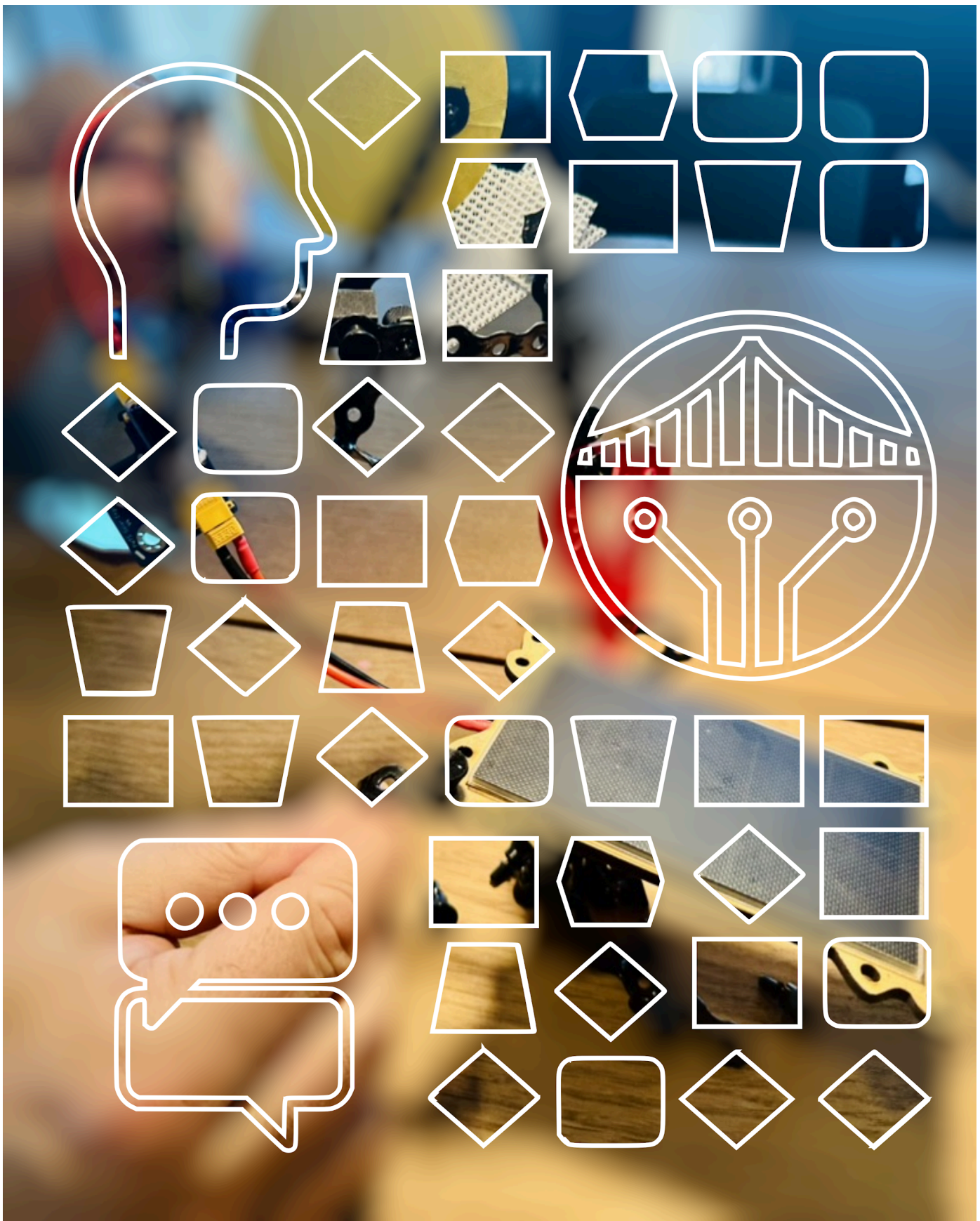
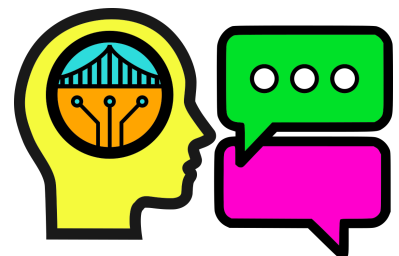




# Roadmap for using Computational Thinking in schools

Part 2



|                                |                                   |
|--------------------------------|-----------------------------------|
| <b>Date of deliverable</b>     |                                   |
| <b>Lead organisation</b>       | University of Deusto              |
| <b>Involved organizations</b>  | Tinkersjop, DOKK1, Studio Kabritu |
| <b>Work package</b>            | 2                                 |
| <b>Goal of the deliverable</b> |                                   |
| <b>Target group</b>            | Teachers and educators            |



# Summary

## PURPOSE

**Helping young people become confident digital thinkers**

This handbook supports teachers and educators in developing **Computational Thinking (CT)** in secondary education.

CT is presented as more than programming or robotics. It helps students **understand complex problems, develop solutions, think critically about technology and participate actively in a digital society.**

The handbook builds on the CT roadmap for primary education and provides concepts, examples and connections that are relevant to young people aged 12–16.

## WHAT'S INSIDE

**From concepts to reflection and classroom practice**

### CT Roadmap

A progression of Computational Thinking for secondary education, connecting CT with problem-solving, digital literacy and critical thinking.

### New CT Concepts

An introduction to emerging topics that help students understand today's digital world.

### Topics for Reflection

Six themes connect CT to everyday life and other areas of learning

### Curriculum Perspectives

An overview of how CT is addressed in secondary education in **Spain, Denmark, the Netherlands and Curaçao.**

## IMPACT

**From technology users to creators and critical digital citizens**

The handbook encourages students to:

