IO 3+4

# Didactic Content

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

<colette/>

# Enhancing Computational Thinking

## ...through linked tasks in <colette/>.

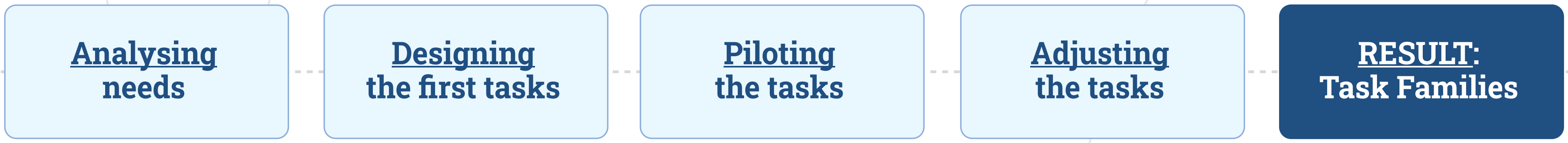
So-called **Generic Tasks** were designed to be easy to use for teachers in mathematics or other subjects in **lower secondary** and **upper secondary** education. They are clustered into the **Task Families.**

Secondary teachers can use ready-made example task sequences, or design and edit their own tasks, in the <colette/> portal (IO2). The tasks are arranged in a (didactical) series, a so called **Path**.

Students can solve their teacher's tasks in the path by using the mobile app (IO1).

Instructions how to produce the path in the portal are available in three different formats of handbook for teachers (IO5) and the possible pedagogies and implementation of the tasks in the schools using the feature of Digital Classroom are described in the short-term curriculum (IO6).

# Designed with Teachers for Teachers



**Analysing needs**

- Analysing national curricula
- Interviewing teachers [focus on mathematics & computer science]

**Designing the first tasks**

- Computational Thinking skill & Task Families

**Piloting the tasks**

- Classroom observations
- Interviewing students
- Teachers providing feedback

**Adjusting the tasks**

- Adjusting the portal
- New scenarios

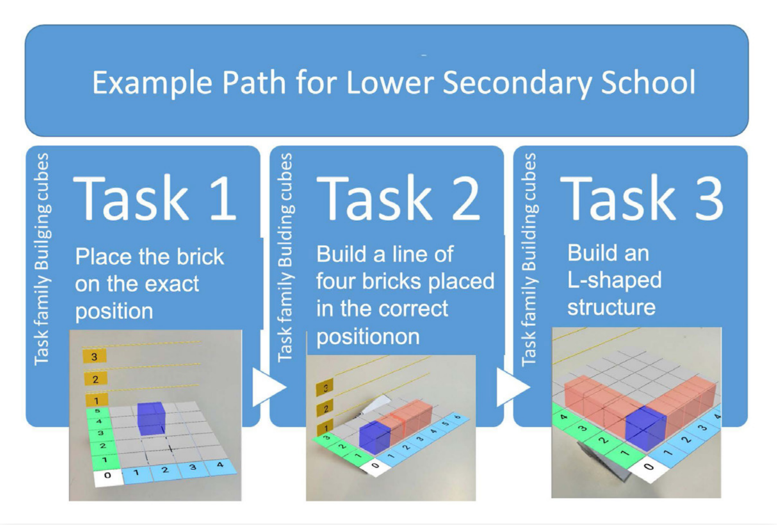
**RESULT: Task Families**

**Spread among teachers at teacher conference and summer schools**

# A <colette/> – path

A Path is a series of tasks that are worked on one after the other by the students in the mobile app. In the webportal, the teachers are free to decide which tasks to add and whether to mix them up or choose them to match each other thematically, to tailor it to the needs of their classroom (e.g. subject, age, level of Computational Thinking, prior knowledge of the students).

A Path consists of tasks from one or from multiple Task Families with gradually increasing difficulty. The pictures show example paths for both lower- and upper-secondary level for the Building Cubes Task Family where the students use a block-based programming language to work with virtual cubes.



