

*Activity A2.4*

# Guidelines for Teachers:

---

## Good Practices in Online STEAM Education

*With a focus on Robotics, Coding & Creative Technology*

Status: PRE-FINAL | Author: R. Alimisi | Expected publication: April 2026

# 1. Introduction

---

The transition from physical to online teaching environments, especially in fields such as robotics, coding, and creative STEAM disciplines, requires far more than simply transferring content to digital platforms. It demands a fundamental rethinking of pedagogy, interaction, and learning design.

Online STEAM teaching must achieve a delicate balance between conceptual understanding, hands-on experimentation (physical or simulated), and sustained student engagement and collaboration. When these three pillars are addressed deliberately, virtual classrooms can become inclusive spaces of experimentation and creation environments where a student in a rural village can build and simulate a robot just as meaningfully as one sitting in a well-equipped laboratory.

This document, produced as part of Activity A2.4, brings together key findings from educational research and classroom practice to provide actionable guidance for teachers working in online or hybrid STEAM contexts. It is structured to move from the challenges teachers face, to the principles that guide effective responses, to practical techniques and concrete examples grounded in the realities of robotics and creative coding classrooms. Teacher personas and case studies are included throughout to illustrate how these ideas play out with real students.

## Who is this document for?

This guide is primarily addressed to:

- Teachers of robotics, coding, and digital making in lower and upper secondary education
- STEAM educators transitioning from face-to-face to online or hybrid delivery
- Teacher trainers and curriculum developers working within EU-funded projects or national initiatives
- School coordinators designing online STEAM programmes

The guidance is applicable to synchronous (live online sessions), asynchronous (self-paced), and blended approaches, though most examples are grounded in synchronous delivery where interaction design is most critical.

## 2. Core Challenges in Online STEAM Education

Online STEAM education, and robotics and coding in particular, presents a distinct set of challenges that stem primarily from translating hands-on, experiential learning into digital environments. Understanding these challenges is the necessary first step toward addressing them effectively.

### 2.1 Limited Access to Physical Hardware

Robotics kits, sensors, microcontrollers, and physical prototyping materials are central to traditional robotics education. When students learn from home, access to hardware is often absent, inequitable, or logistically complicated. This restricts students' ability to engage in tangible experimentation and iterative prototyping, arguably the heart of engineering learning. In addition, not all the schools are well-equipped with labs or have access to open labs.

#### Opportunity

Simulation environments such as Tinkercad Circuits, Open Roberta Lab, micro:bit MakeCode simulator, and Scratch offer credible alternatives that preserve much of the exploratory value of physical prototyping, particularly when paired with structured inquiry tasks.

### 2.2 Reduced Student Engagement

In online settings, passive participation can easily replace active involvement if interaction is not carefully structured. The absence of physical presence removes many of the social cues, body language, peer proximity, shared tools, that naturally sustain attention in a classroom. Students may appear to be present while their minds have wandered, or may disconnect entirely without their teacher noticing.

Research consistently shows that engagement in online learning is strongly linked to interaction frequency and quality (Chen et al., 2018). Without deliberate design, lessons can quickly become one-way broadcasts.

### 2.3 High Cognitive Load

Students in online STEAM sessions are simultaneously navigating digital interfaces, following instructions, understanding abstract concepts, writing or running code, and troubleshooting, often without immediate support. This concurrent demand on working memory constitutes a significant cognitive load that can overwhelm learners, especially beginners, and lead to disengagement or surface-level engagement (Chen et al., 2018).

### 2.4 Diverse Skill Levels

A typical online STEAM classroom contains students who range from complete beginners to confident coders or makers. This diversity is further amplified in online settings, where individualised support is harder to deliver. Beginners may feel lost and unable to ask for help discreetly, while more advanced students may disengage due to insufficient challenge.

### 2.5 Difficulty Providing Real-Time Support

In physical classrooms, teachers move around the room, glance at screens, and intervene at the exact moment a student encounters a bug or a conceptual stumbling block. Online, this ambient monitoring is almost impossible. Delays in feedback can hinder learning progression and increase frustration, particularly in robotics and coding, where errors and debugging are integral to the learning process.

