Cultural heritage in primary education:

innovative teaching practices

CULT-TIPS

PEDAGOGICAL MODEL

Final

IO1 – CULT-TIPS Pedagogical Model

A3 – Pedagogical Model description

fb
finance & banking
Associazione
per lo sviluppo organizzativo
e delle risorse umane

Table of contents

Executive Summary

The CULT-TIPS Pedagogical Model helps provide teachers and educators (e.g. museum educators) with a reference framework for teaching and using Computational Thinking (CT) in the field of humanities and, more specifically, cultural heritage and arts. In this sense, this Pedagogical Model aims at guiding teaching professionals in primary schools, by giving them some practical tools and suggesting some principles to follow.

CT is an analytical and methodical approach that involves breaking down complex problems into more manageable sub-problems, using a sequence of steps (dubbed algorithms in computer jargon), reviewing how similar problems may require similar solutions, and determining if a computer can efficiently solve said problems. Being at the basis of programming and coding activities, CT has been always associated with computer sciences and STEM disciplines in general. However, more and more researchers have started considering CT as cross-disciplinary, and its application to other subject areas as an effective way to teach this method. Accordingly, the CULT-TIPS Pedagogical Model is addressed to educators and promotes the shifting from traditional pedagogical approaches in humanities to a more innovative learning environment centred on the idea of "coding to learn" rather than "learning to code."

The Model has been built on the Compendium on Computational Thinking Methods (A1) and the Compendium on Art and Cultural Heritage (A2) developed by the Consortium. It is based on the learner-centred principles: 1. Including learners in the decision-making process about their learning; 2. Valorising learners' already existing knowledge, skills, competences (SKC); 3. Involving learners in the teaching process as co-creators.

Moreover, the CULT-TIPS Pedagogical Model has a series of features, which makes it extremely effective and efficient. First, the model is holistic in the sense that it provides a comprehensive perspective on CT and on how to teach it in non-STEM disciplines. Second, the model is practical as it encourages teaching professionals to use a "learning by doing" approach and to apply CT to real-world experiences. Third, it is engaging and stimulating

inasmuch as it motivates and empowers students to manage their own learning. Fourth, the model is process-oriented, which means that learning activities follow the continuous improvement cycle principle and that repetition is seen as key to improve final outcomes. Lastly, the model is flexible in that it addresses common European challenges and is transferrable to different national contexts.

In order to ensure the quality of the learning process, the CULT-TIPS Pedagogical Model follows a Quality Assurance Framework, which is based on the Plan-Do-Check-Act quality cycle:

1. PLAN: Teachers should plan how to integrate CT into the curriculum as a cross-cutting method to apply to different subject areas. Curriculum planning should result in a written plan and be followed by detailed lesson plans. Integrating CT in the curriculum can disregard neither a thorough students' and teachers' needs analysis nor the budget for software availability.

2. DO: The teacher should assign tasks to their students, setting deadlines and providing all the necessary inputs. A checklist will help to complete this phase.

3. CHECK: Planning and Implementing need to be checked and evaluated to find out potential inconsistencies in the learning. A questionnaire will be key to effectively carry out this phase.

4. ACT: Evaluating leads to reflecting and giving feedback so as to improve the learning/training activities. A questionnaire will be of use for this phase.

The rationale behind this Pedagogical Model is that of closing the gap between the KSC that students acquire at school and those that today's labour market requires. This ultimately means that students should become creators of knowledge rather than passive consumers. The main skills that CT contributes to acquiring are as follows: 1. Decomposition (breaking down a problem into smaller parts that can be solved more easily); 2. Pattern Recognition (identifying similarities in problems as a way to build solutions); 3. Abstraction (taking off

superfluous details and focusing on essential elements); 4. Algorithm Design (creating a series of instructions to solve a problem); 5. Logical Reasoning (applying rules to problem solving); and 6. Creative Thinking (working with and generating new ideas).

The CULT-TIPS Pedagogical Model therefore turns traditional education into "SMART (sensitive, manageable, adaptable, responsive, and timely) education." Rather than teaching students notions that they are required to retain, this model encourages teaching professionals to provide students with some tools and empower the latter to find their own creative solutions to a given problem. This would, in turn, strengthen the student-teacher relationship and make students active contributors to their learning process.

In conclusion, the CULT-TIPS Pedagogical Model offers a reference framework that teachers and educators in the field of humanities and, in particular, cultural heritage and arts can use to develop their own learning materials and programmes. This Pedagogical Model will be supplemented by the CULT-TIPS toolkit (03).

1. Aims and objectives of the CULT-TIPS Pedagogical Model

Computational Thinking (CT) and coding activities are widely considered new fundamental skills to be developed from an early age.

What is Computational Thinking?

Basically, Computational Thinking is a problem solving process that includes a number of characteristics and dispositions. It involves breaking down complex problems into more manageable and familiar sub-problems (problem decomposition), using a sequence of steps (algorithms) to solve them, reviewing how the solution transfers to similar problems (abstraction), and finally determining if a computer can efficiently solve them (automation). Therefore, CT enables to solve any given challenge through an analytical and methodical approach, teaching students to process information like a computer would. It guides students through a series of steps, similar to an algorithm, to solve open-ended problems. These steps are essential to programming and coding activities but useful far beyond any single discipline. Indeed, besides the development of computer applications, CT can be used to support problem solving in a creative and efficient manner across all disciplines, including the humanities, math and science.