Example Path for Lower Secondary School

Task family Builging cubes

Task 1

Place the brick on the exact position

Task family Building cubes

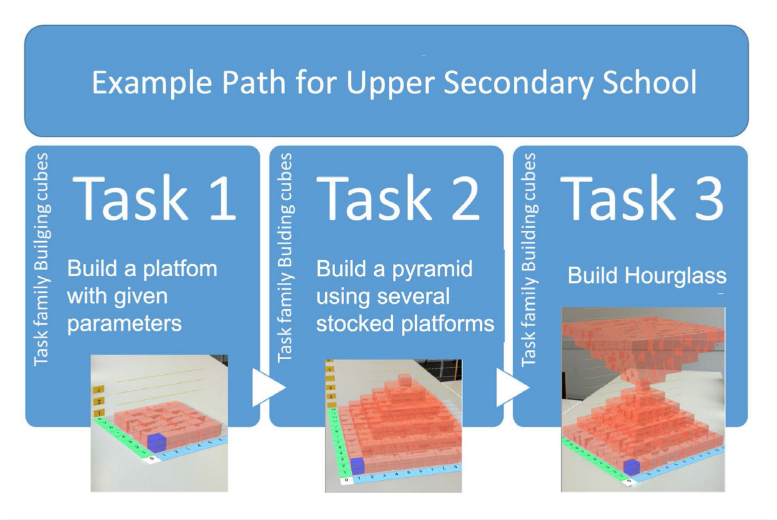
Task 2

Build a line of four bricks placed in the correct positionon

Task family Building cubes

Task 3

Build an L-shaped structure



Example Path for Upper Secondary School

Task family Builging cubes

Task 1

Build a platfom with given parameters

Task family Building cubes

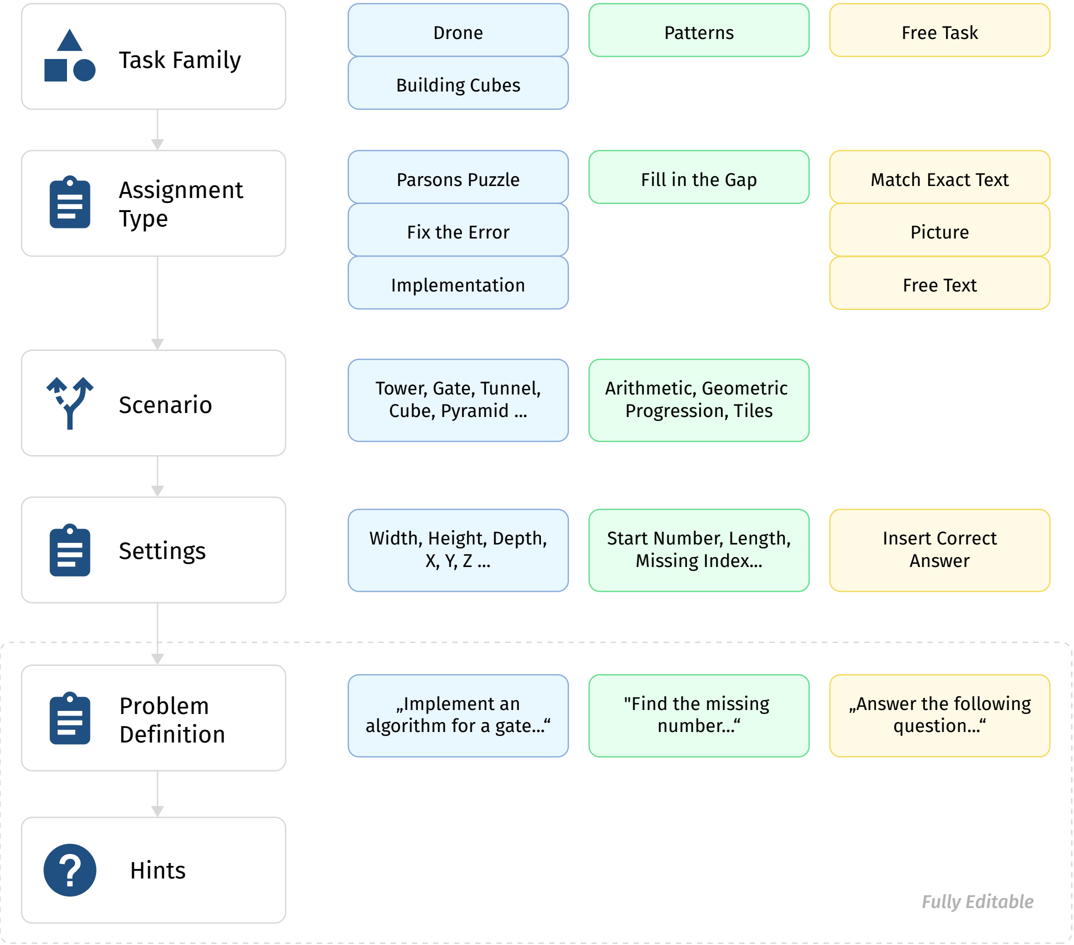
Task 2

Build a pyramid using several stocked platforms

Task family Building cubes

Task 3

Build Hourglass



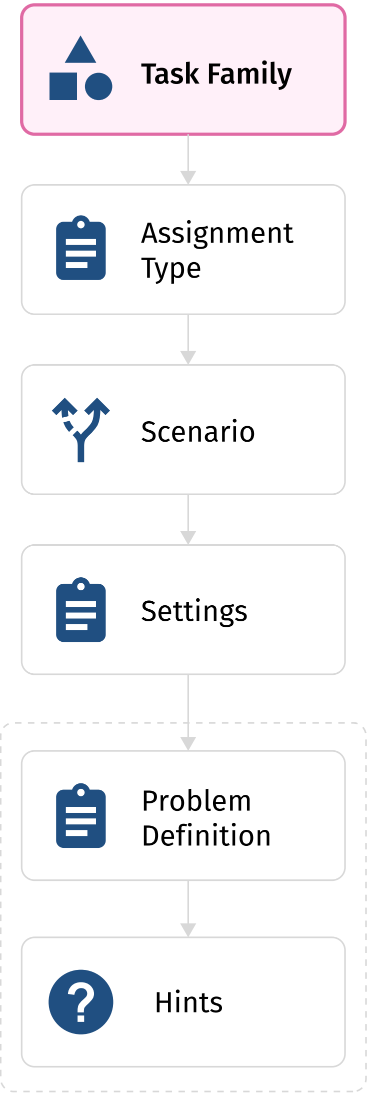
| | | | |
|---|---|---|---|
| Task Family | Drone<br>Building Cubes | Patterns | Free Task |
| Assignment Type | Parsons Puzzle<br>Fix the Error<br>Implementation | Fill in the Gap | Match Exact Text<br>Picture<br>Free Text |
| Scenario | Tower, Gate, Tunnel, Cube, Pyramid ... | Arithmetic, Geometric Progression, Tiles | |
| Settings | Width, Height, Depth, X, Y, Z ... | Start Number, Length, Missing Index... | Insert Correct Answer |
| Problem Definition | „Implement an algorithm for a gate...“ | "Find the missing number...“ | „Answer the following question...“ |
| Hints | | | |

*Fully Editable*

