CSIRO

Australia’s National Science Agency

Computational Thinking Mini Activities

Goal

Use our fun, hands-on activity ideas to encourage students to practice their computation thinking skills.

Time

Activities can be 5-10 minutes or longer.

Age-group

Range from Year P-10. All activities can be adjusted to suit your class.

STEM Professional Partnership

Include examples of how these skills are utilised in your workplace.

Pattern Recognition

Observing patterns, trends and regularities to make sense of data.

Time Patterns – Prediction

Give students a repeating timetable (like a bus timetable or movie timetable) with some times blanked out. Ask students to guess the missing times using pattern recognition.

Movement Patterns

Students create whole body patterns and build on each other's patterns. For example, Student 1: star jump, normal jump; Student 2: star jump, normal jump, wiggle fingers, etc.

Creating Pattens

Give students a collection of coloured shapes, blocks, beads, counters etc. Challenge students to create as many patterns as possible.

Grouping

Give students a random selection of objects and ask them to group them through a shared similarity. Encourage students to rearrange objects into a variety of groupings.

Decomposition

Breaking down data, processes or problems into smaller, manageable parts

Recipe

Students create a simple recipe (e.g. a peanut butter sandwich). Place students into small groups and allow them to brainstorm all the important steps/tasks involved. Have them order each step, then allow them time to complete their task (if relevant) before coming together.

Building

Explain to students that they will be building a tower out of blocks. Place students in groups and have them break the project down into important tasks. Have students assign themselves to tasks and work on these independently before coming back together as a group to begin the build process. Afterwards discuss with students how the build went.

Math Problem

Give students age-appropriate math problems that involve more than one step. These can be straight numerical problems or word problems.

Reverse Engineer Paper Planes

Have students create paper planes and then switch with a classmate. The students' challenge is to work out and record the steps required to make the plane they have been given.

Modelling and Simulation

Developing a model to imitate processes and problems

Simple Circuits

Have students work in small groups to create simple circuits designed to turn on a LED light. This can be done physically using materials or online at CircuitLab. If you would like to do this activity with materials, planning is required.

Spaghetti Bridges

Students have 'x' amount of time to create spaghetti and marshmallow bridges. Structures must be sturdy against resistance (like a weight or fan breeze). Students plan their design before building, and depending on time, they can have multiple attempts, discuss others' work, and address problems and successes.

Programmer Simon Says

In groups, create programmer commands like '1x jump means jump in the air once'. Display the codes at the front of the room and play the game following Simon Says rules. Increase difficulty by giving multiple instructions simultaneously or setting a goal, like getting all students to one side of the room.

Ecosystem Models and Food Chains

Take students outside. Pick an area such as a garden bed and ask students to document the ecosystem within. Students can recreate the ecosystems they saw and create the food chains within this ecosystem. This can be done with hands on materials, in books or on computers.

Abstraction

Identifying and extracting relevant information. The process of ignoring or removing unnecessary information

Improve Instructions

Divide students into pairs. One student creates inefficient instructions with irrelevant information to guide their partner to a specific spot. After the partner follows the instructions, they make it more efficient. Repeat the activity twice, allowing each student to take both roles.

Mind Maps

As a class, choose a broad topic and create a mind map. Focus and narrow down the information on the map, highlighting relevant details. Use this information to create a new mind map and repeat the process. For example, ecosystems -> deserts -> weather -> plants -> adaptations.

Word Problems

Give students word problems. They identify critical information needed to solve each problem before answering the questions. Alternatively, they can create their own word problems and exchange them with classmates.

Venn Diagrams

Provide students with two topics and have them brainstorm different things related to that topic. Ask students to identify the characteristics that each topic/item has in common and create Venn diagrams.

Algorithms

Creating an ordered series of instructions for solving problems or for doing a task.

People Robots

Pair students up. One student writes down the steps required to accomplish a task (e.g., wash and refill a water bottle, put rubbish into a bin), and the other student follows the steps exactly. Look for gaps or missing details in each other's algorithms. Model an example to the whole class before starting the activity. Emphasise following the instructions as written without relying on prior knowledge.

Rock Paper Scissors – Predictions

Group students (minimum size of 3). Play 10 rounds of rock-paper-scissors, with two playing while the other students record their choices and then makes predictions of who would win in a best of 20 situation. Let them compete to test predictions. Collect whole class data to assess overall prediction success.

Everyday Algorithms

Brainstorm with students types of algorithms they follow every day. Have students write out the steps of their algorithm. Pair up students and have them role-play their algorithm. Students guess each other’s algorithm purpose. E.g. brush teeth, school bell routine.

Alien Instructions

Provide students with incomplete set of steps for a normal everyday activity. Explain to students that these steps are for an Alien who knows nothing about Earth. Ask students to look through the instructions and make any changes needed.

Evaluation

Determining the effectiveness of a solution and generalising. Applying that information to new problems

Simple codes

Provide students with snippets of simple code. Have students evaluate the code and look for opportunities to adjust and improve the code.

Back-to-back art

Pair students, give each a picture (complexity by age), and have them sit back-to-back. Instruct each other on drawing their image. Compare the images and discuss improvements. Repeat with new images to evaluate accuracy and determine reasons for improvement.

True or false

Before the lesson, prepare a series of true/false statements. Instruct students to decide if each statement is true or false. Have students stand up and step left for false and right for true. Alternatively, they can write responses on post-its and place them on a board where the question is displayed. Consider including computer science-related questions (e.g., a set of commands making a robot return to its starting position after taking one step at a time).

Data Manipulation

Display a misleading graph and discuss the reasons. Make improvements as a group. Provide misleading graphs for students to evaluate, identify misunderstandings, and redesign. They can work in pairs, groups, or individually. Students should show before and after changes made and explain why.