One example of a resource adopting CT in the classroom is SCRATCH, created by the Lifelong Kindergarten group of MIT's Media Lab. Scratch is a free coding platform addressed to younger coders. By using a drag-and-drop block style, students can create animations, games and simulations without any previous knowledge of computer programming.

"Learning to code helps us to make sense of the rapidly changing world around us" (https://codeweek.eu). CT is the basis for programming and coding: while it helps kids to be prepared for the digital future of our world, it also enhances other competences, such as kids' ability to think creatively, communicate clearly, analyze systematically, collaborate effectively and apply the academic knowledge in other areas of life.

CT can foster creativity by allowing students not only to be consumers of technology, but also to build tools that can have a significant impact on society (Mishra and Yadav, – 2013: 11).

Therefore, introducing computational thinking at school can be the right means through which logical and problem-solving skills can be encouraged among kids from an early age.

Pedagogical approaches to embedding computational thinking in primary schools

Several efforts to embed computational thinking in primary schools based on computational thinking competences have been made all around the world. The majority of these efforts has focused primarily on computer science curricula and programming tool use (e.g. Scratch and Code.org's block-based programming environment). The most prominent idea of CT is to allow students to develop a foundational understanding of computing and develop competences that move them from being users of technology to producers of information technology (Yadav *et al.*, 2014).

While it is valuable for students to learn computational thinking within the context of computer science curricula and programming environments, it is evident that in many countries the constraints of a K-12 classroom might not make it feasible for all schools to have access to standalone computing courses. Moreover, some researchers start to think that computational thinking ideas are cross-disciplinary and can be embedded across the primary and secondary subject areas (Yadav, Hai Hong & Chris Stephenson, 2016).

Several schools and educational experiences around Europe (see CULT-TIPS O1_A1_Compendium on CT methods) follow the same pattern and goal: using CT and Coding as a *meta-competence* and exploring its value in relation to other skills and curricular activities. Problem solving, data representation and modelling, group collaboration, search for patterns, application of rules, usage of logical steps, and logical reasoning may help students to take advantage of computational thinking in many school subjects and in all aspects of their lives.

CULT-TIPS innovative approach

The CULT-TIPS approach is based on the belief that larger importance has to be given to the idea of "coding to learn" rather than "learning to code" and that computational thinking can be applied to the study of all school subjects. So, how to embed computational thinking in humanities and art classes in primary school? In the next chapters, we try to develop a CULT-TIPS Pedagogical Model that could be a theoretical reference for teachers and educators interested in applying it in their humanities classes.

The CULT-TIPS Pedagogical Model helps teachers and educators to be confident with CT tools, supporting them in shifting from traditional pedagogical approaches to a more innovative, new learning environment. The intent is to contribute to the spread of innovative teaching practices applying Computational Thinking approaches to the promotion of art and cultural heritage at school.

The model includes principles, criteria and approaches for introducing pupils into the world of digital learning. It will guide teachers to the transition towards a participatory and intercultural approach to cultural heritage, by fostering dialogue and collaboration between teachers and primary school students (i.e. learning by doing). The aim is to enhance primary school teachers' knowledge of computational thinking methods and become familiar with CT tools that support teaching activities with increasingly interactive, experimental and participatory learning experiences (e.g. using online resources, digital games, experimenting digital story-telling).

The main goal of CULT-TIPS Pedagogical Model is to improve teaching humanities subjects (Cultural Heritage and Arts above all) and in particular the quality and efficiency of CT and digital tools as effective and innovative teaching methods applicable to all subjects.

The model includes principles, criteria, approaches that teachers can use to attract children's attention in class, stimulate their creativity and develop critical thinking. Furthermore, it enhances stakeholders' involvement and motivates activism in this field.

The added value of this Model is that it connects teaching theories, pedagogical methods and necessary skills acquisition in an integrated manner by proposing a pedagogical model that can orient activities and educational scenarios as well as by giving principles and guidelines for teaching practice.

2. Target group of the CULT-TIPS Pedagogical Model

Direct groups:

- Teachers in primary schools responsible for teaching humanities subjects, in particular art and cultural heritage;

- Pupils in primary schools;

- Educational and Training providers for teachers' continuous professional development;

- Local and Regional Public Authorities responsible for primary schools' curricula for humanities subjects, in particular Cultural Heritage and Art;

- Educators in museums and cultural institutions.

Indirect groups:

- Pupils' parents;

- Policy-makers;

- Communities in which the learners are growing up;

- Private and Public initiatives supporting Art and Cultural Heritage;

- Other stakeholders (e.g. public and private organisations, research centres, Universities, VET providers, trade unions, business sector, etc.).

3. CULT-TIPS Pedagogical Model: Main features

Among the main objectives of today's globalized society there is the development of students' 21st century skills, including creativity, critical thinking, and problem solving.