# Creating a task in the web-portal

**During creating tasks, teachers choose the suitable Task Family. Afterwards, the sssignment type according to the Task Family, and a scenario is chosen. For example, in the Task Family Building Cubes, the students have to fix the error in a given blockly code in the tower scenario.**

**Afterward, the teacher can finalize the settings (height of the tower), and individually write a problem definition and create 1-3 hints.**



Task Family

Assignment Type

Scenario

Settings

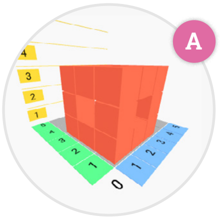
Problem Definition

Hints

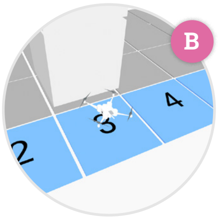
# Task Families

**In the portal, teachers can choose which Task Families they want to use in their path. A Task Family is a category of tasks that have a common core but differ in values for settings. It acts as a blueprint to create different tasks easily. Tasks from the Task Families are adjustable to be used in lower- and upper-secondary schools.**

## Overview of all Task Families

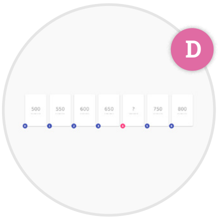


In **Building Cubes** students work with algorithms that create structures out of colored cubes.

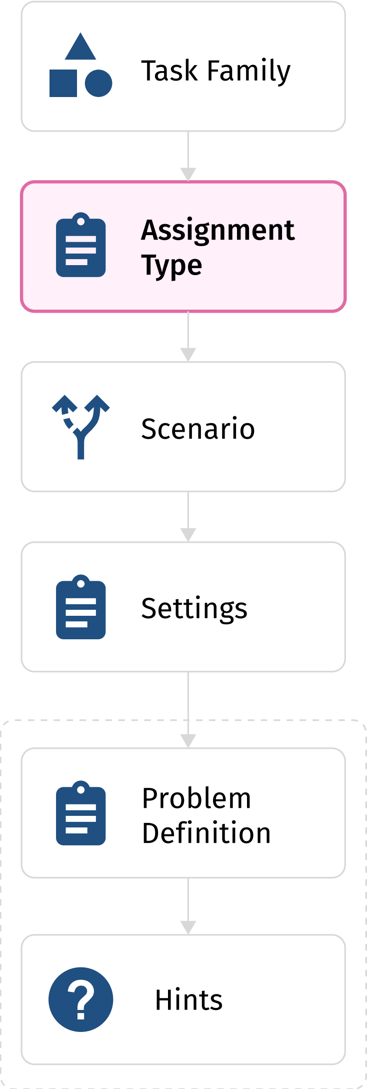


In **Drone** students work with algorithms that create a flying route of a drone that takes pictures on the way of designated spots.