**THINK:** Break down complex problems, recognise patterns, abstract information and develop algorithms.

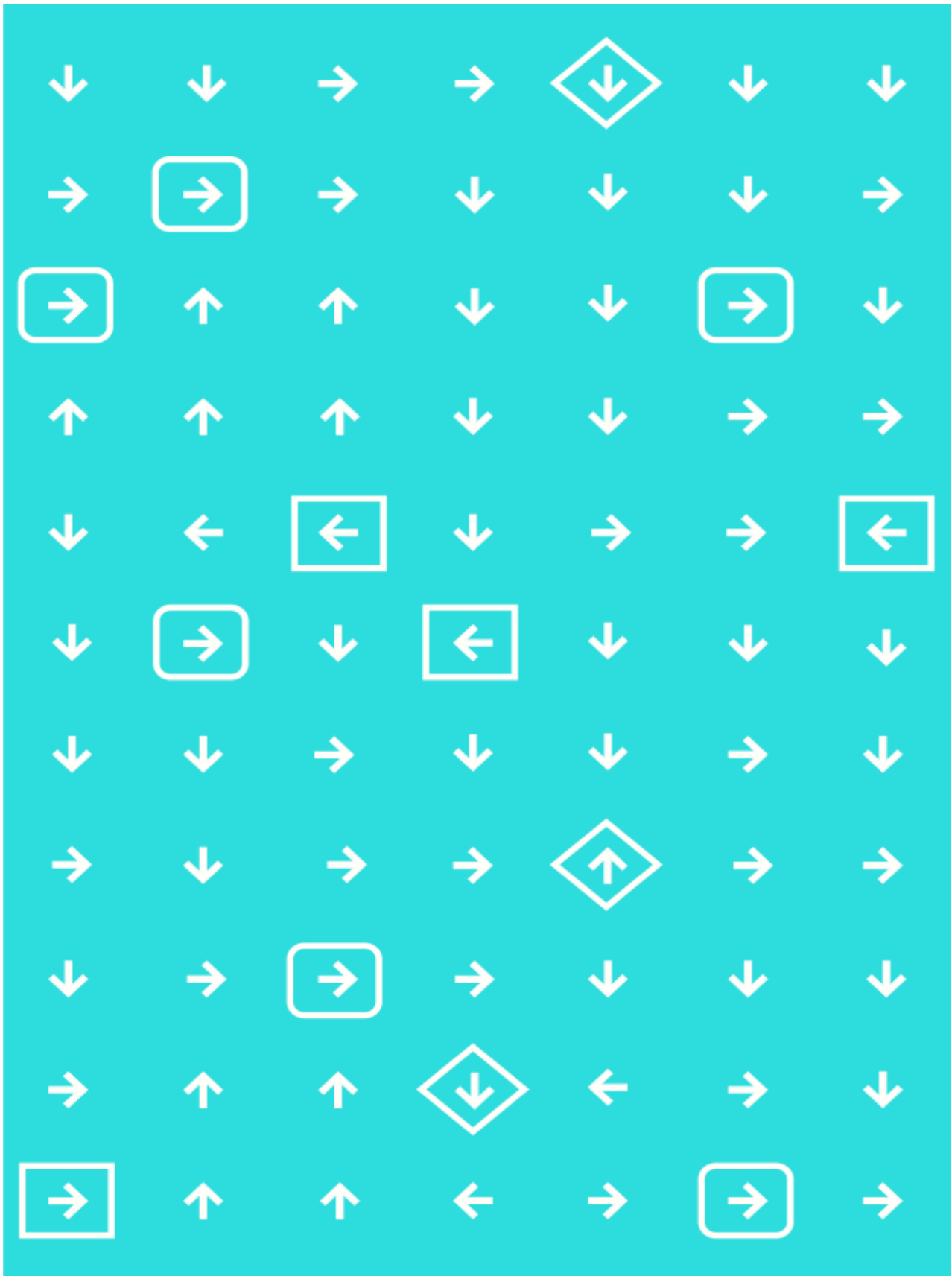
**CREATE:** Use technology to design, test and develop solutions to meaningful problems.

**CONNECT:** Apply Computational Thinking across subjects such as mathematics, science, languages, arts and social sciences.

**QUESTION:** Explore how algorithms, AI and digital technologies influence individuals, communities and society.

**REFLECT:** Evaluate solutions, recognise limitations and consider the ethical and social consequences of technology.

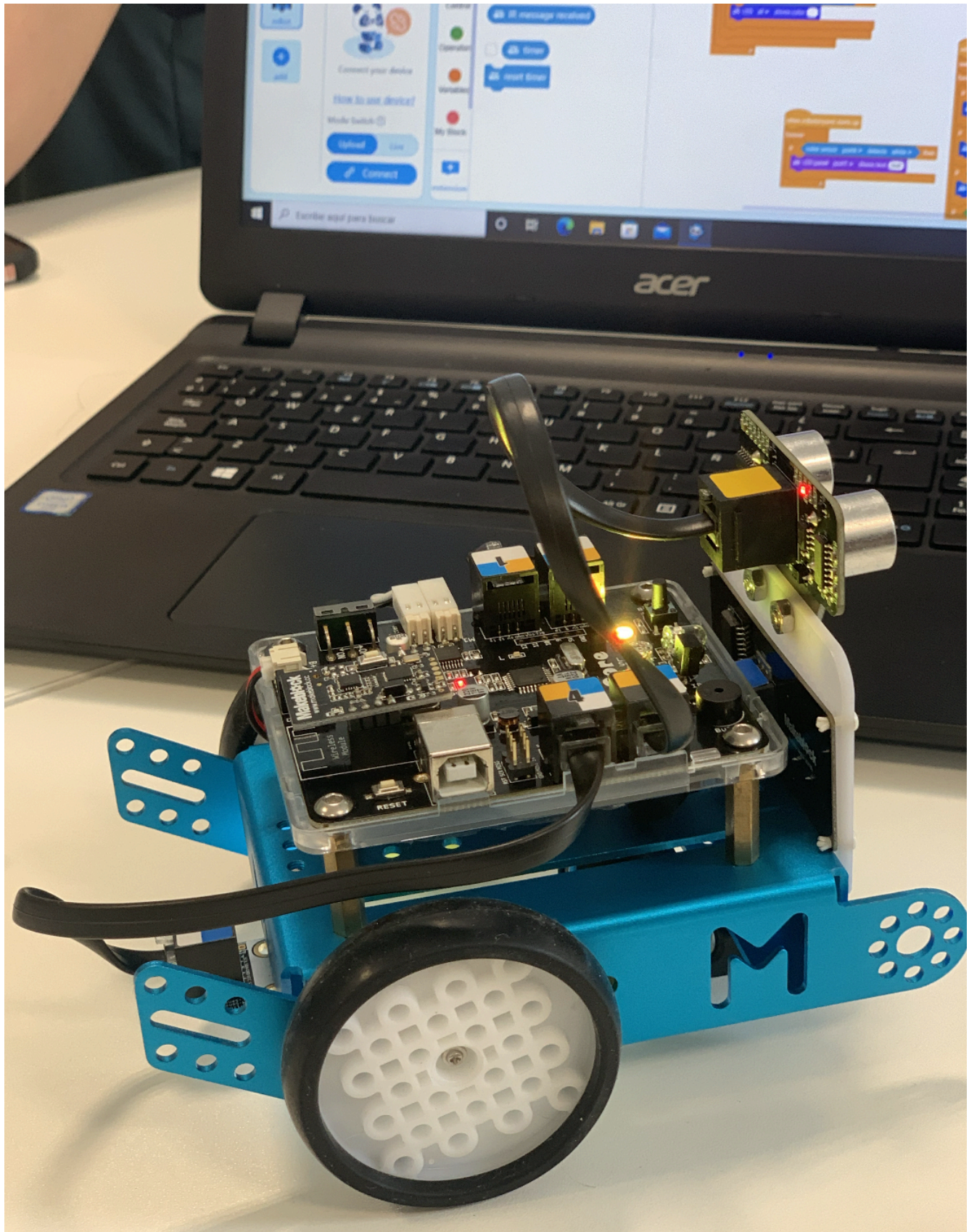




# Contents

|                                                                   |           |
|-------------------------------------------------------------------|-----------|
| <b>Summary</b>                                                    | <b>3</b>  |
| <b>Introduction</b>                                               | <b>7</b>  |
| <b>Computational Thinking Roadmap for Secondary Education</b>     | <b>8</b>  |
| <b>New terms related to CT</b>                                    | <b>10</b> |
| <b>&lt;&lt;visualisation of the 6 new topics&gt;&gt;</b>          | <b>12</b> |
| <b>Topics for reflection</b>                                      | <b>13</b> |
| 4.1 Where do we encounter CT in our daily lives?                  | 13        |
| 4.2 The purpose and impact of CT on society                       | 15        |
| 4.3 How do we talk to machines? How do they talk to us?           | 16        |
| 4.4 How can other fields (e.g. literature) help us understand CT? | 17        |
| 4.5 Connecting CT and AI                                          | 18        |
| 4.6 Connecting CT and IBL                                         | 19        |
| <b>I - Glossary of Terms</b>                                      | <b>22</b> |
| <b>II - CT in secondary schools' curricula in Spain</b>           | <b>23</b> |
| <b>II - CT in secondary schools' curricula in Denmark</b>         | <b>25</b> |
| Frameworks for the Danish Education System                        | 25        |
| <b>II - CT in secondary schools' curricula in the Netherlands</b> | <b>27</b> |
| A new curriculum since 2025                                       | 27        |
| <b>II - CT in secondary schools' curricula in Curaçao</b>         | <b>31</b> |





# 1.

## Introduction

This handbook forms the second part of the roadmap for the development of Computational Thinking and is aimed specifically at secondary school students aged between 12 and 16. Whilst the first **handbook focused on the introduction of computational thinking (CT) in primary education**), this guide introduces new concepts and terms, relates CT to other subject areas, and assesses its presence in various secondary school curricula.