## 2.6 Summary Table

| Challenge                        | Impact on Learning                               | Possible Response                                            |
|----------------------------------|--------------------------------------------------|--------------------------------------------------------------|
| <b>Limited hardware access</b>   | Reduced experimentation and hands-on learning    | Simulation platforms (Tinkercad, Open Roberta Lab, MakeCode) |
| <b>Reduced engagement</b>        | Passive participation, dropout risk              | Breakout rooms, gamification, frequent checkpoints           |
| <b>High cognitive load</b>       | Confusion, shallow understanding                 | Chunked tasks, visual scaffolding, minimal tool stack        |
| <b>Diverse skill levels</b>      | Frustration for beginners; boredom for advanced  | Tiered tasks, peer mentoring, UDL approaches                 |
| <b>Limited real-time support</b> | Errors go unresolved; frustration builds         | Structured checkpoints, peer debugging, forums               |
| <b>Tool proliferation</b>        | Cognitive fragmentation, time lost on navigation | Single platform + one learning tool principle                |

### 3. Key Principles for Effective Online STEAM Teaching

The literature on effective online STEAM education, together with practitioner experience, converges on a set of guiding principles. These are not rigid rules but orientations that should inform planning, tool selection, and moment-to-moment decisions in the virtual classroom. Crucially, research consistently emphasises that the primary barriers to online STEAM education are instructional — not technological. Well-designed lessons can overcome most technical limitations (Chen et al., 2018).

#### 3.1 Pedagogy Before Technology

The most important principle is the most easily overlooked: digital tools should be selected and used in service of clearly defined learning goals, not the other way around. Teachers who begin with 'what can this tool do?' rather than 'what do I want my students to understand or create?' risk building lessons around features rather than learning.

When teachers prioritise pedagogical clarity what students should understand, create, or achieve — they are better positioned to make purposeful use of technology and create coherent, impactful learning experiences.

##### In practice

Before choosing a platform or app, ask:

- What is the specific learning objective for this session?
- Which interaction does this tool support exploration, discussion, assessment, or creation?
- Could this objective be achieved with a simpler or more familiar tool?

#### 3.2 Inclusion and Confidence Building

Online STEAM contexts must be actively designed to be inclusive, particularly for beginners and students from underrepresented groups. The initial encounter with abstract concepts and unfamiliar tools in a virtual setting can be profoundly intimidating. Teachers are encouraged to adopt a 'low floor, high ceiling, wide walls' approach (Resnick & Silverman, 2005): ensuring all learners can enter the activity, while creating room for extension and lateral exploration.

Early success experiences are especially important. Small, achievable tasks completed early in a course or session build the self-efficacy that enables students to take on greater challenges later.

#### 3.3 Minimising and Stabilising the Technological Environment

Rather than introducing multiple platforms that fragment attention, teachers should prioritise the use of one or two reliable, accessible tools that support the core learning objectives. This reduces cognitive friction and allows students to focus on understanding concepts rather than navigating complex digital ecosystems.

Consistency matters: when students know exactly where to find materials, how to join a breakout room, and where to submit work, they can devote their mental energy to the actual learning content.

#### 3.4 Visualisation First

Many core STEAM concepts, such as control flow, sensor logic, conditional structures, feedback loops, are inherently abstract. The principle of 'visualisation first' holds that these ideas should be made visible through demonstrations, simulations, and step-by-step modelling before students are asked to reproduce or modify them.

In online robotics and coding education, showing how a system behaves before asking students to replicate it significantly enhances comprehension and supports diverse learning styles. Live screen-sharing of code being written and tested, annotated diagrams, and simulation walkthroughs are all effective visualisation strategies.

### 3.5 Structured Interaction

Meaningful engagement does not occur spontaneously in online environments. Teachers must intentionally embed interaction opportunities throughout lessons. These include guided questions, chat-based prompts, collaborative problem-solving in breakout rooms, polls, and peer feedback moments. The goal is to ensure students are active participants at regular intervals, not passive consumers of a broadcast.

Research on online STEM courses shows that structured peer discussions and instructor-student interaction are among the strongest predictors of student success and course completion (Chen et al., 2018).

### 3.6 Managing Cognitive Load

Online STEAM lessons should be carefully paced, focusing on a limited number of key ideas per session and building in time for practice and reflection before introducing new content. Worked examples, partially completed code scaffolds, and visual step-by-step guides all help reduce the extraneous cognitive load that can overwhelm learners.

The principle of 'one tool at a time' applies here too: introducing a new platform and a new concept simultaneously can easily exceed students' working memory capacity.

