LEARNING SCENARIO Finding solutions with computational thinking

Educational level: Early childhood, primary, secondary | **Age:** >6

Author: Michael Steiner and Hermann Morgenbesser, Future Learning Lab Vienna

LEARNING OBJECTIVES/ ASPIRATIONS

The Computational Thinking (CT) aims to use analytic and algorithmic approaches to formulate, analyse and solve problems. The CT practices include design and development of computational artefacts, models, simulations; artefacts of natural and artificial phenomena collaboratively and the implementation of computing techniques to solve problems, such as coding, programming and robotics.

NARRATIVE OVERVIEW

Thinking can only be best developed if new problems are constantly being solved. Computation thinking as aimed at developing the following competencies:

- Familiarity in dealing with complexity;
- Resistance in dealing with difficult problems;
- Tolerance for ambiguity;
- Ability to deal with open questions;
- Ability to communicate and to solve problems together.

These are educational values that are becoming more and more important.

To develop CT, students can be faced with the problems that can be open, ambiguous, complex and so difficult that they can only be solved in a team and through good communication. It must also be emphasized that CT can be learned and taught at all ages from early childhood to the school-leaving examinations and of course beyond; in a variety of disciplines: from science and math to writing and literature.

An example is around building a hand-made pedometer. The teacher assigns to think of a problem a student wants to solve. Importantly, students decide based on their own interests. The student chooses something that puzzles him and brings the story to the teacher: The student's sister claims she walks 10,000 steps every day. She works as a kindergarten teacher and walks to and from work every day. She thinks it keeps her fit. The student wants to understand what it means to walk 10,000 steps. Counting all her steps in one day is not possible. Besides, she probably does not walk all the time, but sits at her work sometimes. This is discussed with the teacher and other students. Together, they decide to build a pedometer that would count the steps the student's sister walks. The task is formulated as to build a pedometer that attaches to your wrist or ankle and counts steps as you walk. With each step, a pulse is counted and then shown on the display. Also, it needs to include the possibility to restart the pedometer (reset).

APPROACH TO TEACHING AND LEARNING

- Project-oriented teaching and teamwork are adequate methods to deal with more extensive problems of computational thinking.
- The Computer Science (CS) Unplugged involves problem solving to achieve a goal, dealing with fundamental concepts from CS.
- Computer simulations to explore phenomena.
- Computer models that can be tested, debugged and refined.
- A computer game or app construction project.

ASSESSMENT: 'a system of assessments'

E.g. Computational Thinking Pattern Analysis framework enables to visualize which of nine specific skills students have mastered in game design. Dr. Coding targets seven dimensions of CT competency.

Analysing the artefacts, troubleshooting scenarios

ROLES

TEACHERS: Teachers must learn how to manage a classroom in which the computer serves as both the primary medium for demonstrating performance as well as an occasional teaching aid. The activities can be arranged in the different learning zones of the classroom. Also, to support collaboration in groups, scaffolding is necessary.

LEARNERS: Students work collaboratively (as a team in planning and developing solutions), and take on roles (programmer, analyst, builder etc.) as the group seeks to regulate its work.

OTHERS: External experts can be invited via Conference tools to support activities.

LEARNING ENVIRONMENT

Interaction between the teacher and the students is important throughout the educational experience while the teacher gives instructions, feedback and guides through the process. The students work in teams, and actively exchange the views on the project tasks, roles division. The students develop their draft solution with the support of the teacher. Teams create solutions. They can share different tasks to create a product. For example, one team programs, the other works with the micro:bit, and other materials.

Next, the students share their progress with the teacher, and the teacher further coaches on the next steps and gives possible hints, also discusses possible difficulties or mistakes – again interaction happens. Finally, the students present their work or progress and reflect on how their work has developed, also on the team work.

POSSIBLE CHALLENGES

When integrated into compulsory education, there is an open question of what kind of assessment could elicit students' problem solving and CT skills in authentic contexts.

There is also the need to develop certain behaviour among students – for working with others and dealing with frustration.

Age-appropriate activities need to be chosen, but also there is a need to respond to students' interests and not to demotivate girls activities need to be proposed to their taste as well.

Importantly, CT learning activities require teaching capacity in designing and assessing CT classroom experiences that focus on CT concepts.

RESOURCES

- Programming language, e,g, Python, SNAP
- micro:bit
- old fabrics needle and thread
- velcro tape

LEARNING ACTIVITIES

CT learning activities are built around the key CT concepts such as abstraction, algorithmic thinking, automation, decomposition, debugging, and generalization. Coding and programming is a constituent of CT as it makes CT concepts concrete and can become a tool for learning. However, even more important is what precedes coding and programming – the process of problem analysis and problem decomposition.

Key activities for the Pedometer Story would be:

- The teacher gives instructions around designing pseudo code/UML prototypes, or developing a solution.
- Development of draft solutions in teams.
- General presentation of the running programs or working pieces.
- Reflection on the process and making the necessary changes.
- There is also space for improvement. The last stage could be around re-making or improving the solution.

LITERATURE TO SUPPORT

- Wing, J. (2006): Computational thinking. Communications of the ACM, 49(3), 33-35.
- Computational Thinking Task Force: https://csta.acm.org/Curriculum/sub/CompThinking.html
- Computer Science Unplugged: https://csunplugged.org/en/
- K-12 Framework: https://k12cs.org/wp-content/uploads/2016/09/Computer-Science-Framework.pdf
- Coding with microbits: https://padlet.com/eis/dlplwien

LEARNING SCENARIO VIDEO

https://www.youtube.com/watch?v=Z7xg1yZGeW0

Co-funded by the Erasmus+ Programme of the European Union

The learning scenario is created by the DesignFILS project (http://designfils.eba.gov.tr), funded by EU's Erasmus+ KA2 (grant agreement 2019-1-TR01-KA201-076567). The contents of the publication are the sole responsibility of the authors, and the EC or Turkish National Agency cannot be held responsible for any use which may be made of the information contained therein.