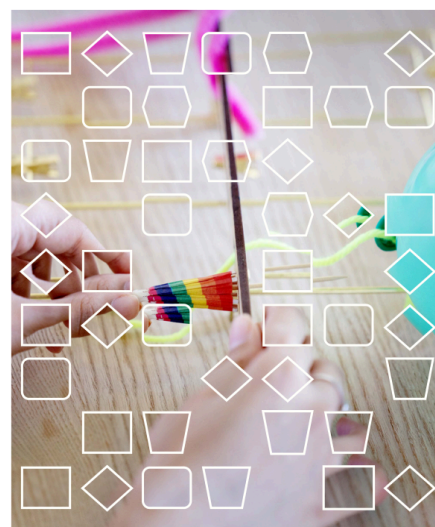
Secondary education represents a particularly important stage for the development of computational thinking. At this age, students begin to tackle more complex problems, engage critically with technology and develop their own identity as digital citizens. They no longer merely use technology, but increasingly become creators, communicators and decision-makers in digital environments. For this reason, CT should not only be linked to

programming or robotics, but also to ethics, communication, creativity, problem-solving and critical thinking.

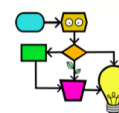
This handbook aims to help teachers connect computing with the realities and interests of teenagers, explore how computing manifests itself in everyday life, how it influences society, and how it relates to other disciplines, such as literature, science and the arts. It also introduces more recent concepts related to artificial intelligence, digital communication and inquiry-based learning.

Similar to the handbook for primary schools, the aim is not to turn all teachers into IT experts, but to provide tools and ideas that can enrich learning experiences across different subjects. Computational thinking can be integrated into maths, languages, science, the arts or social sciences.

And, more importantly, it can help our students develop a structured and reflective way of thinking that prepares them for an increasingly digital and interconnected world.



**Roadmap for using Computational Thinking in schools**



## 2.

# Computational Thinking Roadmap for Secondary Education

The development of CT for pupils aged 12 to 16 should build on the foundations laid during primary education. At this stage, more abstract concepts, more open-ended projects and more complex forms of collaboration can be explored. Pupils can also begin to critically analyse the impact of technology on society and understand how digital systems influence their daily lives.

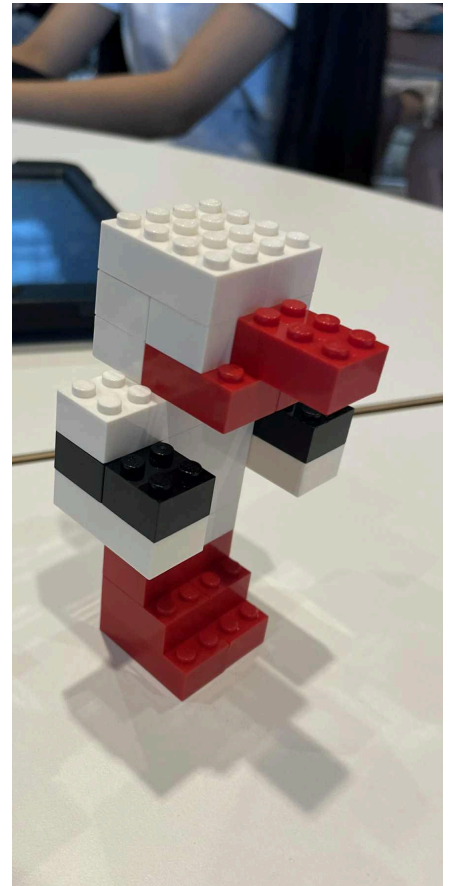
The roadmap for secondary education should continue to develop the core foundations of CT (decomposition, pattern recognition, abstraction and algorithms), but these can now be explored through more sophisticated challenges and interdisciplinary projects involving simulations, data analysis, artificial intelligence, robotics, app development or digital storytelling,

whilst discussing ethical and social issues related to technology.

The progression of computational thinking throughout secondary education can be understood across three interconnected levels.

**The first level** involves reinforcing the basic concepts of CT. Our students must gain confidence in designing algorithms, debugging, analysing systems and organising information. They must also learn to evaluate the different possible solutions to a problem and understand that there may be multiple valid approaches.

**The second level** focuses on technological expression. Programming languages, robotics kits, sensors, online platforms and



digital creation tools become instruments for exploration and creativity. However, technology in itself is not the ultimate goal. The aim is to use technological tools to solve meaningful problems, communicate ideas and create projects related to real-world contexts.

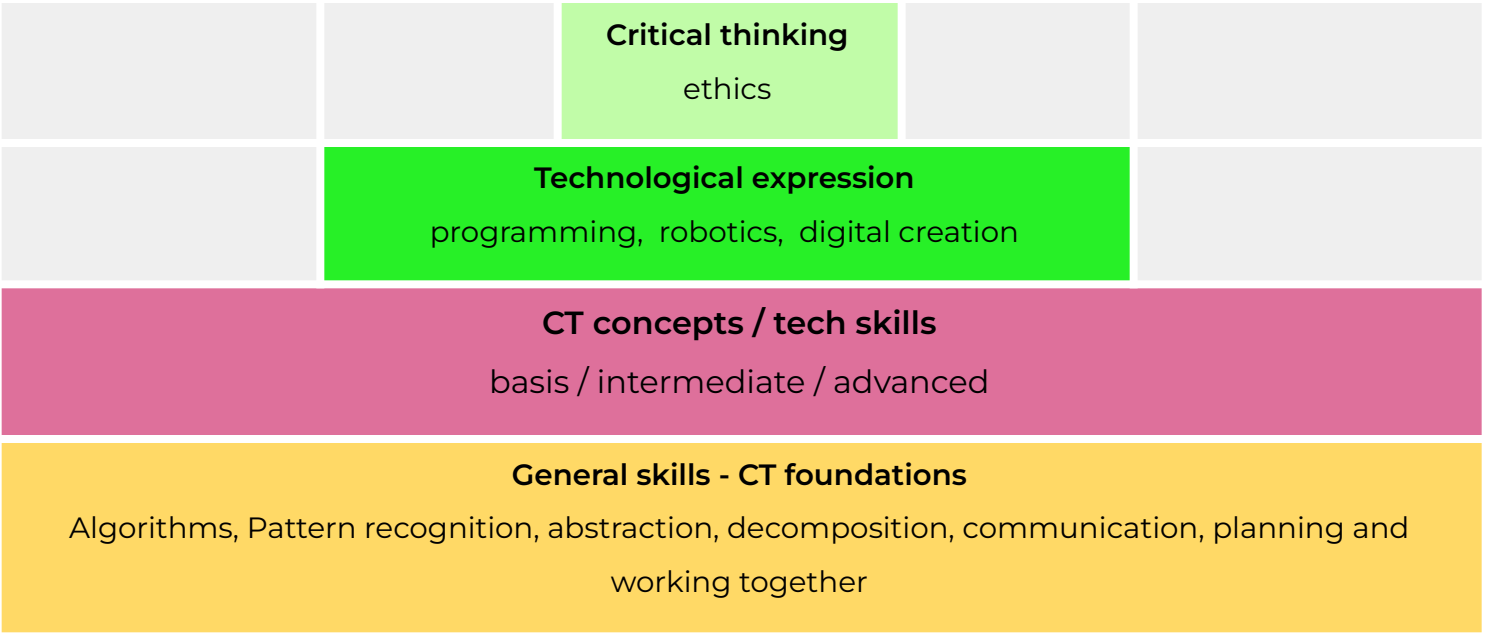
**The third level** involves critical and reflective thinking. Adolescents increasingly interact with recommendation algorithms, social media platforms and AI systems. Therefore, CP education at this stage must include discussions on digital identity, privacy, misinformation,

ethics, sustainability and the social consequences of technology.