"The 21st century skills" were first mentioned in 2003 by the 'Partnership for 21st century skills' following several studies started in 1991 by the U.S. Department of Labour with the study 'What Work Requires of Schools' (Secretary's Commission on Achieving Necessary Skills, 1991). This clearly indicates that U.S. economy requires a different type of worker than before, and that a successful life requires new skills.

21st Century Student Outcomes and Support Systems
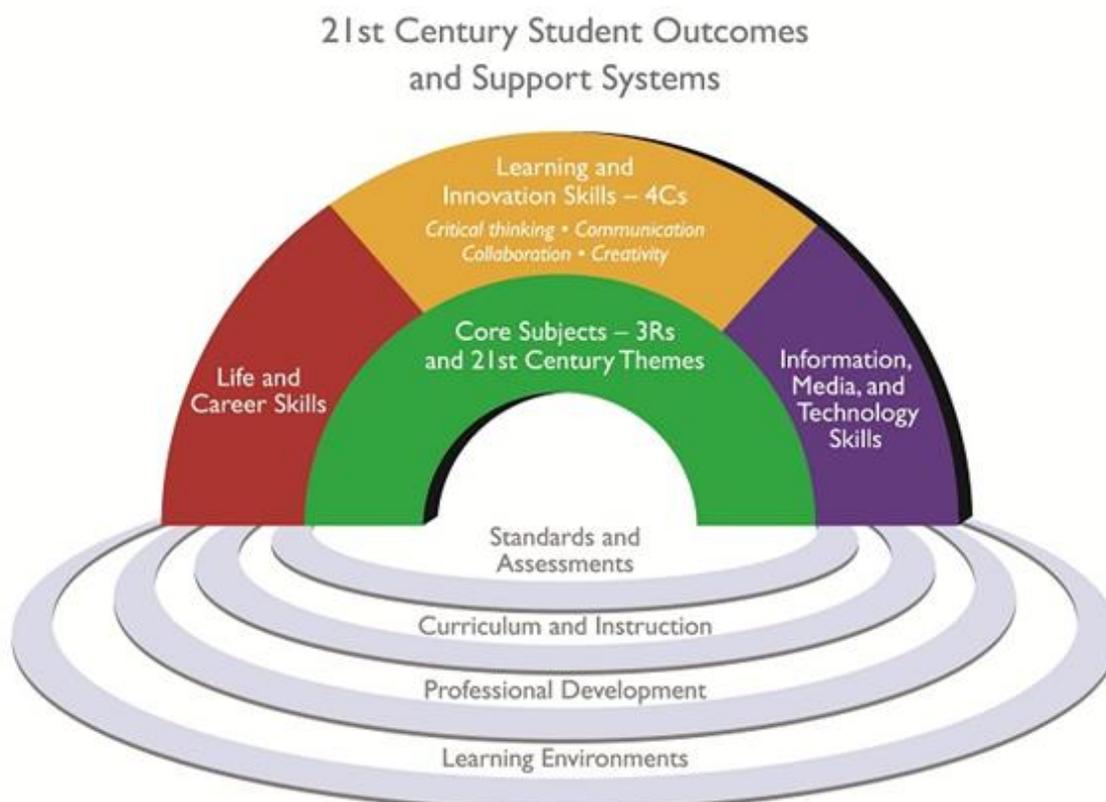
Partnership for 21st Century Skills: Framework for 21st Century Learning

In 2003, the 'Partnership for 21st century skills' was established by various parties from the world of education and business to identify the desired skills and develop them into a framework. The aim was to close the gap between the knowledge and skills that students acquire at school and the knowledge and skills needed in the work and community of the 21st century.

To reach this goal it is necessary for students to actively take part in today's participatory culture, becoming creators rather than passive consumers of knowledge.

The progress and accessibility of computing technologies has the potential to engage students in this process.

This means that also the role of teachers needs to change in a way that they should facilitate the learning processes and at the same time guide and supervise them.

Creativity as well as problem solving and collaboration are among the most targeted skills in the 21st century. Thus, this pedagogical model supports the acquisition of these skills towards a holistic development of students in primary school settings.

In this direction, teaching students to think like a computer scientist, an economist, a physicist or an artist can be achieved through CT practices, as well as arts' activities.

CULT-TIPS Pedagogical Model main features:

The contents are based on an updated literature review on pedagogical models on CT and children engagement and on the Compendium on Computational Thinking Methods (A1) and Compendium on Art and Cultural Heritage (A2) developed by the Consortium.

It is addressed to primary schools teachers and specifies **skills** and **prerequisites** for them.

It facilitates the application of **computational methods** to art and cultural heritage in learning activities.

It consists of principles, criteria, approaches to:

❏ engage children;

❏ stimulate critical thinking, creativity, cooperation;

❏ hands-on approach "learning by doing."

It includes practical recommendations, instruments and tools to enhance stakeholders' involvement and motivate activism in this field.

The CULT-TIPS Pedagogical Model comprises a general and holistic framework for improving the teaching of humanities subjects (in particular Cultural Heritage and arts) applying Computational Thinking methods. It can be adjusted to the specific educational situation and context.