In **Free Task** teachers can create their own tasks without constraints. This allows to create tasks for any kind of subject.



In **Patterns** students analyze the structure of a sequence to identify the underlying pattern and fill in the missing piece.



Task Family

Assignment Type

Scenario

Settings

Problem Definition

Hints

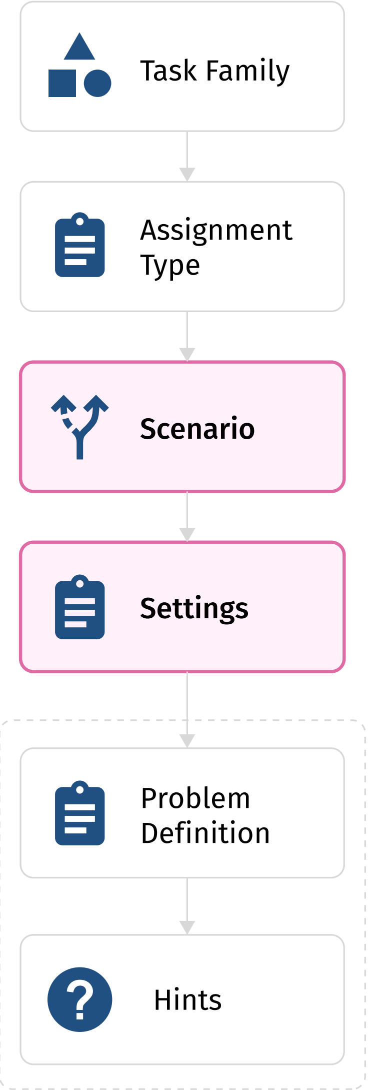
# Generic Tasks [Assignment Type]

**In the portal, teachers further choose for the coding-based tasks the Generic Tasks which are assignment types. This option provides several assignment types for exercises, addressing different Computational Thinking skills.**

## The Generic Tasks available are:

- Implementation: students have to code the desired structure
- Fix the Error: students receive an erroneous code and have to correct it
- Parson's Puzzle: students have to re-arrange randomly placed blocks to create a valid code