Projects for this age group should ideally be interdisciplinary and collaborative. For example, each group of students could design a sustainable city by analysing real-world data to create simulations of how climate change will affect it, create interactive narratives linked to literature lessons, or investigate how algorithms influence society's perception of these issues through social media. These activities help our students realise that CT is not limited to programming in computer

labs, but is a way of tackling problems and organising ideas across many disciplines.

As in primary education, the integration of computational thinking must remain inclusive and accessible. Offline activities, discussions, games and creative projects continue to play an important role. By fostering curiosity, experimentation and hands-on learning, we can progressively increase the complexity and autonomy of our students.



### 3.

## New terms related to CT



Having covered the basics of CT in primary school, our pupils are now ready to explore both the technical aspects and the ethical and social implications of using technology in greater depth.

#### Internet of Things (IoT):

The internet is increasingly less a network interconnecting computers and more a cloud of highly diverse, interconnected devices. The Internet of Things can be compared

with Theme park visitors that wear a smart wristband that provides additional information, such as which rides they have been on, how long they have been waiting for a ride, or the maximum speed they have reached. The wristband could also be used to locate a family member or friend who has become lost. All information together can be seen as the Internet of things.

Our students' daily lives are shaped by smartwatches, home automation systems, smart cars, etc., and they are often unaware that it is computing that makes them 'smart'. This also raises interesting questions regarding design, privacy and usability that can be addressed through projects or discussions.

### **Data science / Big data**

The digital devices and platforms we use every day collect and process huge amounts of information daily.

In the analogy of the themepark, big data is the analysis of the location data of all the visitors in the theme park. It is possible to model and predict how long people will have to wait for each ride.

Proper digital literacy requires the ability to interpret, analyse and question data. Our students must learn that data is not neutral: it is collected, organised and interpreted by people and systems that may contain biases or limitations.

### **Artificial intelligence (AI):**

This has been the buzzword in recent years, but we often struggle to define its boundaries. Understanding the different techniques that fall under the umbrella of AI can help our students see past the aura of 'magic' that sometimes surrounds these technologies. AI would be the agent that plans a suitable order of rides based on your preferences and the current status of each attraction. It could also schedule breaks for drinking water, resting or eating, and time your route so that it coincides with shows that might interest you. This enables them to

make informed decisions about when it is appropriate to rely on AI and when the associated risk is too high.

### **Algorithmic bias:**

Not only can the collection, processing and analysis of data be biased, but so too can the programmes that make decisions about important aspects of our lives (access to housing, work, credit, healthcare, etc.). When applications are developed by teams with little diversity, they are likely to reproduce or amplify stereotypes and inequalities. Or in the analogy of the themepark If the Big Data or AI models described do not have enough data, or if their data is not sufficiently diverse, they may reach incorrect conclusions. For example, they might assume that people choose rides only because they are nearby, when in fact visitors often choose them according to their speed or perceived level of risk. Understanding which decisions are being delegated to algorithms helps us discuss the ethical dimension of technology.

### **Cybersecurity:**

As users of social media, multiplayer video games, streaming platforms and other digital services, it is vital that our students know how to protect their accounts with strong passwords, recognise

phishing and other online scams, and safeguard the privacy of their personal information. The development of the PC in secondary education allows us to go beyond a vision centred on the responsible use of these platforms and to question design decisions that have implications for cybersecurity. And in the themepark, without strong cybersecurity measures in the theme park, someone might be able to open another visitor's locker by using the wrong smart wristband or by exploiting a weak password.

### **Digital citizenship:**

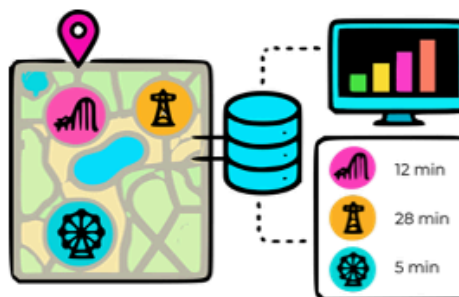
Secondary school students are increasingly participating in online communities and digital environments. Digital citizenship involves responsible participation online, which includes respectful communication, awareness of privacy, critical evaluation of the information received, and an understanding of the implications of one's own digital footprint. If some visitors of the theme park refused to respect their place in the queues or reserved several lockers that they did not intend to use, many people in the park would be unhappy. This illustrates how a lack of responsible behaviour and respect for shared rules can negatively affect the whole community.



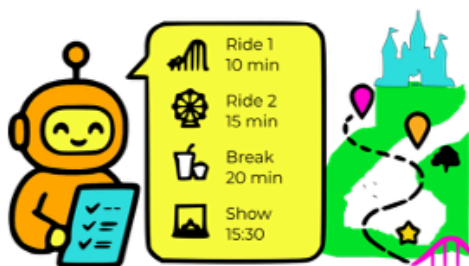
## Internet of things



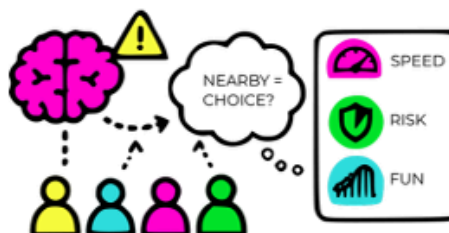
## Big Data / Data Science



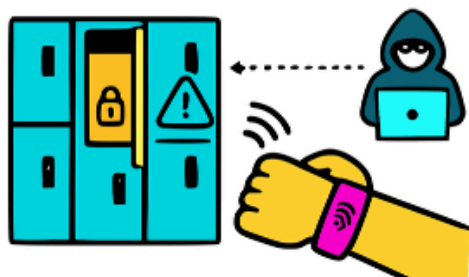
## Artificial Intelligence



## Algorithmic bias



## Cybersecurity



## Digital citizenship



# 4.

## Topics for reflection

### Where do we encounter CT in our daily lives?

Computational Thinking is commonly associated with the use of computers and other technologies. However, CT is present in our daily lives in many different contexts. It enables us to apply a range of mental processes to solve problems in the world around us.

The first context in which we usually identify the use of CT is our interaction with technological devices such as smartphones, tablets and computers. These devices carry out millions of operations and logical decisions every time we send a message, take a photograph or



perform an online search. Similarly, CT can be identified in the algorithms used by the designers of the online services we access every day. These algorithms analyse our activity to predict and prioritise the content most likely to attract our attention.

A much more universal context is following algorithms that involve routines, processes or sequences. Designing an algorithm involves creating a logical and ordered sequence of steps to achieve an objective or solve a particular problem. Examples include following a recipe step by step, carrying out a set of instructions to assemble a piece of furniture, or planning the fastest route to a destination. If we change the order of the steps in any of these situations, the outcome may change dramatically. For example, if we put on our shoes before our socks, the 'program' fails because of a sequencing error.

Similarly, some situations involve so much information and so many stimuli that we need to use abstraction to ignore irrelevant details and focus only on essential features. When using a map, for example, we are dealing with an abstract representation of reality. We must reduce the complexity of its different layers of information to concentrate exclusively on the details we need to reach our destination.

CT is also present in organisation and planning. People generally seek to save time and energy, so we use

mental processes such as decomposition and algorithmic thinking to organise tasks and make plans. Organising a birthday party, for example, requires us to break down a broad, long-term objective into specific tasks, such as creating a guest list, buying food, preparing a music playlist and designing the decorations. It also involves defining an ordered and chronological sequence of actions: in other words, a project-management algorithm that

enables us to achieve the initial objective.