The teachers' training opportunities in CT discussed in the literature largely focus on pedagogical aspects rather than technological skills. Most training seems to be designed for all subject teachers, sometimes with a particular focus on STEM teachers. The pedagogical

approaches addressed include storytelling, problem solving, deductive and inductive pedagogies with a focus on computational models and simulation. Training activities are often designed specifically to be hands-on so that teachers can more easily transfer their new skills to their classroom.

The proposed CULT-TIPS Pedagogical Model is based on the learner-centred principles which allow:

1. The inclusion of learners in decisions regarding how and what they will learn and how the learning will be assessed;

2. The valorisation, respect and accommodation of each learner's unique background, interests, abilities and experiences;

3. The involvement of the learner in the teaching and learning process as co-creator.

With the learner-centred principles, the learning process is influenced by different factors: metacognitive and cognitive; affective and motivational; developmental and social; individual diversity.

Furthermore, the Model follows the "learning by doing" pedagogy in which teachers engage learners in more hands-on, creative modes of learning. This approach is the process whereby people make sense of their experiences, especially those experiences they are actively engaged in making things and exploring the world. The experiential learning allows for the active engagement of students, creating opportunities to learn through doing, reflecting on the activities implemented and finally enabling them to apply their theoretical knowledge to practical activities in a multitude of settings inside and outside of the classroom.
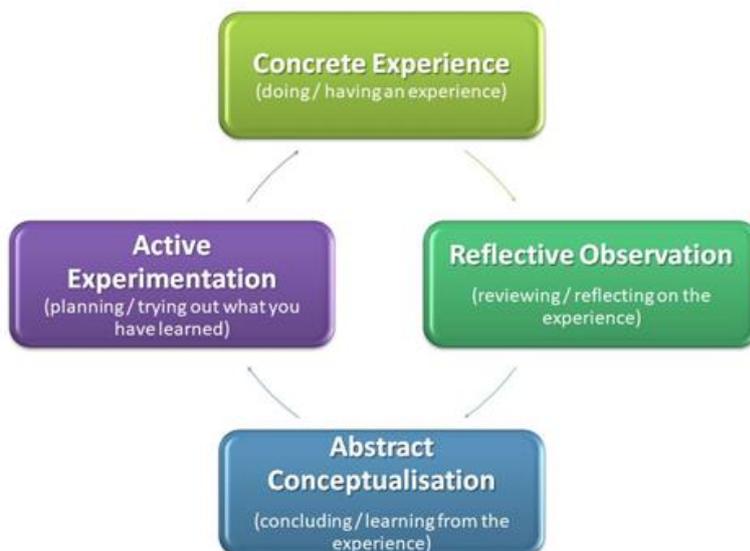
The CULT-TIPS Pedagogical Model is developed to address the different and common European and national challenges and strategies to cultural heritage. At the same time, the framework is also flexible enough to be transferred to different national contexts. It can be considered as a combination and adaptation of the educational theories and positions, and focuses mainly on the following principles:

a) Holistic

CULT-TIPS is holistic in regards to the subjects, their connections and the teaching methods and focuses on understanding general ideas while? applying CT to art and cultural heritage rather than accumulating specialised knowledge.

This model is intended to provide a comprehensive perspective over this area by focusing on how to approach the convergence among CT, collaboration and creativity skills in practice rather than what to teach. It contributes to the learners' personal development.

CT, collaboration and creativity ultimately lead learners to holistic development since each single skill and the whole system of competence reflect the intellectual, feeling and doing pillars.

b) Practical

CULT-TIPS Pedagogical Model supports learners in gaining knowledge, acquiring skills and building competences through real-world experiences and observations.

Introducing and practicing tools, techniques and materials provide a set of skills that help students during a creative process. Precisely the instruction of techniques helps students later on in the implementation of their project.

While experimenting, designing and making, children learn a lot about themselves: about their competences and talents, about dealing with materials and techniques and how they deal with setbacks and the recalcitrant nature of materials. Process and product come into balance with each other.

c) Engaging and stimulating

The Model allows teachers, who know their students, to engage them in building supportive, inclusive and stimulating learning environments. In this way, teachers motivate and empower students to manage their own learning. Intuitive knowledge generates powerful ideas and involves CT (imagination/creativity).

d) Process-oriented

The Model is process-oriented in regards to the learning activities. Learners can explore art and cultural heritage in a self-regulated and creative way through exploration and creation. The process is constant and focuses on the continuous improvement of learners' competences and skills, creating at the same time deep knowledge.

It follows the principle of a continuous improvement cycle: processes are repeated several times to ideally obtain improved results, also in terms of competences, after each cycle.

f) Flexible

The CULT-TIPS Pedagogical Model addresses the demand for new models for CT applications, especially referred to Cultural heritage and art subjects, providing a didactical framework to be adapted to the specific situation and context of the different schools, regions, educational systems and cultures in Europe and outside.