### 3.7 Formative Assessment as a Continuous Thread

Frequent, low-stakes formative assessment helps teachers monitor understanding in real time and helps students self-regulate their learning. In online settings, this is especially important because the ambient monitoring of a physical classroom is absent. Tools such as Kahoot, Mentimeter, padlets, and in-chat polls can be integrated at multiple points in a lesson, not just at the end, to keep assessment woven into the learning experience rather than separated from it.

## 4. Practical Techniques for Interactive Online STEAM Teaching

Translating pedagogical principles into practice requires concrete, structured strategies that actively engage students. This section describes key techniques, with reference to robotics and creative coding contexts. Each technique is connected to its underlying pedagogical purpose to help teachers make informed choices about when and how to use it.

### 4.1 Breakout Rooms for Collaborative Problem-Solving

Breakout rooms are one of the most powerful tools available to online teachers, but their value depends entirely on how they are structured. Unstructured group time in breakout rooms frequently leads to off-task behaviour, unequal participation, or silence.

Effective breakout room design includes:

- **Clear, time-bound tasks:** Students should know exactly what they are expected to produce or decide within the allocated time.
- **Defined roles:** Assigning roles (e.g., driver/coder, navigator, note-taker, presenter) ensures accountability and gives every student a clear contribution.
- **Structured prompts:** A shared digital document or slide with guiding questions anchors the group's work.
- **Reporting back:** Brief report-outs from each group to the main room reinforce accountability and create shared learning moments.

#### Robotics Example

Groups of three are given a scenario: 'Design a system that automatically waters plants when the soil is dry.' Each group uses a shared Tinkercad workspace to sketch their sensor-actuator logic, with roles of 'engineer', 'tester', and 'presenter'. After 15 minutes, each presenter shares one design decision and one problem they encountered.

### 4.2 Gamification and Rapid Engagement Techniques

Gamified approaches are particularly effective in online settings for maintaining attention, generating energy, and providing immediate formative feedback without interrupting lesson flow.

- Kahoot, Quizizz, and Mentimeter can be used as warm-ups, mid-lesson concept checks, or closing reflections. The key is intentional placement not gamification for its own sake.
- 'Live coding challenges' where students race to implement a specific function or debug a broken script within a time limit create focused engagement and authentic problem-solving.
- Leaderboards and badges, used judiciously, can motivate participation but should be designed so that participation itself (not just correctness) is rewarded, to protect less confident students.

#### Creative Coding Example

At the start of a Scratch session on animation, the teacher shares a short broken project and asks students to post in the chat what they think is wrong. The first three correct answers earn a 'Debug Detective' badge displayed in the class padlet. Energy is immediately raised and the diagnostic activity surfaces prior knowledge.

### 4.3 Problem-Based Learning (PBL) Missions

Framing lessons around real-world challenges, 'missions', transforms the purpose of coding and robotics from abstract skill-building to meaningful problem-solving. PBL is among the most well-evidenced approaches in STEM education for developing deeper understanding, transferable skills, and student motivation (Prince, 2004).

Online PBL requires:

- **A compelling scenario:** The problem must feel relevant and authentic i.e. smart cities, environmental monitoring, accessibility technology, social campaign and more
- **Structured phases:** Problem exploration → ideation → prototyping/simulation → testing → reflection and presentation.
- **Scaffolding:** Checkpoint questions, partially completed templates, and teacher check-ins prevent students from getting stuck.
- **Reflection:** Students should articulate what worked, what didn't, and what they would do differently.

#### Robotics Example

Students are presented with the 'Smart Crossroads' mission: design a traffic light system that adapts based on pedestrian and vehicle sensor data. Working in pairs over two sessions, they plan their logic using a flowchart whiteboard, code it in the micro:bit simulator, and present a short screencast explaining their solution.

## 4.4 Visual Scaffolding and Live Demonstrations

Given the abstract nature of much STEAM content, visual support is not a supplement — it is a core instructional strategy. Effective visual scaffolding in online robotics and coding teaching includes:

- Live screen-sharing of code being built incrementally, including deliberate 'thinking aloud' about decisions and corrections.
- Annotated diagrams and flowcharts on digital whiteboards (Mural, Miro, [...]) that students can see built in real time.
- 'Predict before run' moments: the teacher shows a piece of code and asks 'what will this do?' before executing it, activating predictive reasoning.
- Asynchronous video recordings with annotations that students can rewatch at their own pace.

#### Creative Coding Example 1

Before students create their own Scratch game character, the teacher shares their screen and codes a simple animated sprite from scratch, narrating their decisions: 'I'm choosing a repeat loop here because I want this to happen continuously, not just once, what other ways could we achieve this?' Students take annotated notes on a shared document. The teacher then reviews the answers.