In this same search for efficiency, both humans and machines structure information to make searches faster and more precise. When packing a suitcase, for example, we might classify objects by type of clothing or by the date on which we expect to use them, making it easier to find what we need. These processes are direct analogue applications of



computational concepts. In the same example, we might place our toiletry bag in an easily accessible part of the suitcase because we are likely to need it quickly or repeatedly, such as during a security check. Digital devices use a similar principle when they store frequently accessed files in cache memory so that they can be retrieved more quickly.

Another everyday example of CT occurs when we have lost something. If we cannot find our house keys, we often try to reconstruct our previous actions systematically until we identify the last moment at which we had them. Machines use a similar process known as backtracking. This is a basic algorithmic method for exploring decision trees and finding solutions by returning to an earlier point whenever a dead end is reached.

Returning to the use of technology, CT is also present in everyday contexts such as the home and transport. This is due to systems containing microcontrollers and sensors that measure variables such as time, temperature, light levels and the presence of objects. Through the Internet of Things (IoT), CT finds another of its key areas of application, transforming our surroundings into a fully interconnected ecosystem. This network not only allows objects to communicate with one another, but also optimises the automatic and efficient resolution of everyday problems.

As we can see, CT is not an approach used exclusively by machines or programmers. It is a way of addressing and solving problems in our daily lives through logical and efficient processes.

## **The purpose and impact of CT on society**

As our society is increasingly shaped by algorithms, CT involves more than just a set of technical skills; it enables us to examine the social dimension of technology in greater depth. We communicate, learn, work and make decisions based on data and programmes. Secondary school students should be able to reflect on and critically analyse both the benefits and the challenges associated with this technological society.

One of the main objectives of CT is problem-solving. Computational thinking helps us to analyse complex situations, organise information and design step-by-step solutions. These skills are valuable in fields such as medicine, engineering, science, education, the arts and environmental sustainability. Many of the global challenges facing society today, such as climate change, public health or transport systems, could benefit from solutions that harness large amounts of information to reach sound conclusions.

Developing CT also allows us to explore new forms of innovation and creativity. Being able to create apps, games, simulations, works of art or music empowers today's young people to move beyond being mere consumers of technology.

On the other hand, algorithms influence the news we read, hear or watch, the products we buy, and how we interact socially online. Whilst these technologies can improve immediacy, accessibility and access to information, they can also create new inequalities, opinion bubbles, misinformation and other ethical dilemmas.

Artificial intelligence systems are increasingly used in healthcare, education, employment and security. Whilst these systems allow us to detect and treat more diseases, learn in new ways, work in multilingual environments or detect potential threats, automation is also transforming labour markets and changing the nature of many professions. These issues demonstrate that technology is not neutral: technological systems reflect the human decisions, priorities and values of those who created them.

Therefore, our students must learn to question technology critically. Who designs these systems? What data is collected? Who benefits from the technology? Who might be excluded or negatively affected? These questions are essential for shaping responsible digital citizens.



Schools play an important role in this process. ICT education should not focus exclusively on technical competence, but also on ethics, participation and responsibility. Students must feel empowered not only to use technology, but also to understand it, evaluate it and adapt it to their needs. In other words, to be active participants in society, capable of using technology in a reflective, creative and responsible manner.

## How do we talk with machines

Communication between people and machines has not always been easy. Initially, it was necessary to know every detail of the machine in order to make the most of its functions, but little by little we have been creating ways that are increasingly similar to how humans communicate. Thus, at first we had to know how to encode information in binary and understand the instructions each machine was capable of carrying out, whereas now it is possible to speak to a machine using our mother tongue and, in many cases, hold a coherent conversation that allows us to achieve our objective.

Understanding how all these ways of communicating between people and machines work is an important part of computer science.

Computers do not understand language in the same way as people. Machines process information

through instructions, patterns and data. When we write a programme, create a search query or use a voice command, we are translating our intentions into forms that machines can interpret. This process requires precision, structure and logical thinking.

Programming languages are an example of this type of communication between humans and machines. They allow humans to express instructions in a structured way so that computers can execute them. Some languages are text-based, whilst others use visual blocks or graphical interfaces. Regardless of the format, programming involves organising ideas clearly and anticipating how the machine will respond.

At the same time, machines are increasingly communicating with humans using images, sounds, text and interactive systems. Artificial intelligence systems can generate conversations, recommendations or translations that seem natural and similar to those of humans. However, our students must understand that machines do not truly 'understand' language or emotions in the same way as people do. Rather than feeling something akin to what we feel when we communicate, machines analyse patterns in large amounts of data and predict probable responses that make sense in that context.

This difference in the way we communicate is closely linked to

digital literacy and critical thinking. Social media and other digital platforms are designed to influence our attention and behaviour through notifications, recommendations and personalised content. Understanding how machines 'speak to us' involves recognising that interfaces and algorithms are designed with intentions that may not be our own.

Our students can explore these ideas through practical activities. For example, they could compare human instructions with programming instructions, analyse chatbot responses, or investigate how search engines prioritise certain results. These activities help pupils understand the importance of clarity, ambiguity and interpretation in communication.

Another important point for analysis concerns accessibility and inclusion through technology. Communication between humans and machines must take into account different languages, cultures and abilities. Voice assistants, subtitles and assistive technologies demonstrate how technology can facilitate communication for users with functional diversity.

## How can other fields help us understand CT?

CT is often associated with mathematics, programming or engineering. However, literature, art,



history, music and philosophy offer opportunities to explore patterns, structures, systems and interpretations.

Literature has particularly interesting connections with CT. Stories are constructed using structures, sequences and relationships between characters and events. Students can analyse narrative patterns, identify recurring themes or explore how stories branch out into different possibilities. Interactive fiction and 'choose your own adventure' stories resemble algorithms with conditional decisions and paths.

Abstraction also frequently appears in literature. In many novels, certain details are shown whilst others are omitted to create meaning and guide the narrative. Similarly, computer science uses abstraction to focus on relevant information whilst simplifying complexity. Comparing these processes can help students understand abstraction in a more intuitive way.

Pattern recognition also plays a significant role in literature. Poems often contain repeated structures, rhymes, rhythms and themes. The historical events on which they are based can also reveal recurring social or political patterns.

Both computing and literature encourage creativity and experimentation. Creating digital narratives is very similar to creating video games; in both cases, artistic

expression is combined with structure and patterns.

Creating interdisciplinary or transmedia projects can make computer science more appealing to students with non-technical interests.

Linking CT to different fields helps to avoid a narrow view of technology education. It shows students that computer science is fundamentally about organising ideas, solving problems and understanding systems. These skills are valuable in many disciplines and professions.

## Connecting CT and AI

Although Computational Thinking and Artificial Intelligence are closely related, it is important to emphasise that they are not equivalent concepts. CT is a way of analysing problems, organising information and designing solutions that can be carried out by either a person or a machine. AI, by contrast, encompasses a range of techniques that enable computer systems to perform complex tasks such as recognising images, classifying information, generating text, making predictions and recommending content.

Developing a deeper understanding of CT can help dispel the idea that AI is a technology that works by magic or operates entirely autonomously. Behind every AI system are human decisions concerning how the

problem is defined, which data is selected, how the model is designed, which evaluation criteria are used and how the results are presented. CT provides tools for identifying and analysing each of these elements.

The core foundations of CT are present throughout the different stages involved in creating and using an AI system. Decomposition makes it possible to divide a complex problem into simpler tasks: collecting data, organising it, training a model, evaluating its results and deciding how it will be used. Pattern recognition is essential because it allows many AI systems to identify regularities within large amounts of information. Abstraction helps us select the relevant features of the data whilst ignoring details that are not useful for solving the problem. Finally, algorithmic thinking enables us to establish the procedures through which data is processed, the system is trained and responses or predictions are generated.