The use of CT enables the transfer of the problem-solving process to a broad range of situations, like confidence in dealing with complexity, determination in working with

complex problems, tolerance for ambiguity, ability to manage open-ended projects, and ability to communicate and work in collaboration with others for a common goal or solution. [International Society for Technology in Education, and Computer Science Teachers Association, "Operational definition of computational thinking for K–12 education." (2011)]

3.1 Quality Assurance Framework

The quality of the pedagogy is crucial for the learning process. It allows pupils to have access to knowledge, feel confident in using different skills, and accept challenges as an integral part of their learning. To guarantee the quality, schools and teachers should be aware of the challenges that may arise and come up with a solution.

In several European countries (Bocconi et al., 2016) the debate on the need for CT skills is taking place, but there is not yet a consensus on what is at the heart of this learning domain. CT-skills for many schools are new elements of the curriculum, and for this reason, there are concerns about its collocation within the curriculum: should it be separated or integrated? Despite the relatively large amount of research focused on definitions, frameworks and CT-related tools, researches on the perspective of teachers concerning integration of CT are still limited. Finally, there are concerns about the assessment of CT-skills: how to assess what students have learned?

The Quality Assurance framework of the CULT-TIPS Pedagogical Model will be based on the **Plan, Do, Check, Act** quality circle.
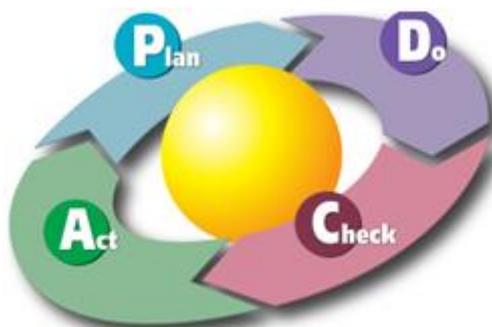
Engels Deming circle

Here below, we present the quality principles that teachers and educators should follow in order to ensure the quality of the implementation of the model.

Plan: Quality starts with planning. The teacher should first plan the pedagogical approach, taking into account teachers' needs, learning objectives, available resources and other factors.

Planning in CULT-TIPS model has two levels:

A. **Curriculum level:**

Schools must have a plan for CT and a written curriculum including clear answers to the following questions: What do we want to teach? How do we want to teach (unplugged, with devices, apps, websites)? Who do we teach and who will teach? When do we teach and how often?

With a curriculum as foundation to bring CT into schools, teachers, and their professional development, are involved in the process.

The majority of teachers are not trained to teach CT skills (Yadav et al., 2018) and still a little has been done in the area of professionalization.

In order to promote teachers professionalization, Yadav et al. (2018) argue for networking of teachers of this subject and for supporting IT teachers and teacher trainers from the first-grade.

Teachers should plan how to integrate computational thinking into the curriculum, introducing interdisciplinarity into different courses. They should take into account the following problems in teaching CT:

- providing feedback and assessing progress for many learners at once,
- pedagogical challenges of teaching CT concepts at certain ages,
- inadequacies in existing digital learning environments to teach CT,
- a lack of systematic assessment procedures,
- challenges of differentiation, lack of learner knowledge and lack of resources,
- limited ways to get certified for in-service and pre-service teachers,
- ways to collaborate with other educators as a form of community building.

Planning a curriculum can be for the whole school, involving all teachers and staff, or at the level of individual classes, involving the class teacher and maybe other teachers of specific subjects (e.g. arts, drama, etc.). Depending on the school, curriculum planning can be for the whole school year, or for a shorter period (e.g. semester, trimester etc.).

Curriculum planning takes into account different factors, such as grade, class size, pupils' skills, needs in skills, etc. Taking into account all these inputs, the teacher should define the objectives of introducing CT into the curriculum. The objectives will be different for each class level (e.g. different for first graders), and different for classes with different experience with computational thinking (e.g. classes where computational thinking is introduced for the first time or classes that have embraced it before).

Resource planning involves the requirement in terms of teachers, teachers' skills, equipment, software, learning resources etc. The availability of resources will determine what type of pedagogical method will be used, how the courses will be planned, and what extra resources are needed.

A rich learning environment is fundamental to provide the necessary guidance and challenge for learners as they explore topics, study scientific visualizations and collaborate with others. The best approach is to use an embedded assessment in these technological environments in such a way that it provides immediate feedback for learners but also allows teachers to monitor progress real-time for targeted tutoring

It is likewise fundamental to look at current methods for creating time and resources so that teachers can plan together, share their practices, and learn from each other.

Curriculum planning will end up in a written plan, stating the objectives of the introduction of computational thinking in the curriculum, the overall strategy, the required resources, the pedagogical methods to be used, and the allocation of responsibilities among teachers.

A. **Lesson plan level:**

At the level of the lesson, the teacher should prepare the lesson plan for each lesson. Lesson plans are linked to the overall curriculum and provide details on specific objectives, learning contents, activities, pedagogical methods, and learning resources at micro level.

a) **How to integrate Computational Thinking in the curriculum.**

I. Students' needs' analysis. As mentioned before, the curriculum should be based on students' needs, so the more customised it is, the better and more effective it would be for learners. According to this analysis, students will state their IT level along with coding experience (if any) along with their computational skills level.

II. Teacher's training needs. Teachers need to be trained in order to impart their knowledge and skills to students. Their training should be developed according to the students' needs' analysis so as to cover the gaps detected in previous phase (I) as much as possible. During their training, teachers need to be active and realise that it will upgrade their IT skills and support them in dealing efficiently with interdisciplinarity.