# Scenario & Task Settings



Task Family

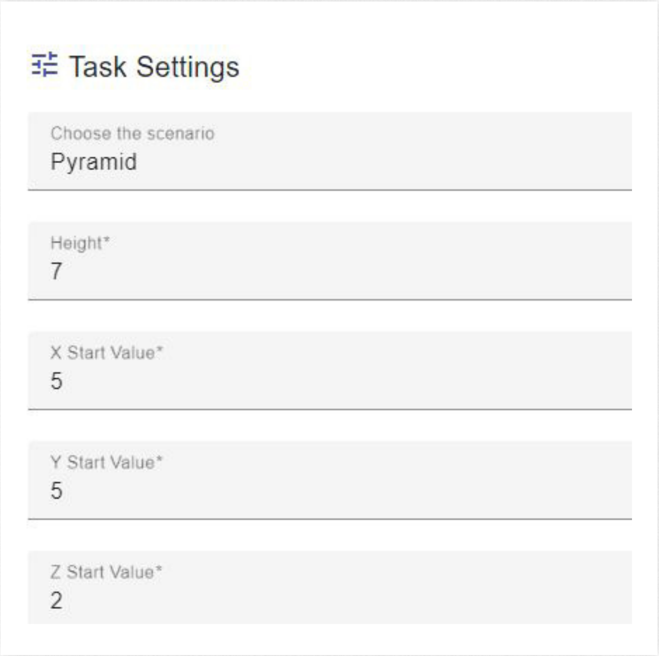
Assignment Type

Scenario

Settings

Problem Definition

Hints



Task Settings

Choose the scenario
Pyramid

Height*
7

X Start Value*
5

Y Start Value*
5

Z Start Value*
2

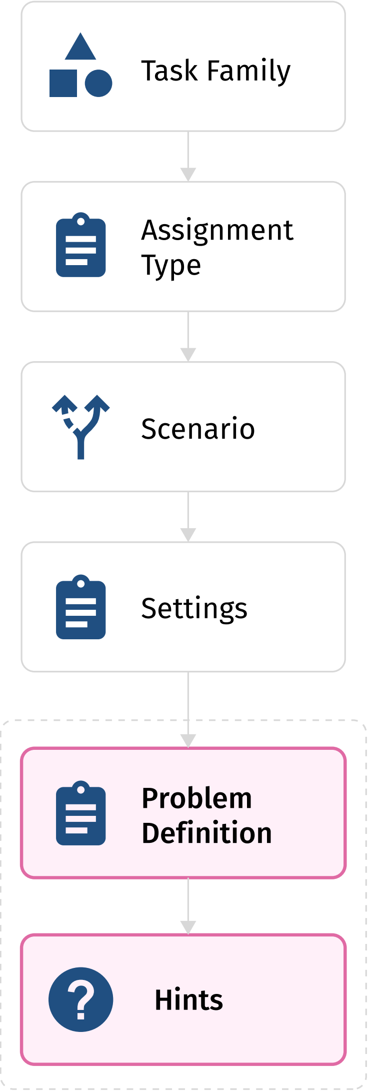
**Lastly, in the portal, teachers choose the so-called scenario of the task and the related settings, to define what the students should do in the app.**

**For example, in the Task Family Building Cubes, there are scenarios such as tower, cube, chair, pyramid, etc. In each case, the settings give the opportunity to differentiate in level of difficulty by creating different sizes of the structure.**

# Problem Definition & Hints

**Teachers can add the Problem Definition to each task. This way teachers can address the task directly to their students.**

**Further, hints can be provided by the teacher, to help the students to solve the tasks.**


Task Family

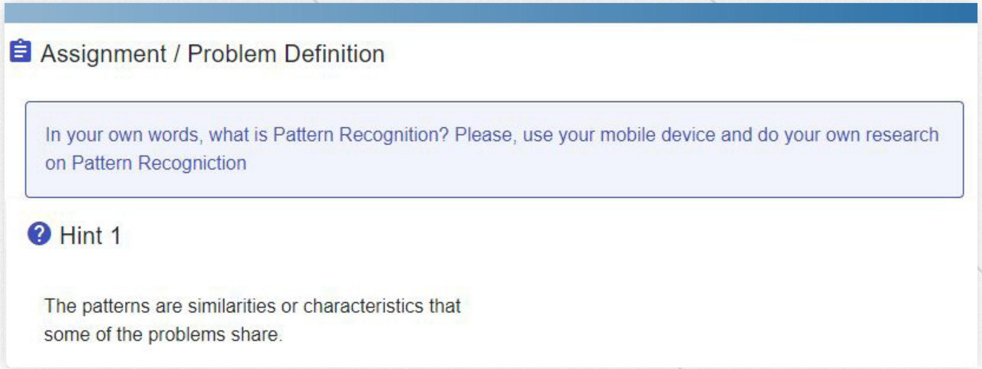
Assignment Type

Scenario

Settings

**Problem Definition**

**Hints**


Assignment / Problem Definition

In your own words, what is Pattern Recognition? Please, use your mobile device and do your own research on Pattern Recogniction

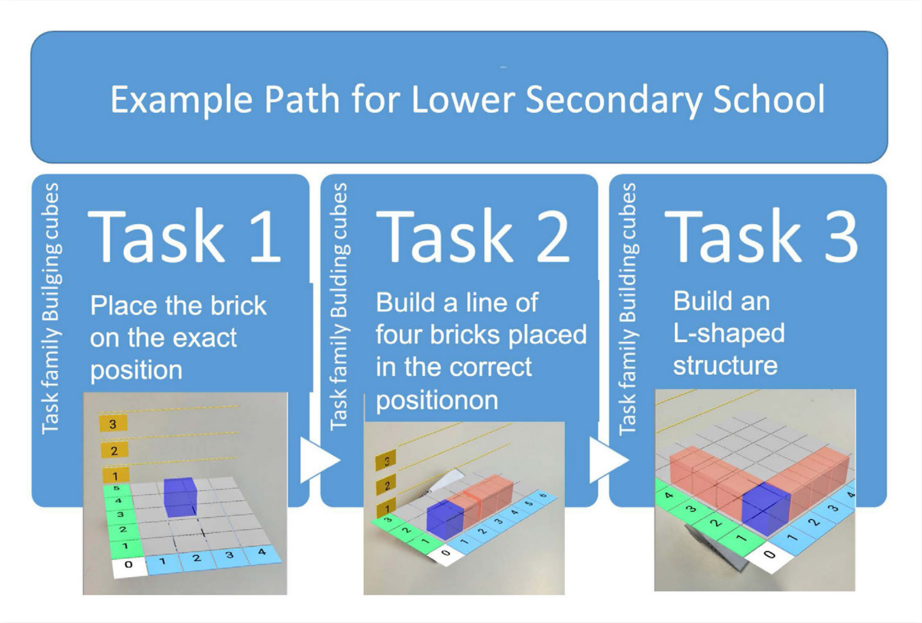
Hint 1

The patterns are similarities or characteristics that some of the problems share.