#### Creative Coding Example 2

The students are given a specific artwork (i.e. The Nighthawks of Edward Hopper) and they are invited to bring some interactivity in the painting (i.e. moving characters, turning on and off the bar lights, adding music and more). The teacher selects and shows the result (i.e. turning on and off the lights in the bar) and then asks ways according to which this could be achieved? Students take annotated notes on a shared document.

## 4.5 Student-as-Teacher and Peer Learning

When students explain concepts to each other, they deepen their own understanding, develop communication skills, and create a more inclusive, peer-supported learning community. In online settings, peer learning requires intentional design.

- 'Micro-teaching' moments: a student explains their solution to a coding problem in 60 seconds before the class moves on.
- Peer code review: students are paired to review and comment on each other's Scratch projects or micro:bit code, using a simple structured rubric.

- Peer mentoring channels: a dedicated Teams thread, or forum where students can ask and answer questions between sessions.

## 4.6 Interactive Digital Whiteboards

Digital whiteboards (Miro, Mural, Padlet ) extend visual thinking into collaborative space. In online STEAM teaching, they can serve as:

- Collective brainstorming canvases for the ideation phase of a PBL mission.
- Co-created system diagrams where groups contribute their component to a shared architecture.
- Exit ticket boards where students post one thing they learned and one question they still have.

Importantly, whiteboards create visible artifacts of student thinking that the teacher can review during and after the lesson to identify misconceptions and plan next steps.

## 4.7 Asynchronous Extension via Forums and Shared Spaces

Not all learning happens in live sessions, and not all students can contribute confidently in real time. Online forums and shared spaces extend interaction beyond synchronous lessons and provide a lower-pressure environment for reflection, question-asking, and peer support.

In robotics and coding contexts, a 'Debugging Wall', a dedicated thread where students post bugs they have encountered and solutions they have found creates a living, community-built knowledge resource.

## 4.8 Summary of Techniques

| Technique                       | Pedagogical Purpose            | Example in Robotics/Coding           | Platforms                     |
|---------------------------------|--------------------------------|--------------------------------------|-------------------------------|
| <b>Breakout Rooms</b>           | Collaboration, peer learning   | Design a smart traffic system        | Zoom, Teams                   |
| <b>Kahoot / Quizizz</b>         | Formative assessment, energy   | Identify correct sensor logic        | Kahoot, Quizizz, Mentimeter   |
| <b>Live Demos + Think-Aloud</b> | Modelling, visualisation       | Build a Scratch sprite step by step  | Screen share, OBS Studio      |
| <b>Digital Whiteboards</b>      | Co-creation, brainstorming     | Draw robot system flowchart          | Miro, Mural, Padlet           |
| <b>PBL Missions</b>             | Authentic problem-solving      | Smart crossroads challenge           | Simulation + shared docs      |
| <b>Peer Code Review</b>         | Peer learning, communication   | Review partner's Scratch game        | Google Docs, Teams, Padlet    |
| <b>Discussion Forums</b>        | Inclusion, async reflection    | Debugging Wall thread                | Slack, Teams, Moodle          |
| <b>Student Micro-Teaching</b>   | Deep understanding, confidence | Explain your code solution in 60 sec | Screen share, video recording |

## 5. From Theory to Practice: Teacher Personas

The following personas illustrate how different teachers navigate the challenges and apply the principles described in this document. Each persona is composite, drawn from common patterns observed in online STEAM education practice, and is intended to help teachers recognise aspects of their own contexts and draw inspiration for their practice.

### Persona 1: Maria - The Beginner Supporter

|                                            |                                                                                                        |
|--------------------------------------------|--------------------------------------------------------------------------------------------------------|
| <b>Maria</b><br>Lower Secondary<br>Teacher | <b>Age group:</b> Students aged 12–13                                                                  |
|                                            | <b>Key tools:</b> MakeCode Simulator, Scratch, Padlet, breakout rooms                                  |
|                                            | <b>Main challenge:</b> Building confidence with students who have little or no prior coding experience |

#### Profile

Maria teaches a mixed-ability class of 25 students, the majority of whom have never written a line of code. She is acutely aware that the initial online encounter with programming can be daunting and that losing a student in the first two sessions may mean losing them permanently. Her approach is anchored in emotional safety, gradual skill progression, and the belief that mistakes are data, not failures. She designs every session around the question: 'How can I make sure every student feels they can do this?'