III. Budget for Software availability or purchase. Schools will inevitably have to invest in modernising their equipment and software. This investment will engage teachers in integrating technology in their classes and students in upskilling themselves.

b) **How to develop a Lesson Plan**

Teachers could/should introduce new lesson plans. In this way, they will trigger their students' curiosity and invite them to dynamically participate in class. The challenge of incorporating Computational Thinking into a lesson plan can also reinforce the teacher-student relationship as well as the peer-to-peer one, since CT boosts cooperation and team spirit.

Do: This is the implementation phase: what has been planned, here finds its time and place. Teachers have developed a lesson plan which is in line with the Pedagogical Model and start

assigning tasks and mini projects to students. They set the deadlines, give necessary learning resources, suggest a significant method to be applied by all students, and provide examples in order to make everything clear.

A checklist will help teachers to run their courses and have the Pedagogical model as a reference. This list should also support those assignments and projects, on which students should work at home. In this phase, what has been planned is now implemented, so it is a chance for both teachers and learners to discover whether they are successfully responding to the tasks assigned to them. It is also an opportunity to write down any problems encountered so far, in order to make some corrective actions on time during the *Check-Act* phases.

Check: In this phase, what has been achieved needs to be analysed and reviewed with a critical mind. As far as the sustainability of the Pedagogical Model is concerned, both Planning and Implementing need to be checked and evaluated in order to see whether there are any inconsistencies in the Learning/Training activity or whether the Pedagogical Model is essentially supporting Computational Thinking or not.

The following questions have to be answered:

Who is responsible for each evaluation? (One for the LTA and one for the Pedagogical Model)

Is it going to be an external evaluator or an in-house one e.g. the school manager?

When are these evaluations taking place? (after the LTA, one every semester, once a year)

What is the method or the tools used for these evaluations? (questionnaires, interviews, focus groups etc)

a) A questionnaire will help on the evaluation of the teaching/learning activity (LTA):

- What has been accomplished successfully?

- What was the training/learning activity impact on teachers and students?

- Is their feedback positive or negative?

Here are some key evaluation questions:

- Are the teachers using Computational Thinking as a daily practice?

- How many of the teachers trained have been actually integrating Computational Thinking in their class?

- What is the impact of teachers' training on students' performance?

b) A questionnaire will help for the evaluation of the Pedagogical model, and it will be documented with a SWOT orientation: what are the Strengths and Weaknesses of the Pedagogical Model, and what are the Opportunities and Threats that will bring to all stakeholders (students, teachers, school directors, parents)?

Act: Reflection notes will give feedback from the previous level (Check), but in the meantime, these notes will be fundamental to carry out corrective actions on the lesson plans or to design next ones.

- What measures need to be taken for the improvement of the learning/training activity (LTA) and the Pedagogical Model in general?

-What are the lessons learned after the first year of incorporating Computational Thinking in the school curricula?

- Are there any recommendations for the improvement of the LTA or the Pedagogical Model?

- Will these improvements be referring to the content or the method(s) used for the implementation of the LTA or the Pedagogical Model?

Quality criteria:

·   Pedagogical model's appropriateness for students: Did the model fit on students' skills, knowledge level, age, etc.?

·   Pedagogical model's appropriateness for every activity: Did these activities fit students' needs, skills, educational purposes, etc.?

INDICATORS			
	low	**medium**	**high**
PLAN			
Students' knowledge on coding			
Students' computational skills			
Teachers' knowledge on coding			
Teachers' computational skills			
The availability of software in the school			
Purchase cost			
Confidence on introducing and supporting new lesson plans			
Feasibility on applying new lesson plans in the curricula			
Possibility on implementing study visits and gain awareness on new lesson plans' formats			
Students' assignments based on projects			
DO			

Integration of a specific programming language (e.g Scratch) in the lesson plans			
Teachers' ability to explain the assignments based on the model			
Students' ability to understand and correspond to these assignments			
Adequacy of the Pedagogical Model compared with previous ones			
CHECK			
Students' consolidation			
Students' ability to apply what they have learned			
Pedagogical model's impact on teachers			
Possibility of integrating the model in the following years			
ACT			
Teachers' satisfaction with referring to and using the model			
Students' satisfaction with referring to and using the model			

3.2 Knowledge, skills and competences

The CULT-TIPS Pedagogical Model is related and connected to the concept of knowledge, skills and competences (KSC) as well as the abilities of teachers to build them.

Following the learner-centred principles, learners are put at the core of the learning process through the design of appropriate learning objectives, environments and tasks for self-directed or teamwork learning.

CT is considered among the main skills for the future workforce, and although it is thought of as a problem solving technique applicable specifically to computer science, today educators perceive its potential in a whole range of different subjects.

The main purpose of introducing CT to humanities subjects, in particular Art and Cultural Heritage is to enable children to meet and find solutions to current and future challenges.

Teachers, on the other hand, should be supported in this new challenge of introducing CT in the curricula.

