

UNIT 4: Materials and Resources Preparation



Structure of the Stage

- 1 Aim and Objectives
- 2 Outcomes
- 3 Main CT principles applied during this unit
- 4 Relationship with WP2 modules
- 5 Challenges
- 6 Justification
- 7 Activities
- 8 Additional Resources
- 9 What is the value of this stage?



1. Aim and Objectives

This unit aims to help adult trainers understand why careful preparation and selection of resources and materials are essential when introducing computational thinking (CT) in adult education. The same activity can lead to very different learning experiences depending on how it is presented.

OBJECTIVES

By the end of this stage, adult educators will be able to:

1. Analyse how different types of learning materials (digital and non-digital) support specific CT skills in adult education contexts;
2. Identify clear and practical criteria for selecting accessible and meaningful materials for adult learners;
3. Prepare learning activities that develop CT using non-digital and everyday resources familiar to adult learners;
4. Prepare and adapt resources that promote active participation, reflection, and the transfer of CT skills to everyday life.



2. Outcomes

By the end of this unit, adult trainers are able to:

1. Understand why the preparation and selection of resources and materials directly influence the development of CT skills in adult learners.
2. Understand how well-designed materials can make CT accessible, meaningful, and engaging for adult learners.
3. Design unplugged and technology-based activities that make CT accessible to adult and senior learners.
4. Facilitate learning experiences that encourage participation, reflection, and the application of CT to everyday problems.
5. Reflect on their role as trainers in organising, adapting, and evaluating learning materials according to learners' needs and contexts.



3. Main CT Principles Applied

All four CT principles are applied in this Unit:

- Decomposition
- Abstraction
- Pattern Recognition
- Algorithmic thinking



4. Relationship with WP2 Modules

This unit is closely linked to M3, "Designing Lesson Plans and Activities", which presents the structure and key elements of CT-based lesson planning. This unit builds on this content by guiding trainers in designing and completing learning resources. It also connects with M4, "Adapting CT Strategies for Different Learning Levels", supporting their adaptation to learners' needs and abilities.



5. Challenges

When adult trainers prepare materials and resources for a course on CT, they may soon realise that this task involves much more than choosing tools or activities.

Another challenge is related to understanding that CT resources truly support thinking, not just doing. Familiar, hands-on activities can be gradually adapted to develop the desired thinking skills.

Limited time for preparation can feel overwhelming. Creating clear, accessible, and meaningful resources often requires careful planning. The trick is to plan small, reusable activities and collaborate with colleagues; even simple templates can save time while keeping learning impactful.



6. Justification

Every teaching-learning process relies on materials that actively support the interaction between trainers and learners. In CT, materials play a particularly important role, as they help to make abstract thinking processes visible and tangible.

Familiar, everyday materials reduce cognitive load and allow learners to focus on developing CT skills rather than mastering complex technologies.



Studies show, that learners remember significantly more information when concepts are supported by well-designed materials rather than by explanation alone.



7.1 Activities

Let's look at the types of activities you can create within the framework of applying computational thinking in the classroom.

Unplugged Activities

Unplugged activities do not require digital devices and can help overcome barriers such as unfamiliarity with programming languages or limited access to digital resources. There is evidence that this type of activity supports learning in programming and facilitates access to robotics.

Tinkering Activities

Tinkering involves analysing the elements of an object, such as building blocks, puzzles, simulators, or program code, and then changing or modifying them. This shows how a small change can affect an algorithm or the solution to a problem.

Making Activities

These activities focus on enabling learners to solve problems, plan their work, select appropriate tools, communicate ideas, and connect different concepts.

Remixing Activities

In remixing activities, you share and modify existing code or algorithms in order to adapt them or combine them with other elements to solve a task, problem, or challenge.



7.1 Unplugged Activities

Who said you need computers to teach CT? Sometimes, the most powerful learning happens away from the screen. Unplugged activities help learners focus on thinking processes rather than technology, making learning more accessible and engaging.