#### Practice in Action

Maria never begins a new topic with an open-ended task. She opens with a guided exploration: students run a pre-built MakeCode simulator program, observe what it does, and post one observation in the chat. This 'notice and wonder' opening has zero failure states every student can contribute.

She structures breakout rooms so that each pair includes one slightly more experienced student who acts as the 'driver', while the less experienced student reads the instructions aloud and makes suggestions roles Maria has explained and rehearsed. She checks each room briefly every four minutes, narrating what she sees: 'I love this approach you've found a different way to solve this than the group next door.'

When students get stuck, Maria doesn't immediately provide answers. She asks: 'What have you tried? What happened?' This metacognitive scaffolding builds problem-solving resilience, not just code that works.

#### ERGO Key Takeaway for Teachers

Design for the most anxious learner in the room. If your entry task has a right and wrong answer that is immediately visible to peers, beginners will disengage before they begin.

### Persona 2: Thomas - The Tool Optimizer

|                                              |                                                                                 |
|----------------------------------------------|---------------------------------------------------------------------------------|
| <b>Thomas</b><br>Experienced STEM<br>Teacher | <b>Age group:</b> Students aged 13–15                                           |
|                                              | <b>Key tools:</b> Zoom, Tinkercad, shared Google Slides                         |
|                                              | <b>Main challenge:</b> Reducing cognitive overload caused by too many platforms |

#### Profile

Thomas is a confident, technology-enthusiastic teacher who, in his first year of online teaching, assembled an impressive digital toolkit: one platform for video calls, another for coding, a third for sharing resources, a fourth for quizzes, and a fifth for discussion. He noticed, however, that students were spending more time asking 'where do I go now?' than thinking about circuits or sensors. He radically simplified.

## Practice in Action

Thomas now runs all sessions on a single video platform, with a consistent lesson structure that students could navigate with their eyes closed: five-minute warm-up quiz in Mentimeter, demonstration with live Tinkercad screen-share, 20-minute build task, pair share in breakout, whole-group reflection via the chat.

He created a one-page 'Digital Classroom Map' a simple visual showing which tool is used when and posts it at the start of every session. New tools are only introduced when they directly serve a learning goal that cannot be achieved with existing tools, and they are introduced one at a time, with a dedicated 10-minute orientation session.

Thomas also produces two-minute 'how-to' video clips for each tool and makes them available in a shared folder. These eliminate the need for him to explain navigation during lesson time.

## ERGO Key Takeaway for Teachers

Every minute spent explaining a tool is a minute not spent on learning. Ruthlessly simplify your digital environment, and invest the time saved in richer interaction.

## Persona 3: Helena - The Visual Explainer

|                                             |                                                                                            |
|---------------------------------------------|--------------------------------------------------------------------------------------------|
| <b>Helena</b><br>Experienced STEAM Educator | <b>Age group:</b> Students aged 13–15                                                      |
|                                             | <b>Key tools:</b> Digital whiteboards, screen share + annotation, simulator (Open Roberta) |
|                                             | <b>Main challenge:</b> Making abstract programming concepts visible and comprehensible     |

## Profile

Helena is a STEAM educator who has spent years building her understanding of how students develop mental models of systems. She has observed that most misconceptions in robotics and coding, infinite loops that students don't intend, sensors that behave unexpectedly, variables that get overwritten, arise not from carelessness but from the fact that the code is invisible: students cannot see the system running inside the machine. Her teaching philosophy is built around making the invisible visible.

## Practice in Action

When introducing conditional logic, Helena doesn't start with code. She opens a blank Miro board and draws a flowchart collaboratively with the class, asking students to contribute decision nodes via sticky notes. Only when the logic is clear on the board does she open the coding environment.

Her live coding demonstrations are structured around the 'predict-run-explain' cycle: she shows a piece of code, asks students to type their prediction of the output into the chat, runs it, and then collectively analyses why predictions were or weren't correct. This approach is particularly effective for surfacing and correcting misconceptions about loops and sensor thresholds.

She regularly records short annotated screencasts after each session, 'lesson replay' videos of two to three minutes that highlight the three key concepts from the class. Students report these are among the most valued resources in the course.

## ERGO Key Takeaway for Teachers

Abstraction is the enemy of understanding in early STEAM learning. Always provide a visual anchor, a diagram, a simulation, a physical analogy, before asking students to work with symbolic or textual representations.