# Didactics behind a task sequence in a path

**The didactics of a task sequence in a path involve starting with a simple task and gradually adding complexity. This method introduces new knowledge one step at a time. This approach ensures smooth learning progression and effective knowledge integration.**

**In the picture you can see an example path, with three tasks, of gradually increasing difficulty.**



Example Path for Lower Secondary School

Task family Builging cubes

Task 1

Place the brick on the exact position

Task family Bulding cubes

Task 2

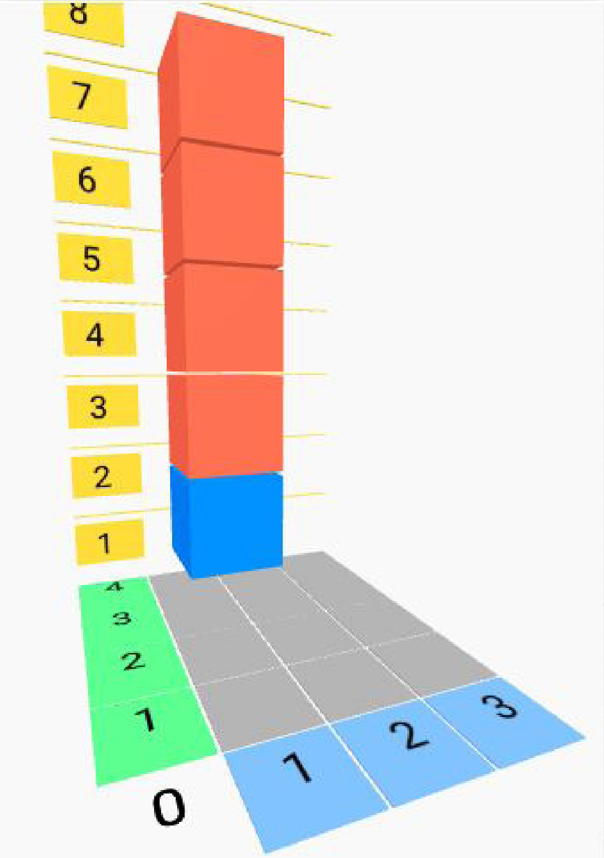
Build a line of four bricks placed in the correct positionon

Task family Building cubes

Task 3

Build an L-shaped structure

# Building Cubes [Lower Secondary Education]



**Focus on Computational Thinking skill:**
Algorithmic Thinking, Generalization, Automation and Debugging

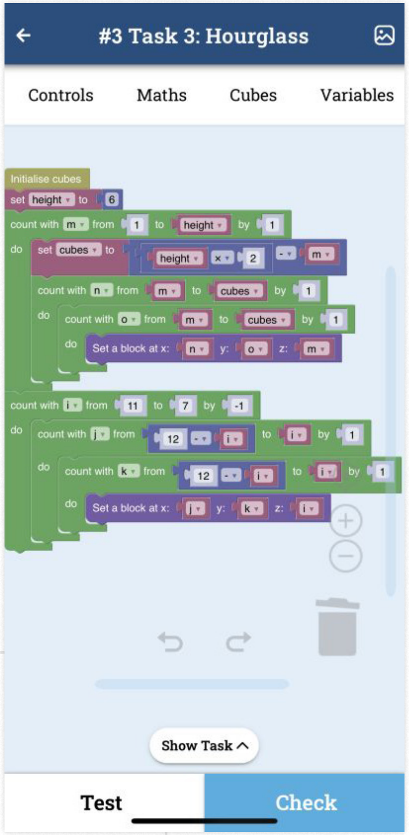
- Introductory shapes include tower, L-shape, gate, Computational Thinking logo
- Shapes to introduce nested loops are cube, cuboid, cube wireframe etc.