However, not all computer systems that make decisions use AI. A traditional program follows rules defined in advance by its designers. For example, we can use a very simple program without AI to grant access to an application using a password. By contrast, if we want to access the same application through facial recognition, variations in lighting, angle, focus or even our own appearance make the problem considerably more difficult and are likely to require the use of AI. Many AI systems learn statistical



relationships from examples and produce probability-based results. A spam filter, for instance, can learn to classify messages after analysing numerous examples of legitimate and fraudulent emails. Comparing these two approaches allows students to explore the difference between an explicit instruction that produces an exact response and a model trained on data that produces a probabilistic one.

The relationship between CT and AI can also be observed when using generative AI chatbots. When students write a prompt requesting a text, an image or an explanation, they need to define their objective clearly, provide the necessary context, establish constraints and evaluate the result. These actions involve decomposition, abstraction, sequencing and debugging. When the response is not appropriate, students can revise their request, identify ambiguities, add information or try a different strategy. In other words, they debug their solution through an iterative process of design and improvement.

Nevertheless, it is important to remember that using an AI tool does not necessarily mean that students are developing CT. If they delegate the formulation of the problem, decision-making and evaluation of the response to the system, they may have fewer opportunities to reason independently. AI should therefore be used to support the exploration of alternatives, the generation of examples, the comparison of

explanations and the review of processes, rather than as a substitute for critical thinking. Its educational value does not lie solely in obtaining an answer, but in understanding how the problem was formulated, which steps were followed and how the quality of the result can be assessed.

CT also enables students to analyse the limitations of AI systems. Their results depend on the data used, the variables selected and the objectives defined by their creators. If the data contains errors, inadequately represents certain groups or reproduces social inequalities, the system may generate biased outcomes. Generative models may also provide inaccurate or fabricated information in a convincing form. For this reason, our students must learn to verify responses, compare sources, identify potential biases and recognise when human intervention is necessary.

These issues can be explored through practical activities adapted to secondary education. Students can build a simple classifier from examples, compare a rule-based solution with a machine-learning approach, analyse the data that would be required to train a system, or investigate the errors made by an AI tool. They can also examine how recommendation systems work, evaluate automatically generated responses, or discuss which decisions should not be delegated to a machine.

Addressing CT and AI together enables our students to understand these technologies from two complementary perspectives. On the one hand, they acquire tools for designing, experimenting and solving problems using digital systems. On the other hand, they develop criteria for questioning how these systems work, whose interests they represent and what consequences they may have. In this way, students stop perceiving AI as a black box and become capable of using it critically, creatively, ethically and responsibly.

## Connecting CT and IBL

Both Computational Thinking and Inquiry-Based Learning are approaches centred on solving real-world problems through critical thinking and exploration.

“Inquiry is the process in which students engage in the investigation of scientifically oriented questions, actively conduct experiments, formulate explanations based on evidence, evaluate their own explanations and alternative explanations, and communicate and justify the explanations they propose” (National Research Council, 2000).

Inquiry-Based Learning guides students in formulating questions and hypotheses to understand the world around them, as well as in designing experiments to test their hypotheses using evidence that enables them to

approach knowledge as objectively as possible. Computational Thinking, meanwhile, provides tools for processing information from the surrounding environment, helps to organise inquiry and research processes in a more logical, planned and reflective way, and supports the design of solutions.

The core processes of Computational Thinking are clearly integrated into the different phases of the IBL methodology.

**Ask and Engage phase:** The aim of this initial phase of IBL is to capture students' interest and encourage them to formulate questions that will help them better understand the problem or subject under investigation. This phase is related to decomposition, as students begin with a research question and identify simpler, more manageable questions.

**Investigate and Explore phase:** During this phase, students design an experiment, test their hypotheses and directly observe how a phenomenon behaves. This phase is related to algorithmic thinking, as students design a step-by-step plan for carrying out the investigation. It is also linked to pattern recognition, as

students look for trends, similarities or anomalies in the data collected during the investigation.

**Create and Explain phase:** During this phase, students construct descriptive or explanatory models and discuss the data. This phase is related to both algorithmic thinking and abstraction. On the one hand, students design a step-by-step plan for creating a solution or model that helps explain how the phenomenon under study works. On the other, they filter out irrelevant information and unnecessary details in order to identify the essential data needed to explain their model.

**Reflect and Evaluate phase:** During this phase, students analyse and evaluate their work. It is related to debugging, as students assess their results, identify flaws in their reasoning or experimental procedure, and correct them.

Several studies show that applying IBL is effective in improving secondary school students' Computational Thinking skills. They also indicate that developing Computational Thinking skills improves students' application of the

scientific method from both analytical and procedural perspectives.

The two approaches are complementary, essential to the development of secondary school students' competences, and oriented towards problem-solving. Through inquiry, students can investigate a local issue, such as the impact of air pollution in their area, whilst applying the four core foundations of Computational Thinking.

First, through decomposition, students simplify the complexity of the problem by dividing it into subtasks, such as identifying the main sources of emissions and planning the collection of data. They then apply pattern recognition by analysing the data collected to identify trends, such as peaks in pollution at particular times of day. Next, through abstraction, they discard irrelevant information in order to focus on the critical variables. Finally, they apply algorithmic thinking by designing an explanatory or predictive model based on decision trees and logical instructions that can help solve or mitigate the problem.



# Annex

## I - Glossary of Terms

### **Abstraction**

Focusing on the essential information of a problem while leaving out unnecessary details.

### **Algorithm**

A sequence of instructions or steps used to solve a problem or complete a task.

### **Algorithmic Bias**

Bias in the data, design or rules used by an algorithm that can lead to unfair or inaccurate outcomes.

### **Artificial Intelligence (AI)**

Technologies that enable computers or machines to perform tasks that are commonly associated with human intelligence.

### **Big Data**

Very large and complex amounts of data that can be collected, processed and analysed to identify patterns and generate insights.

### **Computational Thinking (CT)**

A structured way of approaching problems by using strategies such as decomposition, pattern recognition, abstraction and algorithmic thinking.

### **Cybersecurity**

The protection of digital systems, networks, accounts and data against unauthorised access, misuse or attacks.

### **Data Science**

The process of collecting, analysing and interpreting data to identify patterns, answer questions and support decisions.

### **Decomposition**

Breaking a complex problem or task into smaller, more manageable parts.

### **Digital Citizenship**

The ability to participate in digital environments in a responsible, critical, safe and constructive way.



**Digital Literacy**

The knowledge and skills needed to use, understand and critically engage with digital technologies and information.

**Debugging**

Identifying, analysing and correcting errors or problems in a program, process or solution.

**Inquiry-Based Learning (IBL)**

An approach to learning in which students investigate questions, gather information, develop explanations and reflect on their findings.

**Internet of Things (IoT)**

A network of physical devices and objects that are connected and can collect, exchange or process data.

**Pattern Recognition**

Identifying similarities, trends, regularities or anomalies within information or data.

**Programming**

Creating instructions for a computer or digital system to perform a particular task.

**Robotics**

The design and use of programmable machines or robots that can perform physical or digital tasks.

**Digital Identity**

The information, behaviour and representation associated with a person in digital environments.

**Digital Communication**

The exchange of information through digital technologies and platforms.

**Digital Ethics**

The consideration of ethical questions and consequences related to the development and use of digital technologies.

**Misinformation**

False or inaccurate information that is shared or presented as factual, regardless of whether there was an intention to deceive.