These are the main skills and aptitudes to be learnt within the chosen subjects:

1. **Decomposition** i.e. the skill of breaking down a problem into smaller parts to solve it.

   Breaking things down into smaller parts and dealing with each separately is a problem-solving approach which programming languages can simplify through procedures, functions or objects. Moreover, it can be used, whether pupils are programming or not: planning a story, working on collaborative projects, solving multi-step problems in maths or putting on a play.

   Students can use this skill to tackle a complex task, feeling less discouraged.

2. **Pattern Recognition** i.e. common solution for common problems.

   Once a complex problem is decomposed, it is useful to look for similarities or "patterns" in each segmented part of the problem. Recognizing a pattern or similar

characteristics helps to break down the problem and also build a construct as a path for the solution more effectively.

After short time coding, pupils will be able to recognize that they keep writing similar blocks of code repeatedly. Often these patterns become part of accessible code libraries: when learning how to code, it can be useful to look at or remix others' code.

Recognising and re-using patterns will be familiar to children from other subjects, too.

3. **Abstraction** i.e. taking off superfluous details and focusing on essential elements.

   Basically, once patterns are recognized, the process of ignoring unnecessary characteristics and specific details of patterns follows suit.

   This is about capturing the important structure of a system or a problem, being not too much concerned about details. We can therefore create a representation (idea) of what we are trying to solve.

   Effective programming sometimes means involving and working at more than one abstraction layer: looking both at the big picture and the detail.

4. **Algorithm Design (thinking)** i.e. creation of a series of clear and logical instructions to solve a problem.

   Algorithms are clear step by step instructions or the storyboard for an animation.

   These instructions or rules, if accurately followed (whether by a person or a computer), lead to answers to both the original and similar problems. Teaching children about algorithms means teaching them to think of the steps they, or a computer, need to follow to accomplish a task.

5. **Logical Reasoning** i.e. the process of applying rules to problem solving.

Computers simply follow the programs using the data or inputs received. It is possible to predict what they will do by understanding their algorithms.

Enabling pupils to make predictions about a piece of software they are using will help them to develop their ability to reason logically and make deductions from the information available to them.

6. **Creative thinking**: i.e. the attitude of working with and generating new ideas

Computational thinking is not confined to STEM fields, but it enhances learning in all disciplines.

Applying all the above mentioned skills and practices to art and cultural heritage education may have the effect of amplifying creativity among young children. By engaging children through diagramming sentences, creating storyboards or looking for artistic patterns, thinking computationally becomes a creative act.

Computational thinking teaches children how to observe and dismantle phenomena, events, stories, images: by doing so, it helps tackle larger problems with a step-by-step process that empowers children and supports them to unleash their imagination.

Step-by-step thinking and pattern recognition may therefore be used not only to understand art, history, geography, but also to encourage kids' creativity.

Furthermore, **Essential Attitudes** are required to solve problems as a computational thinker: empowering pupils to tackle uncertain problems and persist through challenges requiring iteration and experimentation; strong communication skills to facilitate collaboration and presentation; general curiosity across all disciplines.

In the end pupils will be able to:

- understand the digital transformation and its effects on the society,
- use digital tools and media,
- be critical and develop a responsible approach to digital technology,

- learn to put one's own creative ideas into action,
- learn how to solve problems.

4. Roles of teachers and learners

The CULT-TIPS Pedagogical Model is based on a strong relationship between teachers and learners.

In the digital age, CT is an essential skill for students and educators alike. This systematic approach to solving problems is at the foundation of not just computer science, but many other subject areas – and careers – as well.

A new approach to education is being developed currently —at all education levels— for including "computational thinking" as an essential element of the curricula, since it helps to promote problem-solving abilities, critical thinking and creativity among other skills.

Computational thinking provides a new opportunity for training 21st century skills and for developing new learning strategies. They will improve our students' access to the labour market regardless of the profession or sector involved.

These new skills will encourage schools to transform and adapt in order to cope with learner's needs.

Schools should make an effort to encourage the development of computational thinking as an opportunity to transform traditional pedagogies into smarter methodologies. Thus, it will be possible to transform traditional education into "SMART (Sensitive, Manageable, Adaptable, Responsive and Timely) education".

In today's technology-driven world, teachers have a new role in their students' growth. Teaching used to be focused on learning and retaining information. However, teaching is

currently changing, so students can take what they have learned and apply it to school, work and beyond.

This is where the thought process of computational thinking comes into play. Computational thinking is a higher-level process, whereby students can decipher problems and form innovative solutions. This valuable skill provides a unique method of problem-solving which is integral to many jobs of the future.

Computational thinking is an active teaching methodology, which includes game-based learning, problem-solving, learning by teaching, jigsaw techniques, etc. It places the responsibility of learning in the hands of the learners themselves. The teacher is the facilitator whose task is to let the learners learn on their own through the use of different interactive training activities.

The ability to solve a problem that no one taught you to solve is very empowering. Once you believe you can overcome hurdles, and you as a student believe that 'a teacher didn't teach me this but I might be able to figure it out anyway,' **you take ownership of your own education**.

To ensure that students develop this ability, it is important to provide teachers with adequate knowledge about CT and how to incorporate it into their teaching.