**Path Code:** P40262

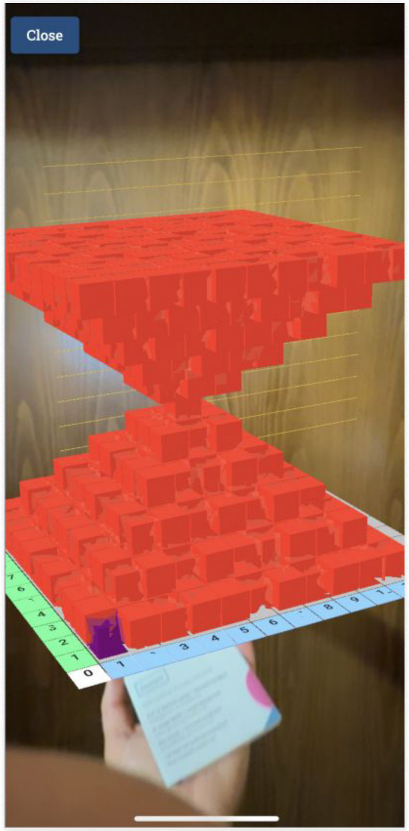
# Building Cubes [Upper Secondary Education]

**Focus on Computational Thinking skill:**
Algorithmic Thinking, Generalization, Automation and Debugging

- The more complex shapes include the use of nested loops in symmetric corner, tunnel, table, chair, and hourglass
- Dependent nested loops are addressed in the shapes square pyramid, corner stairs
- Shapes that address Decomposition are house front, house, shelves, spiral
- Teachers can finetune the difficulty of the tasks by changing the settings



```
Initialise cubes
set height to 6
count with m from 1 to height by 1
do  set cubes to height × 2 - m
    count with n from m to cubes by 1
    do  count with o from m to cubes by 1
        do  Set a block at x: n y: o z: m
count with i from 11 to 7 by -1
do  count with j from 12 - i to i by 1
    do  count with k from 12 - i to i by 1
        do  Set a block at x: j y: k z: i
```



Path Code: P79260

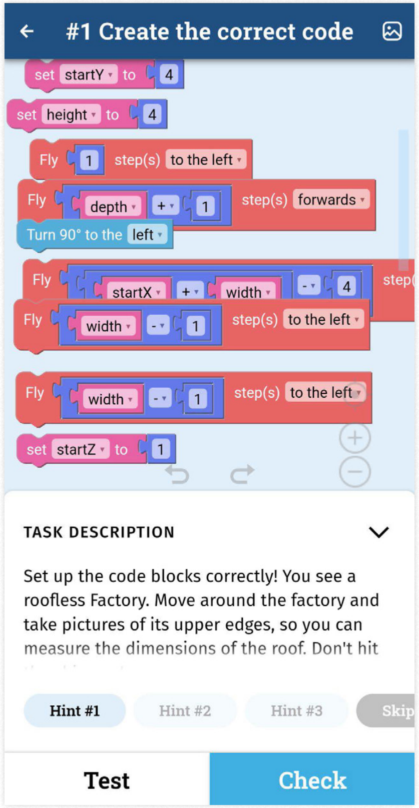
# Drone [Upper Secondary Education]

**Focus on Computational Thinking skill:**

Algorithmic Thinking, Generalization, Automation and Debugging

- Students can implement an algorithm for a flight in different scenarios
- Scenarios as flying over a pillar, billboard, roofless factory, or through a window in a wall

Path Code: P46348

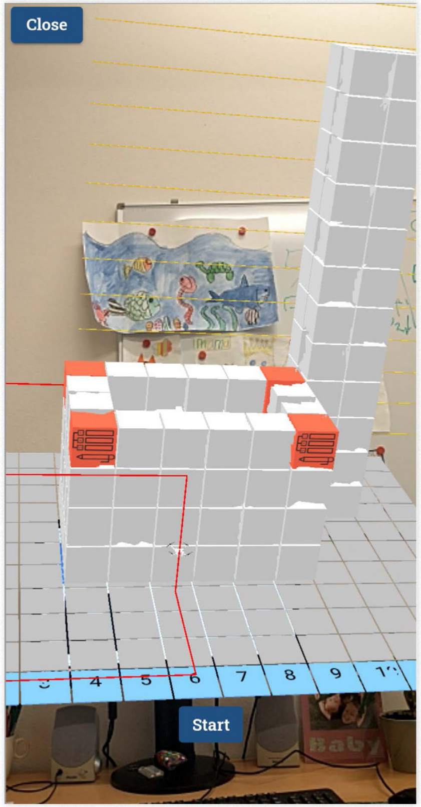


#1 Create the correct code

```
set startY to 4
set height to 4
Fly 1 step(s) to the left
Fly depth + 1 step(s) forwards
Turn 90° to the left
Fly startX + width - 4 step(
Fly width - 1 step(s) to the left
Fly width - 1 step(s) to the left
set startZ to 1
```