**Abstraction & Modelling**

Representing a real-world system or problem in a simplified form in order to understand, analyse or work with it.



# Annex

## II - CT in secondary schools' curricula in Spain

The Spanish Education Act, known as the LOMLOE (Organic Law 3/2020), defines the main principles and structure of the Spanish education curriculum. It integrates Computational Thinking transversally, including it explicitly both in the definition of the key competences that make up the student exit profile and in the knowledge and assessment criteria of several subjects within Compulsory Secondary Education, known in Spain as Educación Secundaria Obligatoria (ESO).

Firstly, Digital Competence is defined as one of the key competences that students must acquire by the end of compulsory education. Within this competence, Computational and Critical Thinking are explicitly included as essential elements, together with data literacy, programming and problem-solving.

The operational descriptors of this competence that refer most directly to Computational Thinking are as follows:

### **CD 2 - Content creation**

Creating and editing content in different formats, including through the use of programming languages.

### **CD 5 - Problem-solving and Computational Thinking**

Students should develop simple computer applications and creative technological solutions. They are also expected to understand the general principles and underlying logic involved in the evolution of digital technologies.

Computational Thinking is also linked to Competence in Mathematics, Science, Technology and Engineering (STEM). This relationship is defined more specifically through the following descriptors:

### **STEM 1**

The use of inductive, deductive and logical methods to select and apply problem-solving strategies.

### **STEM 4**

The interpretation and communication of information using specific languages, symbols and codes, whilst drawing on digital culture

The following overview shows how Computational Thinking is addressed through different secondary school subjects. This curricular approach also encourages the use of active methodologies such as project-based learning and learning by doing. In this way, students are encouraged to become not only users of technology, but also creators who use CT to produce outcomes with eco-social value and develop sustainable solutions. This approach makes CT a fundamental problem-solving tool that can be applied in mathematics, science and everyday life.



| Topic                                                | Description                                                                                                                                                                                                                                                                              |
|------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>BIOLOGY AND GEOLOGY</b>                           | Using reasoning and/or Computational Thinking to solve problems related to biology and geology, or to explain biological and geological processes occurring in everyday life. This involves critically analysing responses and solutions and reformulating the procedure when necessary. |
| <b>MATHEMATICS</b>                                   | Using the principles of Computational Thinking by organising data, breaking problems down into parts, recognising and interpreting patterns, and modifying, generalising and creating algorithms to model situations and solve problems effectively.                                     |
| <b>NATURAL SCIENCES</b>                              | Understanding and relating the causes of the main natural phenomena and processes by using scientific reasoning, scientific laws and theories and/or Computational Thinking to solve problems or explain processes found in everyday life.                                               |
| <b>TECHNOLOGY AND DIGITALISATION</b>                 | Developing algorithms and computer applications in different environments, applying the principles of Computational Thinking and incorporating new technologies to generate solutions to specific problems, automate processes and apply them within control systems or robotics.        |
| <b>DIGITALISATION</b>                                | Using the principles of Computational Thinking to design and develop algorithms and computer applications that solve specific situations and problems effectively, creatively and innovatively through the use of a simple programming language.                                         |
| <b>TECHNOLOGY</b>                                    | Developing automated solutions to the problems presented, applying the necessary knowledge and incorporating emerging technologies to design and build programmable or robotic control systems.                                                                                          |
| <b>INTRODUCTION TO TECHNOLOGY AND DIGITALISATION</b> | Developing simple algorithms using block-based programming languages and applying the principles of Computational Thinking to generate solutions to straightforward problems.                                                                                                            |



# Annex

## II - CT in secondary schools' curricula in Denmark

### Frameworks for the Danish Education System

The Danish education system is known for its flexibility and local autonomy. While there are national goals and competence descriptions, schools and teachers have significant freedom to plan teaching and choose methods, materials, and ways of working across subjects.

In the upper secondary level of compulsory school (approx. ages 12–16), emphasis is placed on student involvement, collaboration, and project-based learning. Teaching is often organized around topics and themes, allowing students to work on real-world problems involving multiple subjects.

In this context, computational thinking (CT) is not a standalone subject but is integrated into all student activities—addressing

problems, digital culture, and critical thinking.

#### ■ Computational Thinking in Subjects and Teaching

Although CT is not explicitly described in the curricula, many of its core elements are part of the teaching:

**Mathematics** Students work with problem-solving, pattern recognition, and logical thinking. These competencies are closely linked to central CT concepts such as decomposition, abstraction, and algorithmic thinking.

**Science (Physics, Chemistry, Biology, Geography)** Students use models, data, experiments, and simulation programs. This involves analyzing complex systems, finding patterns in data, and working with data-based explanations.

**Danish and Humanities** Students work with structure, communication, and interpretation,

often on digital platforms. Here, they develop the ability to organize information, critically analyze content, and communicate complex relationships clearly.

Across subjects, open-ended problems and project work are common, where students help define problems and develop solutions. This approach provides excellent opportunities to practice computational thinking in practice, even if the term itself is not always used directly.

#### ■ Digital Literacy and Technology Comprehension

There is a strong focus on digital literacy (digital dannelse) in Danish compulsory schools. This covers both the use of digital tools and a critical understanding of technology's impact on the individual and society.

In recent years, there has been increased focus on technology Comprehension as a concrete subject area. This includes



programming, data literacy, digital design, and reflection on technology. The goal is for students not only to use technology but also to understand how digital systems function and what consequences they involve.

This development makes the connection between digital literacy and computational thinking clearer, even though it is not yet fully integrated into the curricula.

### ■ Current Practice and Challenges

In practice, many schools already work with elements of computational thinking through activities like programming, robotics, data processing, and digital projects. These activities are often integrated into existing subjects or included in interdisciplinary courses.

A strength of the Danish approach is the focus on project-based and inquiry-based learning, where students collaborate on real-world

problems. This offers great opportunities to develop both CT competencies and broader skills like creativity, collaboration, and critical thinking.

However, the lack of a unified national framework for CT means implementation varies from school to school. Teachers' competencies, access to resources, and local priorities significantly influence how the area is addressed.

### ■ Outlook

Computational thinking in Denmark can be described as an area under development. Although it is not yet a standalone subject, many of its principles are already integrated into teaching through digital literacy, interdisciplinary working methods, and a focus on problem-solving.

Looking ahead, CT appears poised for a stronger and more systematic connection to compulsory education, particularly in relation to

work on technology understanding and digital competence.

### ■ Technology Comprehension

will be introduced as a new two-year elective subject in the upper secondary level of compulsory school from the 2027/2028 school year.

The subject will also be integrated as an interdisciplinary element in Danish, Mathematics, and Nature/Technology to strengthen students' digital competencies.

The subject aims to educate students for engagement and agency in a digital world. It covers five central content areas: Digital phenomena in students' everyday lives, digital phenomena in the world, creativity and design, computational modeling and programming, and Inquiry and taking a stance.

If offered, the subject concludes with a practical/oral exam, to be held for the first time in the 2028/2029 school year.



# Annex

## II - CT in secondary schools' curricula in the Netherlands

### A new curriculum since 2025

In the Netherlands, Computational Thinking is increasingly positioned within a broader vision of digital literacy. The new curriculum for primary education gives digital literacy a much stronger and more explicit place in national curriculum objectives. Programming is included as part of Designing and creating, while Computational Thinking strategies are connected to using digital technology to create, solve problems and approach tasks.

For teachers, this means that Computational Thinking is not simply about teaching children to code. It is about helping them develop a way of approaching problems: systematically,

creatively, critically and iteratively. In this sense, Computational Thinking is both a foundation for digital literacy and a transferable way of thinking that can enrich learning across the curriculum.