Some practical examples include:

- Sequencing a routine by breaking it down into all intermediate steps
- Describing a drawing to a partner so they can reproduce it
- Writing a set of instructions (code) to guide another person in completing a construction, movement, or similar challenge
- Designing decision trees
- Working with algorithms
- Playing games to encode or decode binary code.



<https://www.csunplugged.org>



7.2 Technology-Based Learning Activities

One of the most well-known tools for visual programming is Scratch. Scratch makes it easier to move towards more advanced programming languages while allowing learners to experiment, create, and learn through trial and error. With Scratch, you can design quiz games to work on content, create simple applications or games, and make abstract concepts visible, visual, and intuitive.

Finally, it is also worth exploring platforms linked to robotics, such as Arduino and Micro:bit, as well as others associated with robots and devices like mBlock or LEGO.

These tools offer hands-on opportunities to connect CT with real-world actions, making learning both concrete and highly motivating.

Do you want to know more about Scratch?

LEARN MORE



Scratch is used worldwide not only in schools but also in universities, community centres, and adult learning programmes. Its strength lies in turning complex ideas into something learners can see, manipulate, and understand.



7.3.1. Activity. Tips for Content Design

When creating learning activities, adult trainers should ask themselves:

Who will use this content? How easily can it be adapted? And what kind of learning experience will it create? Keeping these questions in mind helps ensure that your materials truly support meaningful learning.

To achieve this, the content you design should be:

- **Accessible:** It should follow recommended accessibility guidelines so that learners with different abilities can use it. For adult learners this includes clear instructions, familiar objects and visual clarity.
- **Easily editable:** Activities should be created using free, simple, and cross-platform tools. When content is easy to modify, it becomes easier to adapt to different learners, contexts, and needs.
- **Innovative:** Learning materials should go beyond information delivery and act as tools for change, creativity, and innovation.

For this reason, a well-designed learning activity is one that:

- Makes the learning situation tangible, realistic, and feasible.
- Clearly describes the role of both learners and trainers, so everyone knows what is expected.
- Clearly explains how digital technology is used in the teaching-learning process.
- Promotes active learning, where learners develop competences through participation and reflection.

Can be linked to a specific subject area, or connect several areas at the same time, supporting cross-cutting learning.



7.3.2 Activity. Tips for Content Design

One of the most widely recognised frameworks is the Brennan–Resnick framework. Have you ever wondered how educators make sense of all the elements of CT? This framework is a powerful tool to guide you. It's organised around three dimensions: concepts, practices, and perspectives, helping you see the big picture while breaking it down into manageable parts.

With this framework at your disposal, you can develop clear criteria for key decisions:

- What do you need to learn as a trainer?
- What should you teach your learners?
- How can you integrate CT into the activities of your subject area?
- How can you assess learning effectively?
- What content should you select, prioritise or omit?
- How can you distinguish high-quality resources from less effective ones?

To explore it in more detail and deepen your understanding, you can consult: Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. Proceedings of the 2012 Annual Meeting of the American Educational Research Association. Vancouver: American Educational Research Association.



Click to download

[Brennan, K., & Resnick, M. \(2012\).](#)



7.4 Activity. Sorting Game with Everyday Objects

Provide learners with a collection of everyday objects (keys, buttons, coins, pens, etc.).

Ask them to work in small groups and sort the objects according to different criteria, such as size, colour, material, or function.



1. Let each group decide on their own sorting rule and explain why they chose it.
2. Ask them to change the rule and sort the same objects again using a different criterion.
3. Invite learners to describe their sorting process as a simple set of instructions (an algorithm).

This activity develops abstraction, classification, and algorithmic thinking skills, helping learners understand how computers organise and process information. When learners sort and re-sort objects using different rules, they practise the same logical processes that data scientists and programmers use every day!



7.5 Activity. Patterns Using Geometric Shapes

Provide your learners with a variety of geometric shapes (circles, squares, triangles, etc.) and ask them to create patterns using these shapes.

Tell them that they need to create a pattern by alternating circles and squares, or by progressively increasing the size of the triangles. Try to combine colours so they can also work on colour association, which is always effective.

