfenbarger, C. D., & Sipe, L. R. (2007). A unique visual and literary art form: Recent research on picturebooks. *Language Arts, 84*(3), 273–280. https://www.csun.edu/~bashforth/305_PDF/305_FinalProj/RecentRessaerchPicBooks_Jan07_LA.pdf
Yadav, A., Zhou, N., Mayfield, C., Hambrusch, S., & Korb, J. T. (2011). Introducing computational thinking in education courses. In *Proceedings of the 42nd ACM technical symposium on computer science education* (pp. 465–470). ACM. http://dl.acm.org/citation.cfm?id=1953297

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Early Childhood Education Journal is a copyright of Springer, 2023. All Rights Reserved.