UNIT 4: Materials and resources preparation

Structure of the Stage

This stage consists of 8 parts:

1. Aim and Objectives;
2. Outcomes
3. Justification
4. Activities
5. Challenges
6. Practical resources
7. Additional resources
8. Conclusion

Aim and Objectives

The aim of this stage is to support you in understanding why the careful preparation and selection of resources and materials really matters when introducing computational thinking in adult education. The same activity can lead to very different learning experiences depending on how it is presented

This stage invites you to look at resources and materials not as simple supports, but as active pedagogical tools. The materials you choose can either open doors to understanding or unintentionally create barriers.

Well-prepared resources help make computational thinking more accessible, meaningful, and transferable, especially for adult learners who benefit most from clear, practical, and relatable learning experiences

Objectives:

1. To analyse how different types of learning materials (digital and non-digital) support specific computational thinking skills in adults education contexts

2. To Identify clear and practical criteria for selecting accessible, and meaningful materials for adult learners.

3. To design learning activities that develop C.T using non-digital and everyday resources familiar to adult learners.

4. To prepare and adapt resources that promote active participation, reflection, and learning transferpf C.T skills to everyday life

Outcomes

By the end of this unit, you will be able to:

- Understand why the preparation and selection resources and materials directly influences the development of computational thinking skills in adult learners.
- Understand how well-designed materials can make CT accessible, meaningful, and engaging for adult learners.
- Design unplugged and technology-based activities that make CT accessible to adult and seniors learners.
- Facilitate learning experiences that encourage participation, reflection, and the application of CT into everyday problems.
- Reflect on your role as a trainer in organising, adapting, and evaluating learning materials according to learners' need and contexts.

Did you know that...?

Sometimes a simple piece of paper or a real-life object can teach more about problem-solving than the most advanced digital tool, it all depends on how you use it.

Justification

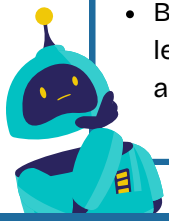
Why is this stage so important? Have you ever noticed how learning becomes easier when the right support is in place? Every teaching-learning process relies on materials that actively support the interaction between trainers and learners. In computational thinking, materials play a particularly important role, as they help make abstract thinking processes visible and tangible.

Well-design learning materials guide learners through problem-solving processes, support reflection, and help structure thinking. Familiar and everyday materials reduce cognitive load and allow learners to focus on developing CT skills rather than on mastering complex technologies

By adapting the training content to a pedagogical approach

By supporting your teaching tasks related to planning and delivering instruction

- By facilitating learning activities
- By assessing the learning outcomes achieved



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

Types of activties

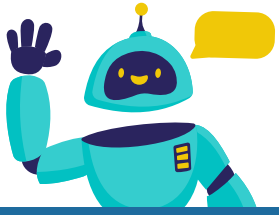
Let's look at the types of activities you can create within the framework of applying computational thinking in the classroom.

- **Unplugged activities**: these are activities that do not require the use of devices, helping you avoid possible barriers such as 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:** these involve analysing the elements of an object, such as building blocks, puzzles, simulators, or program code, and then changing or modifying them. The aim is to show how a small change can affect the way an algorithm works or is solved.

- **Making activities**: these focus on enabling learners to solve problems, plan their work, select appropriate tools, communicate ideas, and connect different concepts.

- **Remixing activities**: in these 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.

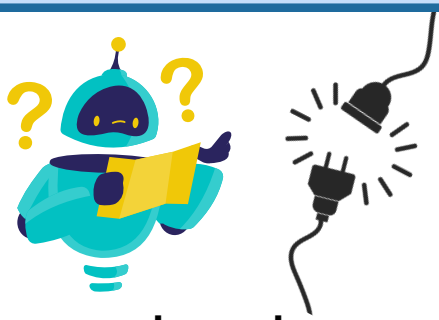


Unplugged activities

Who said you need computers to teach computational thinking? 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

Technology-based learning activities

When working with technology-based learning activities, you have access to a wide range of possibilities.

