

Course title	Coding and computational thinking in the classroom
Course code	047
Course category	STEM and digital education
Course purpose and overview	<p>Do you know the difference between coding and computational thinking skills when it comes to classroom practice and instruction? Join this course to learn new concepts, practices and perspectives about coding and computational thinking.</p> <p>The course is about what coding and computational thinking are, and why students can benefit from them, while strengthening their problem-solving skills. The course will outline some of the online tools used to teach coding, such as Scratch and other educational resources, such as CoSpaces Edu. You will learn how to use these tools in a creative and engaging way through hands-on activities and exercises to develop projects and lessons for your students. Moreover, the focus of these activities will also be to understand how they can help foster computational thinking and how to explain to your students the steps required to solve problems as a computational thinker. In addition, you will have the chance to discuss those topics with colleagues from other schools in Europe.</p>
Course structure and content	<ul style="list-style-type: none"> • Introduction to coding and computational thinking • Why integrate coding and computational thinking in your classroom? • Sharing experiences with coding • What is unplugged coding? • Advantages and challenges of using unplugged activities in the classroom • Unplugged coding activities • What is a visual programming language? • Why use visual programming in schools? • What visual programming language can teachers introduce in their classroom? • Why use Scratch? • How can teachers bring Scratch to their classroom? • What robots can teachers use in the classroom? • Why should teachers integrate robotics into the classroom? • What robots can teachers use in the classroom? • What are tinkering and making? • Why should teachers integrate tinkering and making into the classroom? • Work presentations • Course roundup and review <p>Validation of learning outcomes</p>
Duration	One week
Daily programme example	<p>Here is an example of the programme:</p> <p><i>Day 0</i> (usually Sunday) Arrival date</p> <p><i>Day 1</i> Welcome and introduction Ice breakers and team-building exercises Course introduction and what is coding? Scratch overview Cultural and social activities Feedback day 1</p> <p><i>Day 2</i> Scratch overview 2 and project work</p>

	<p>The use of graphical effects</p> <p><i>Day 3</i></p> <p>What is computational thinking?</p> <p>Unplugged activity and topic introduction (how to create a story)</p> <p>Cultural and social activities</p> <p><i>Day 4</i></p> <p>Pass-it-on project</p> <p>Unplugged activity and topic introduction (interactive coding)</p> <p>Interaction and debug projects</p> <p>Cultural and social activities</p> <p><i>Day 5</i></p> <p>Inputs for lesson design</p> <p>Course wrap-up and certificate ceremony</p> <p>Discussion of future cooperation and planning follow-up activities / Brainstorming dissemination ideas</p> <p>Final feedback</p> <p>Validation of learning outcomes and a certification ceremony</p> <p>Cultural and social activities</p> <p><i>Day 6</i></p> <p>Full-day trip</p> <p><i>Day 7</i></p> <p>Departure date</p> <p>Programme details may be subject to amendment based on trainer, participant needs and other factors such public holidays. Changes might be needed to make up for time lost due to unforeseen or changing circumstances which might be out of the reasonable control of the hosting organisation.</p>
Learning objectives	<p>Understanding the importance of integrating coding into the classroom</p> <p>Understanding the importance of integrating computational thinking into the classroom</p> <p>Understanding the importance of integrating robotics into the classroom</p> <p>Becoming familiar with innovative tools and approaches such as visual programming tools, unplugged activities, robotics, tinkering, and making and coding for all subjects</p> <p>Discovering training materials and resources</p>
Learning outcomes	<p>Understand what coding and computational thinking are, what the benefits are, and how you can use them in your class</p> <p>Familiarise yourself with computational thinking and explore its use in everyday-life processes</p> <p>Learn coding new concepts, practices and perspective</p> <p>Feel confident in developing projects on Scratch</p> <p>Create interactive virtual space with CoSpaces Edu</p> <p>Design coding and computational thinking lessons plans</p>
Assessment and validation of learning outcomes	<p>Observations, simulations, evidences extracted from individual and group work.</p>
Target audience	<p>Teachers of any subject, educators, and school administrative staff. No prior experience with Scratch is necessary</p>
Admission requirements for participants	<p>No specific requirements</p>
Language of delivery	<p>English</p>

Language level requirements for participants	B1 or more
Maximum number of participants	16

Please note that the course outlined is intended as an example only and may not necessarily be fully executed in accordance with all its details. Our need analysis is primarily based on enrolment information, information shared at kick-off meetings, and pre-evaluation of competencies. As such, it is possible that the programme may be adjusted to better accommodate the diverse needs of participants.