Then, they need to express these patterns as simple algorithms, such as “circle, square, circle, square” or “add one triangle each time”. This activity develops pattern recognition and algorithmic thinking skills, helping learners learn how to identify, for example, errors in programming code.

Did you know that...?

Creating patterns with shapes and colours isn't just fun; it trains the same brain skills that programmers use to spot errors in code! By alternating shapes or increasing sizes, learners practise pattern recognition and algorithmic thinking, which are fundamental in coding, problem-solving, and even everyday decision-making.



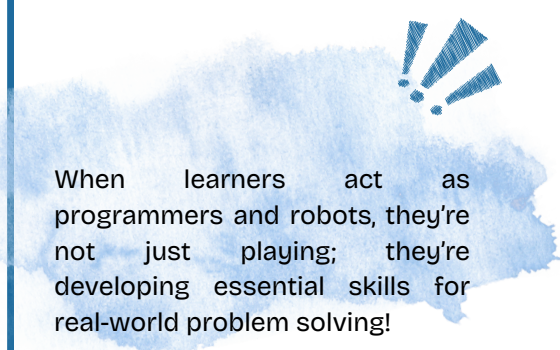
7.6 Activity. Role-Playing Game With Logic Problems

Organise role-playing activities in which learners act as programmers and robots, and must communicate with each other to solve logic problems.

Design a scenario in which a robot must follow a series of precise instructions to find a hidden treasure, using clear and concise commands.

This activity promotes collaboration, effective communication, and algorithmic thinking as learners work together to overcome challenges.

Encourage learners to work together so that they become familiar with collaborative development, a common approach in professional programming projects.



When learners act as programmers and robots, they're not just playing; they're developing essential skills for real-world problem solving!



7.7 Activity. Reflecting on Everyday Problems

In this activity, learners should discuss in pairs the problems they face in their daily lives. You will open the dialogue by asking: “What problems do you face every day at home or at school?”

You will write the ideas on the board. Your aim is to help learners become familiar with the concept of everyday problems that can be solved.

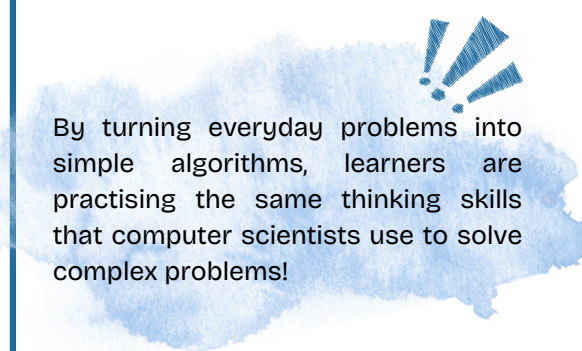
After 15 minutes, you will lead a whole-class brainstorming session, during which learners will share the problems they have discussed.

Next, explain the concept of an algorithm to them in a simple way, using everyday examples (such as making a sandwich or preparing a smoothie).

You will tell them that an algorithm is simply a series of steps that are followed to complete a task.

Afterwards, divide the class into groups of four to five learners, and have each group choose one problem from the board to work on together. They should clearly define the problem and then sketch a simple algorithm on paper.

In this way, they will begin to see how computational thinking can be used to solve everyday problems.



By turning everyday problems into simple algorithms, learners are practising the same thinking skills that computer scientists use to solve complex problems!



7.8 Activity. Challenges

Imagine a situation in which you notice that many adult learners arrive late, feel rushed, or struggle to organise their daily tasks. Instead of just talking about time management, you decide to turn this real, everyday problem into a short and powerful computational thinking activity.

Your challenge is to design a 20-minute learning activity that helps adult learners to plan a realistic morning routine using only simple, unplugged materials (paper, pens, or cards). You have limited preparation time, no digital tools, and a mixed-ability group.

Set a clear goal:

- What do I really want my learners to practise through this activity?
- Is the focus on problem decomposition, sequencing, prioritisation, or conditional thinking?
- Which of these skills would make the greatest difference to their everyday lives?