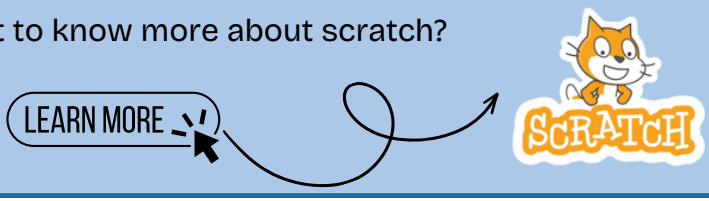
From online platforms that introduce concepts such as sequences, loops, and conditionals, to block-based, visual, and text-based programming, technology can help bring computational thinking to life in engaging and interactive ways.

One of the most well-known tools for visual programming is Scratch. Don't let its playful appearance fool you. Scratch is a powerful learning environment. It 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.

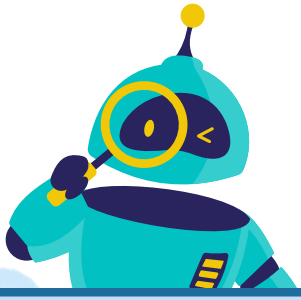
Finally, it is also worth exploring platforms linked to robotics, such as Arduino and Micro:bit, as well as others associated with robots and devices like mBlock or LEGO.

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

Do you want to know more about scratch?



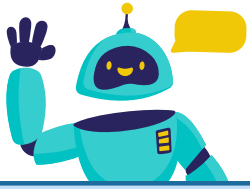
Technology-based learning activities



Did you know that...?

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.

Don't forget that when you work with computational thinking, your goal goes far beyond simply carrying out an activity, playing a game, using digital devices, or giving instructions to a robot. But then, what is the real purpose? It is about helping learners think differently and approach problems with clarity and confidence.



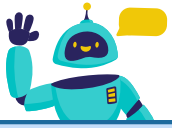
The true power of computational thinking lies in the design process. This process guides learners towards a solution and helps them reach it in a way that is efficient, effective, structured, and focused on problem-solving. It is not just about getting an answer, it is about understanding how and why that solution works.

Your aim is to help learners better understand the world around them, and feel capable of transforming it.

That is why the learning activities you create should invite participation, exploration, and reflection, turning learners into active thinkers rather than passive observers. Some of the most innovative solutions in technology and science started with simple questions and small design choices. By focusing on active learning, you give your learners the tools to ask better questions, and find better answers.

When creating learning activities, it is important to pause and ask yourself:

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 objetc 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.

You might notice that your first contact with computational thinking can feel challenging. But here's the exciting part: using reference frameworks can make the process much more manageable and give you a clear path forward.

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 computational thinking? 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 you need to learn as a trainer
- What you should teach your learners
- How to integrate computational thinking into the activities of your subject area
- How to assess learning effectively
- What type of content to select, what to prioritise, and even what to discard
- How to 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)

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!

Patterns using geometric shapes to develop logic and critical thinking

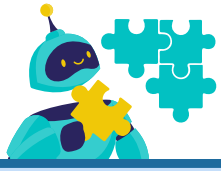
Provide your learners with a variety of geometric shapes (circles, squares, triangles, etc.) and ask them to create patterns using these shapes

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 practice pattern recognition and algorithmic thinking, which are fundamental in coding, problem-solving, and even everyday decision-making.

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.

Try that learners work together so they become familiar with group-based development, a very common practice in programming projects for designing applications or websites in professional settings.

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

Reflecting on Everyday Problems

In this activity, learners will 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 practicing the same thinking skills that computer scientists use to solve complex problems!

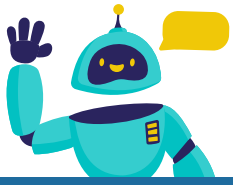
Challenges

When you prepare materials and resources for a course on computational thinking, you may soon realise that this task involves much more than choosing tools or activities

One challenge you might face as a trainer is thinking that computational thinking always requires advanced technology. In reality, unplugged activities and everyday materials can effectively support skills such as problem decomposition, pattern recognition, and stru+ctured solution design, especially in adult education contexts.

Another common difficulty is preparing resources that truly support thinking, not just doing. It's easy to get distracted by engaging games or attractive tech, but the real goal is choosing materials that guide learners to think critically and creatively. A practical solution? Start with familiar, hands-on activities and gradually adapt them to develop the desired thinking skills.

Finally, limited time and 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.



Challenges

Imagine this situation:
You notice that many of your 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 this:
Design a 20-minute learning activity that helps learners plan a realistic morning routine using only simple, unplugged materials (paper, pens, post-its, or cards). You have limited preparation time, no digital tools, and a mixed-ability group.