## Persona 4: Alex - The Interaction Designer

|                                            |                                                                                             |
|--------------------------------------------|---------------------------------------------------------------------------------------------|
| <b>Alex</b><br>Secondary School<br>Teacher | <b>Age group:</b> Students aged 13–16                                                       |
|                                            | <b>Key tools:</b> Zoom breakout rooms, Padlet, Kahoot, collaborative coding platforms       |
|                                            | <b>Main challenge:</b> Preventing passive participation and ensuring equitable contribution |

### Profile

Alex came to online teaching from a background in collaborative and project-based learning. He was quickly frustrated by the tendency of online sessions to default to a teacher-talks, students-listen dynamic. He views learning as fundamentally social and has invested significant effort in designing interaction structures that make genuine collaboration happen, not just co-presence.

### Practice in Action

Every lesson Alex teaches is planned with an interaction map: a diagram showing when each type of student interaction is designed to occur (teacher-to-student, student-to-student, student-to-content), and which tool enables it. He aims for no more than seven minutes without a student interaction moment.

His breakout rooms are highly structured. Each group has a shared slide in a collaborative presentation tool. The slide has four sections, one for each role (programmer, debugger, designer, presenter), and students fill in their sections during the activity. When groups reconvene, the slide also serves as the basis for the report-out, reducing the preparation time needed.

Alex uses a 'cold call with safety net' technique for whole-class questioning: he nominates a student to answer, but immediately offers them the option to 'phone a friend' (ask a breakout partner) or 'ask the audience' (post the question to the chat). This maintains a sense of active readiness without creating anxiety.

### ERGO Key Takeaway for Teachers

Interaction in online lessons does not happen by itself. It must be designed, timed, and structured, with explicit role assignments, clear tasks, and regular opportunities to contribute.

## Persona 5: Paul - The Project Designer

|                                                 |                                                                                          |
|-------------------------------------------------|------------------------------------------------------------------------------------------|
| <b>Paul</b><br>Upper Secondary STEM<br>Educator | <b>Age group:</b> Students aged 14–17                                                    |
|                                                 | <b>Key tools:</b> Scratch, micro:bit simulator (MakeCode), Tinkercad, shared Google Docs |
|                                                 | <b>Main challenge:</b> Connecting robotics and coding to meaningful real-world problems  |

### Profile

Paul teaches upper secondary students who often ask 'why are we doing this?' and deserve a good answer. He is motivated by the conviction that robotics and coding should be taught not as sets of isolated technical skills, but as tools for understanding and improving the world. His course is organized not by syntax or sensor type, but by challenge: each unit opens with a real-world problem that students work toward solving.

### Practice in Action

Paul's current unit opens with a five-minute documentary clip about floods in coastal cities and a question: 'How could sensing and automation help communities prepare for or respond to flooding?' Students spend ten minutes in breakout rooms generating ideas without any coding constraint. This problem-framing phase is deliberate, it positions the technical work that follows as a means to an end, not the end itself.

Over the following three sessions, students develop a simulation of a flood sensor and warning system in Tinkercad and micro:bit MakeCode. The final session is a peer showcase, a 'Demo Day', where groups present their simulations via screen-share to the class and receive structured feedback using a peer review rubric focused on functionality, clarity of design, and real-world plausibility.

Paul integrates reflection throughout. At the end of each session, students update a shared 'design journal' documenting their decisions, problems, and next steps. These journals are shared with him and inform his planning for the following session.

### ERGO Key Takeaway for Teachers

Motivation in online learning is sustained by purpose. When students understand why they are learning to program sensors, they are more willing to persist through debugging and difficulty.

## Persona 6: Leila - The Creative Coding Enabler

|                                                       |                                                                                    |
|-------------------------------------------------------|------------------------------------------------------------------------------------|
| <b>Leila</b><br>Arts & STEAM Cross-Curricular Teacher | <b>Age group:</b> Students aged 11–14                                              |
|                                                       | <b>Key tools:</b> Scratch, Padlet                                                  |
|                                                       | <b>Main challenge:</b> Integrating artistic expression with computational thinking |

### Profile

Leila works at the intersection of arts and technology, teaching in a school that emphasises creative and interdisciplinary approaches. She has found that students who resist 'coding lessons' often engage enthusiastically when the same computational concepts are encountered through music, animation, or visual art. She uses creative coding, coding as a medium for artistic expression, as her primary entry point into computational thinking.