You can ask learners to visualise a typical morning and list all their tasks on cards or post-its. Have them organise these tasks into a logical order, then challenge them with realistic “what if” situations (waking up late, skipping breakfast, having an early appointment). Invite them to test their routine step by step, identify what doesn’t work, and improve their plan through reflection and debugging.



8.1 Additional Resources

You can use online mind-mapping tools such as MindMeister or Coggle to help groups visualise how they break down problems and sequence their algorithm. This will make the process clearer and more structured for them.

Use examples and learning materials that include diverse representations, both in algorithms and in everyday situations. Consider problems that reflect the experiences of different ethnic and socio-economic groups.

Form heterogeneous groups that include learners with different abilities and backgrounds. This promotes peer learning, where one learner can support another in achieving their learning goals.



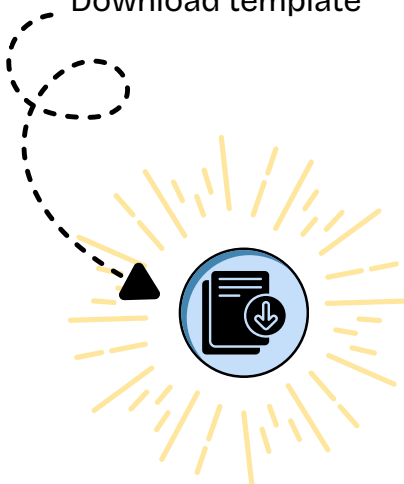
“When learners see problems from different perspectives and work together, they don’t just build algorithms; they build understanding, empathy, and skills that last a lifetime.”



8.2 Additional Resources

As an adult trainer, you have the opportunity to turn everyday information into a powerful learning moment. This fact-checking checklist offers a simple and practical way to guide learners through the complexity of disinformation using computational thinking principles. By following clear steps, learners move from instinctive reactions to structured reasoning, developing skills they can confidently apply beyond the training room.

Download template





FACT-CHECKING CHECKLIST

STEP 1 – Decomposition
Break the information into parts

- ☐ What is the main claim being made?
- ☐ Who is the author or source?
- ☐ When was it published or shared?
- ☐ What evidence is provided (data, quotes, links)?

STEP 2 – Pattern Recognition
Look for common signs of disinformation

- ☐ Emotional or sensational language
- ☐ "Shocking", "secret", or "they don't want you to know" phrases
- ☐ Lack of named sources or experts
- ☐ Strong opinions presented as facts
- ☐ Similar posts appearing on multiple questionable pages

STEP 3 – Abstraction
Focus on what really matters

- ☐ Which information is essential to judge credibility?
- ☐ What details can be ignored (images, tone, personal opinions)?
- ☐ What key question must be answered to verify this claim?

STEP 4 – Algorithmic Thinking
Follow a step-by-step verification process

- ☐ Search for the same claim on reliable sources
- ☐ Check the source's "About" page
- ☐ Verify facts using fact-checking websites
- ☐ Compare with information other example?



Final decision:

☐ Reliable
 ☐ Misleading
 ☐ False
 ☐ Not enough information



8.3 Additional Resources





9. What is the value of this stage?

The value of this unit lies in demonstrating that materials and resources are not merely supplementary teaching aids; they are essential components of effective learning design. Their selection, preparation, and presentation directly influence how adult learners understand, engage with, and apply computational thinking (CT).

This unit enables trainers to make purposeful decisions about which materials best support specific learning objectives and learner needs. It demonstrates how unplugged activities, familiar everyday objects, digital tools, or a combination of these resources can make abstract CT concepts more concrete, accessible, and relevant to real-life situations.

The unit also highlights that effective CT education does not require advanced technological expertise. Its value lies in strengthening trainers' ability to begin with manageable activities, adapt resources to different learners, remove barriers to participation, and emphasise thinking processes rather than technological complexity.

Ultimately, this unit strengthens trainers' capacity to create inclusive, practical, and engaging learning experiences. By preparing meaningful and flexible resources, trainers can help adult learners develop confidence, practise problem-solving, and transfer computational thinking skills to their everyday lives.