Set a clear thinking goal, ask yourself:

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

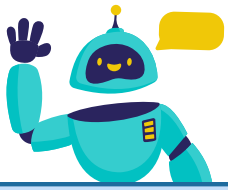
You can ask learners to visualise a typical morning and list all their tasks on cards or post-its. Have them organise these tasks into a logical order, then challenge them with realistic “what if” situations (waking up late, skipping breakfast, having an early appointment). Invite them to test their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

Practical Recourses

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.

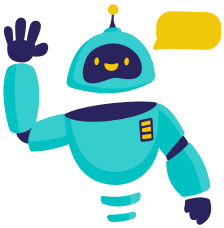
Form heterogeneous groups that include learners with different abilities and backgrounds. This promotes peer learning, where one learner can support another in achieving their learning goals.



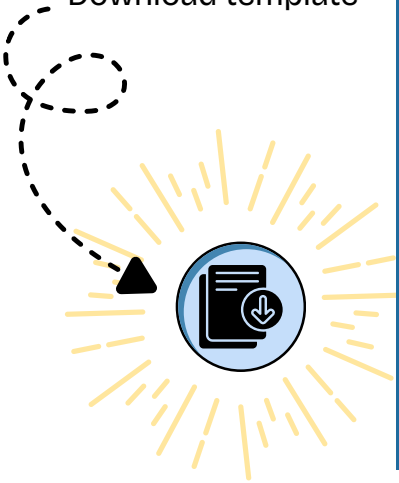
"When learners see problems from different perspectives and work together, they don't just build algorithms,they build understanding, empathy, and skills that last a lifetime."

Practical Recourses

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

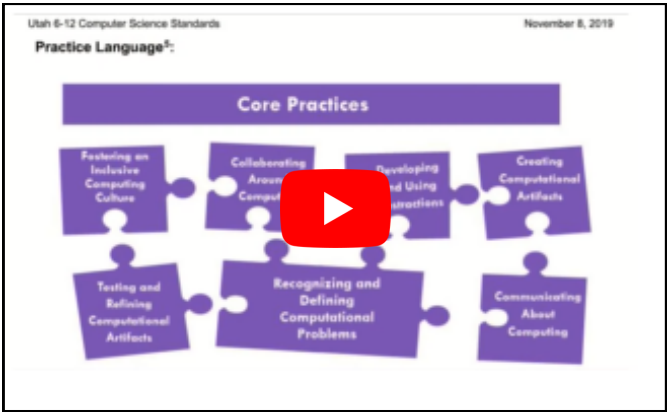
STEP 3 — Abstraction
Focus on what really matters
- ☐ Which information is essential to judge credibility?
- ☐ What details can be ignored (images, tone, personal opinions)?
- ☐ What key question must be answered to verify this claim?

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?

STEP 5 — Evaluation (Debugging)
Review and improve your conclusion
- ☐ Does the evidence support or contradict the claim?
- ☐ Did any step in the process fail or need adjustment?
- ☐ Would this checklist work for another example?

Final decision:
☐ Reliable ☐ Misleading ☐ False ☐ Not enough information

Aditional Recourses



Conclusion

As you reach the end of this unit, remember that the resources and materials you choose are not just supports for your teaching, they are powerful drivers of learning.

You have seen how carefully designed, accessible, and meaningful materials can open doors to computational thinking, helping adult learners approach problems with clarity, confidence, and creativity. Whether you use unplugged activities, everyday objects, digital tools, or a mix of all three, what truly matters is how you design experiences that invite participation, reflection, and understanding.

You do not need to be a technology expert to make a real impact. By starting with small, manageable steps, adapting activities to your learners' needs, and focusing on thinking processes rather than tools, you can transform learning into something practical, engaging, and empowering.

Your role as a trainer is central. You are not just delivering content, you are shaping mindsets, building problem-solving skills, and helping learners see challenges as opportunities. By selecting and creating materials that are inclusive, flexible, and motivating, you give your learners the tools to think differently and act confidently in the world around them.

So move forward with curiosity and confidence. Every thoughtful choice you make brings your learners one step closer to becoming active thinkers and capable problem-solvers. And that is where real learning begins.