### Practice in Action

A typical Leila lesson opens not with a coding environment, but with a stimulus: a short animation, an interactive artwork, or a piece of algorithmically-generated music. Students discuss what they notice, what they imagine the code behind it looks like, and what they would create if they had the same tools.

Using Scratch, Leila introduces variables, loops, and conditional logic through animation projects: students create a 'generative artwork', a pattern that changes based on user input or random values. The computational concepts are identical to those in any robotics lesson, but the creative frame lowers the perceived risk and raises motivation, particularly for students who don't identify as 'technical'.

She uses Padlet as a class gallery: every student's project is posted and anyone can leave a comment using a structured prompt ('I notice... I wonder... What if...'). This creates a peer feedback culture that is appreciative rather than evaluative and extends engagement beyond the live session.

### ERGO Key Takeaway for Teachers

STEAM, not STEM: the Arts dimension is not decoration. Creative coding provides a powerful on-ramp to computational thinking for students who may not respond to technical or engineering framings.

## 6. Making Online STEAM Teaching More Engaging: Evidence-Based Strategies

---

Engagement in online learning is not a single thing: it has cognitive, behavioural, and emotional dimensions (Fredricks et al., 2004). Students can be behaviourally compliant (cameras on, task completed) while being cognitively absent. The goal is not compliance but genuine intellectual and social involvement.

The following evidence-informed strategies, drawn from research on online STEM education (Chen et al., 2018; Prince, 2004) and digital pedagogy more broadly, are particularly effective in robotics and coding contexts.

### 6.1 Hook and Launch Every Session Intentionally

The first five minutes of an online lesson determine whether students settle into active engagement or drift into passive attendance. Effective hooks include:

- **A broken code challenge:** Students are shown a non-functioning Scratch project or micro:bit program and asked to diagnose the problem. No context needed; the challenge is immediate.
- **A provocative question:** 'Could a robot make a fair referee? Why?'- posted in chat for 90 seconds of silent reflection before a brief discussion.
- **A short 'predict the output' demo:** A piece of code shown on screen; students type their prediction into the chat before it runs.

### 6.2 Keep the Cognitive Engine Running: The 7-Minute Rule

In online settings, student attention drifts significantly after sustained passive periods. A practical guideline is to ensure that no more than seven minutes pass without a student action: a chat response, a vote, a short task, a breakout moment, a prediction. This rhythm keeps the cognitive engine engaged and provides the teacher with continuous, low-cost formative data.

### 6.3 Make Thinking Visible

Students engage more deeply when their thinking is externalised and visible to themselves, their peers, and their teacher. Strategies include:

- Shared collaborative documents where students annotate their reasoning as they code.
- Digital whiteboards where groups map out a system before coding it.
- 'Show your thinking' chat prompts where students explain their next step before executing it.
- Design journals that document decisions, not just outcomes.

### 6.4 Connect to Student Identity and Experience

Engagement is sustained when students perceive a connection between the learning and their own lives, interests, or futures. In coding and robotics, this means:

- Offering missions with genuine social relevance: accessibility technology, environmental monitoring, community safety.
- Allowing student choice in project themes: a student who builds a Scratch game about their favourite sport will invest more deeply in understanding variables and loops.
- Explicitly discussing careers and communities where these skills are used including non-obvious ones like music production, healthcare, architecture, and storytelling.

## 6.5 Celebrate Errors and Debugging Culture

In robotics and coding, debugging is not a failure mode, it is the primary learning mode. Teachers who explicitly celebrate interesting errors, create 'bug museums' in shared documents, and model their own debugging process in live demonstrations create a culture where mistakes are assets, not embarrassments. This is especially important in online settings where students may be more inhibited about sharing difficulties.

## 6.6 Use Storytelling and Narrative

Framing technical challenges as stories or missions dramatically increases engagement. 'A city council needs your help...' or 'A team of marine biologists has a problem...' transforms a coding exercise into a narrative that students want to resolve. This is particularly effective with younger students but has proven value across all age groups in STEAM contexts.

## 7. Building Your Own Online Teaching Style

---

Developing an effective online teaching style in STEAM should not be understood as adopting a fixed methodology, but as assembling and constantly refining a personal pedagogical repertoire. No single approach works for every teacher, class, or context. The personas described in Section 5 are not prescriptions, they are invitations to reflection.