#### ■ Digital literacy

In the Netherlands, Computational Thinking is embedded within the broader concept of digital literacy (digitale geletterdheid). This means that CT is not necessarily treated as a separate school subject. Instead, it can be taught through digital literacy and connected to subjects such as mathematics, science, language, citizenship and the creative subjects.

This approach reflects an important principle of the Dutch education system: schools have

considerable freedom to determine how they organise their curriculum, while national curriculum objectives define what students should learn.

#### ■ Computational Thinking and digital literacy

The Dutch concept of digital literacy is broader than the ability to use digital devices. It encompasses knowledge, skills and understanding related to digital technology, digital media, information, data, artificial intelligence and the relationship between technology, people and society.

The new Dutch curriculum framework for digital literacy makes this broader approach particularly clear. The new primary



education curriculum describes three interconnected domains: Practical knowledge and skills, Designing and creating, The digitalised world

Within these domains, students learn not only how to use digital technology, but also how digital systems work, how digital products can be created, how programming can be used, and how digital technology influences individuals and society. Computational Thinking is therefore best understood as one of the ways in which students learn to understand, create and critically engage with digital technology.

### ■ A new curriculum for digital literacy in primary education

An important recent development is the introduction of a dedicated curriculum framework for digital literacy in Dutch primary education.

Until recently, digital literacy did not have its own statutory set of curriculum objectives in Dutch primary education. Schools were already expected to prepare students for a digital society, and many schools had developed their own approaches to areas such as ICT skills, media literacy, information literacy, programming and Computational Thinking.

However, there was no separate statutory framework defining national objectives for digital literacy. This has now changed. In

2025, the new draft curriculum objectives for digital literacy were published, and in 2026 SLO published the updated primary education curriculum objectives. The new objectives give digital literacy a much clearer position within the national curriculum.

The new primary education objectives explicitly include programming within the domain Designing and creating. They also refer to the development of computational thinking strategies when pupils use digital technology to create products, solve problems and approach tasks.

For teachers, this represents an important shift. Digital literacy is no longer simply about teaching pupils how to operate digital devices. The curriculum places greater emphasis on understanding digital technology, creating with it, thinking computationally and considering its impact on society.

### ■ Computational Thinking across the primary education curriculum

The Dutch curriculum does not require Computational Thinking to be taught exclusively as a separate subject. Instead, digital literacy is designed to connect with other areas of learning. The national curriculum documentation explicitly describes relationships between digital literacy and subjects such as Dutch language, mathematics, citizenship, science, social studies

and the creative subjects. For example, students can work with data in mathematics, critically evaluate digital sources in language education, or design digital products in creative subjects. This cross-curricular approach creates many opportunities for Computational Thinking.

### ■ The role of the teacher

In the Dutch context, the teacher plays an important role in connecting Computational Thinking to meaningful learning situations. Rather than focusing exclusively on teaching specific programming commands, teachers are more used to ask questions such as: What is the problem we are trying to solve? Can we break this problem into smaller parts? What information do we actually need? These questions help students develop the underlying habits of Computational Thinking.

### ■ From consumption to creation

A central idea in the new Dutch approach to digital literacy is that students should not only be consumers of technology, but should also become creators and critical participants in a digital society. The new primary education objectives reflect this by combining practical digital skills with designing and creating, programming, and understanding the digitalised world.



Computational Thinking contributes directly to this ambition.

### ■ A developing curriculum

It is important to recognise that the Dutch curriculum is currently undergoing a significant period of change. The existing primary education curriculum objectives date from 2006. The Netherlands has been updating these objectives in order to provide greater clarity about what students should learn while maintaining room for schools and teachers to make professional choices about how learning is organised.

The new digital literacy objectives are part of this wider curriculum reform. The updated objectives for primary education were published by SLO in 2026 and are now being incorporated into the national regulatory framework. The implementation of the revised curriculum is taking place in phases.

For schools, this means that the coming years are likely to be a period of curriculum development. For teachers working in Dutch primary education, the new curriculum provides both a clearer framework and an opportunity.

### ■ Computational Thinking in Secondary Education

In the Netherlands, the new curriculum for digital literacy is

designed as a continuous learning progression from primary education into secondary education. The underlying principle is that students first become familiar with digital technology and computational thinking strategies in primary school and then progressively develop greater independence, abstraction and critical understanding in secondary education. In the lower years of secondary education, the foundations established in primary education are maintained, broadened and deepened, while the contexts and problems students work with become increasingly complex.

The new Dutch curriculum objectives for digital literacy in secondary education are organised around three domains: Practical knowledge and skills. Designing and creating, The digitalised world. These domains are closely interconnected. Students learn to use digital technologies, but also to understand how these technologies work, create digital products and critically consider the impact of digital technology on themselves, other people and society.

The three domains are reflected in three overarching curriculum objectives: The student uses digital technology and digital media. The student creates digital products. The student participates in the digitalised world.

Within Designing and creating, programming is explicitly identified as an area of learning. This provides a direct curricular connection to Computational Thinking.

### ■ Computational Thinking in secondary education

In the first years of secondary education, Computational Thinking should increasingly move from concrete activities towards more abstract and complex problem-solving. Designing an algorithm that uses conditions and repetition; analysing and modifying an existing program; working with variables and data; identifying patterns in larger datasets; modelling a real-world process; decomposing a complex problem into computational sub-problems; testing, debugging and evaluating solutions. The important progression is therefore not simply that students learn more programming commands. Rather, they become increasingly capable of independently analysing problems, selecting appropriate strategies and evaluating the quality of their solutions.

### ■ From Computational Thinking to Informatics

There is also a more specialised pathway in Dutch secondary education. In havo and vwo, digital literacy in the lower years is



intended to provide a foundation that can connect to Informatics (Informatica) as an elective subject in the upper years. The curriculum explicitly recognises this connection. This distinction is important. Digital literacy is a broad competence that all students should develop. Informatics, on the other hand, provides a more specialised understanding of computing and digital systems.

Computational Thinking sits at an interesting intersection between the two. All students can benefit from learning to decompose problems, recognise patterns, design algorithms and debug solutions. Students who choose Informatics can subsequently

develop these ideas in greater depth, alongside topics such as programming, data, algorithms and computational systems.

### ■ Different pathways, shared foundations

The Dutch secondary education system contains several educational pathways, including vmbo, havo and vwo. The new curriculum aims to provide a common foundation in digital literacy while allowing the context and level of application to become increasingly relevant to students' educational pathways.

In vmbo, digital literacy has a particularly strong connection with

vocational education. Students can encounter digital technology in practical and professional contexts and develop the ability to use technology effectively within vocational profiles. In havo and vwo, digital literacy provides a broader academic foundation. Students are expected to use digital technology critically and across different subjects, while aspects of digital literacy can also provide preparation for the Informatics elective in the upper years. This means that Computational Thinking can have different contexts while retaining the same fundamental principles.



# **Annex**

## **II - CT in secondary schools' curricula in Curaçao**





# Colofon

© BRIDGE

This publication is a product of BRIDGE (2024-1-NL01-KA220-SCH-000245907), which was funded with support from the Erasmus+Programme of the European Union.

This publication solely reflects the views of the authors, and the Commission cannot be held responsible for any use that may be made of the information contained therein.

## Project Coordinator

Stichting Children's Science Museum Curaçao, Curaçao

## Partners

Universidad de la Iglesia de Deusto Entidad Religiosa, Spain

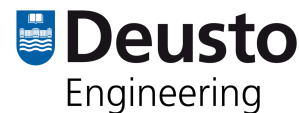
Aarhus Public Libraries, Denmark

Studio Kabritu, Amsterdam

## Design

Logo, Design and website: Studio Kabritu, Amsterdam

AARHUS  
PUBLIC  
LIBRARIES



Co-funded by  
the European Union