TASK DESCRIPTION

Set up the code blocks correctly! You see a roofless Factory. Move around the factory and take pictures of its upper edges, so you can measure the dimensions of the roof. Don't hit



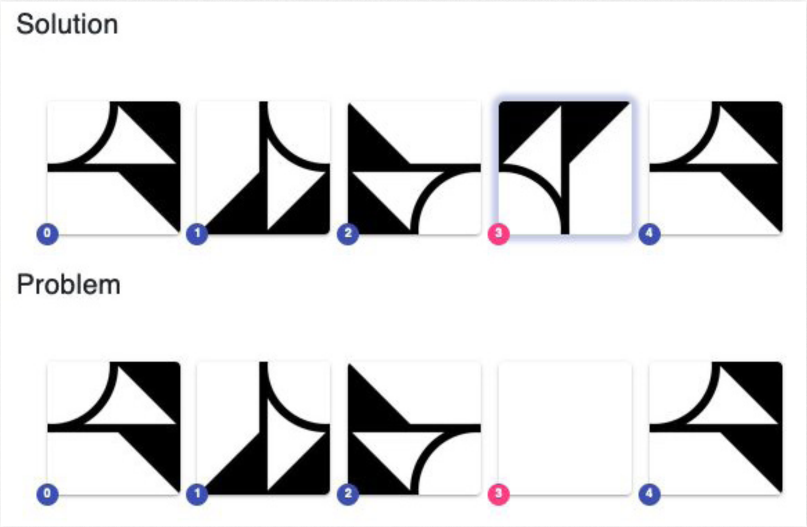
# Patterns [Lower Secondary Education]

**Focus on Computational Thinking skill:**

Pattern Recognition

- Is especially suitable for lower-secondary students on transition from arithmetic to algebraic reasoning and to work with Generalisation

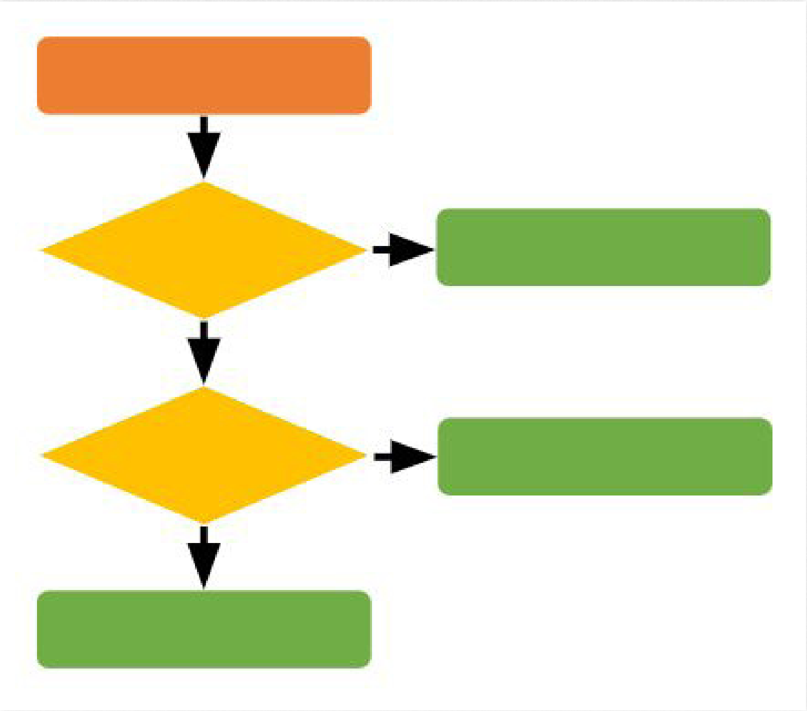
**Path Code:** P98267



# Free Task [Lower & Upper Secondary Education]

**In the Task Family Free Task there are three assignment types, i.e. ‘Match Exact Text’, ‘Free Text’, ‘Picture’. By using these types, teachers can tailor their own tasks and questions, and focus on a particular Computational Thinking skill.**

**Example paths about a drawing robot, process diagrams, and graph algorithms, respectively, illustrate the possible use of this Task Family and are available in the portal.**



Path Codes: P76293 // P92285 // P80289 // P45321

# IO 3+4: Didactic Content

colette-project.eu  portal.colette-project.eu

The creation of these resources has been (partially) funded by the ERASMUS+ grant program of the European Union under grant no. 2020-1-DE03-KA201-077363. Neither the European Commission nor the project's national funding agency PAD are responsible for the content or liable for any losses or damage resulting of the use of these resources.

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

<colette/>