Teachers are encouraged to begin with a self-audit, honestly assessing their current practice against the principles outlined in this guide. The following questions can support that reflection:

- Where in my lessons do students shift from receiving to doing? How frequently does this happen?
- What is the simplest technology that would support my key learning objective for each session?
- How do I currently know, in real time, whether students are understanding or confused?
- What visual representations do I make available to support students who struggle with abstraction?
- How do I currently support the least confident student in an online session?

**Iteration is key.** Effective online teachers consistently describe their development as a process of trying one new strategy per session, observing what happens, and adjusting. Small experiments — one structured breakout room, one predict-the-output moment, one Debugging Wall thread — are more sustainable and revealing than wholesale redesign.

**Collaboration with colleagues**, whether within the same school or across the partner network of this project, is also a powerful developmental resource. Peer observation, co-planning sessions, and shared repositories of missions and tasks accelerate growth in ways that individual reflection alone cannot.

## 8. Teacher Toolbox: Online STEAM Teaching Toolkit

The following toolkit provides a quick reference for teachers planning online STEAM sessions. Tools are organised by pedagogical function — what they help you achieve — rather than by technology category.

| Strategy / Tool                            | Design Element                  | Pedagogical Purpose                  | Example in Robotics/STEAM                        |
|--------------------------------------------|---------------------------------|--------------------------------------|--------------------------------------------------|
| <b>Breakout Rooms</b>                      | Active / Peer Learning          | Collaboration, problem-solving       | Design a smart irrigation system in small groups |
| <b>Kahoot / Quizizz</b>                    | Formative Assessment            | Check understanding, energy          | Identify correct sensor logic                    |
| <b>Digital Whiteboards</b>                 | Active Learning / Visualisation | Externalise thinking, co-create      | Draw robot system flowchart                      |
| <b>Discussion Forums / Debugging Wall</b>  | Async Engagement                | Inclusion, peer support, reflection  | Post bugs and solutions between sessions         |
| <b>Instructor Videos + Annotated Notes</b> | Instructor Content              | Clarity, re-watchability             | Video: how ultrasonic sensors work               |
| <b>Peer Mentoring / Code Review</b>        | Peer Learning                   | Collaborative knowledge building     | Review partner's Scratch animation               |
| <b>Simulation Platforms</b>                | Hands-on Learning               | Experimentation without hardware     | Tinkercad circuit simulation, Webots robot       |
| <b>Flexible Tasks (UDL)</b>                | Universal Design                | Inclusivity, accessibility           | Choose: physical robot OR virtual simulation     |
| <b>Polls / Mid-lesson Checks</b>           | Formative Assessment            | Maintain attention, monitor progress | 'Which sensor fits this task?' live poll         |
| <b>PBL Missions / Design Challenges</b>    | Project-Based Learning          | Authentic context, motivation        | Smart crossroads; flood warning system           |
| <b>Padlet Gallery / Creative Showcase</b>  | Peer Feedback / UDL             | Appreciation, community, expression  | Share Scratch creative coding projects           |

## 9. References

---

Chen, B., Bastedo, K., & Howard, W. (2018). Exploring design elements for online STEM courses: Active learning, engagement & assessment design. *Online Learning*, 22(2), 59–75.  
<https://doi.org/10.24059/olj.v22i2.1369>

Fredricks, J. A., Blumenfeld, P. C., & Paris, A. H. (2004). School engagement: Potential of the concept, state of the evidence. *Review of Educational Research*, 74(1), 59–109.

Haak, D. C., HilleRisLambers, J., Pitre, E., & Freeman, S. (2011). Increased structure and active learning reduce the achievement gap in introductory biology. *Science*, 332(6034), 1213–1216.

Prince, M. (2004). Does active learning work? A review of the research. *Journal of Engineering Education*, 93(3), 223–231.

Resnick, M., & Silverman, B. (2005). Some reflections on designing construction kits for kids. *Proceedings of the 2005 Conference on Interaction Design and Children*.

Yamal, J. M. (2023). Teaching and designing online STEM courses to support self-directed learning skills. *Post-Secondary Collaborative*. <https://postseccollab.org>

Additional resource: Cambridge ELT Blog (2023). Enhancing engagement: 6 strategies for interactive and effective online learning. <https://www.cambridge.org/elt/blog/2023/11/30/enhancing-engagement-6-strategies-for-interactive-and-effective-online-learning/>

<https://www.cambridge.org/elt/blog/2023/11/30/enhancing-engagement-6-strategies-for-interactive-and-effective-online-learning/>

<https://olj.onlinelearningconsortium.org/index.php/olj/article/view/1369/378>