Students need to learn computational thinking early and often, with an emphasis on understanding computational processes. CT has the potential for application in a wide range of disciplines outside of computer and information sciences, it must be integrated in a variety of disciplines, so that students will learn CT throughout the school day, having the opportunity to use and combine their skills within different subject matter.

The goal is to teach students how to think like an economist, a physicist, an artist and to understand how to use computation to solve their problems, to create and to discover new

questions that can fruitfully be explored and not for everyone to think like a computer scientist.

Computational thinking is often associated with computers and coding, but it is important to note that it can be taught without a device. For that reason, CT can be a part of any classroom, including the classrooms of youngest learners in the primary grades.

By explicitly teaching and allowing space for the development of computational thinking, teachers can ensure that their young students are learning to think in a way that will allow them to access and understand their digital world. **Teaching computational thinking, in short, primes students for future success**.

The Computer Science Teachers Association (CSTA) has emphasized the role of CT in a classroom as "*a problem-solving methodology that can be automated and transferred and applied across subjects*"

Hence, in order to maximize the benefits of CT and get students interested in computing, we need to integrate CT in core content areas.

Many teachers are finding that the best way to teach CT is by integrating it into other subject areas. Integration creates a more authentic and interesting learning environment. To this end, teachers need to have computational thinking pedagogical capability to teach children.

The first step in this direction must be to train teachers to teach their students how to think computationally to incorporate it into their classroom. Some pre-service programs are already starting to fold computational thinking into their elementary education courses.

Teachers must be prepared to embed the basic principles of CT into their classes, what Dr. Yadav calls the **three A's**:

- *Algorithm, using a sequence of steps to solve problems;*

- *Abstraction, reducing a problem to its bare essence;*

- *Automation, using tools to automate the solution to a problem.*

It is essential to provide authentic computational thinking training to teachers who will then be able to do the same for their students. These efforts, however, need to involve content-area teachers and not just computer science teachers.

The table below represents the role of teachers and students (or both) for specific topics:

Topics	TEACHERS	BOTH	STUDENTS
PROJECT PRESENTATION **INITIAL SETTINGS**	Presentation of the Cult-Tips project to the students: its main objectives and the pre-existing idea for the work methodology;		Appropriate definitions of cultural heritage, humanities, education through art, computational thinking, scientific method, design methodology, problem solving methodology, …;
INITIAL PROBLEM **START POINT** **THEMES**	Presentation to students of strategies to identify/schematize problems;	Define the problem(s) to be solved and the starting issue(s), through brainstorming; definition of a single major theme; in the case of several themes born from brainstorming or large group meeting, teach strategies to	Identify problems to be solved or activities / themes they are interested in and would like to work on / contribute to;

		group problems according to variables; ..;	
KEYWORDS	Prior definition of possible keywords to work on with students;		Appropriate (or define in group) the first key words related to the problem/issue of departure/theme of the project;
HYPOTES ANALYSIS		Analysis of hypotheses to solve the problem(s) / to answer the starting question(s);	
LOGICAL REASONING **PROCESS LAYOUT** **RULES AND PROCEDURES**	Presentation to the students of the process of applying rules to problem solving - **LOGICAL REASONING**	Start defining the rules and schematising the whole process for solving the problem(s): what has been done so far and what can be defined already for the next steps;	
WORKING TEAMS	Training of working groups according to students' interests, skills and school level;	To combine the pupils' wishes and the needs of the project/problem when forming the working groups (according to the	Creation of the working groups: according to the partnerships desired by the students (friendships);

		partnerships desired by the pupils as well as their interests, skills and school level);	
PROBLEM DECOMPOSITION	Presentation to students of strategies for the skill of breaking down a problem into smaller parts to solve it – **DECOMPOSITION**;	Begin to break down the problem into smaller parts and/or new and more specific questions to answer the starting questions;	Choose, by the groups and their leaders, the parties
PLANNING	**PLAN**: Define resources, infrastructure and develop a Lesson Plan document;	Meet with groups of students to jointly participate in the define resources, infrastructure and develop a Lesson Plan document;	Approval of the work plan (lesson Plan Document);
CHECKLISTS **EMPLOYMENT AGREEMENT**	**DO: CHECKLIST**		Phased filled-in of checklists, built by the teacher, in Class Assembly, throughout the project and for all its tasks (the checklists arise from the basis for the tasks that is the Student Work Contract);
RECOGNISE STANDARDS **COMMON SOLUTIONS**	Presentation to students of strategies for the definition of common solution for	Definition of the common solution for common problems - PATTERN	Application of the hypothesis of solutions to the problem(s);

	common problems - **PATTERN RECOGNITION**;	RECOGNITION;	
ABSTRACT **THE SUPERFLUOUS vs. THE ESSENTIAL**	Take off superfluous details and focus on essential elements – **ABSTRACTION**;		Through the work of ABSTRACTION carried out by the teacher, the students appropriate strategies (by way of example) to apply them autonomously in the future;
THEORIZING ALGORITHMS **INSTRUCTIONS**	Creation of a series of clear and logical instructions to solve a problem - **ALGORITHM DESIGN (THINKING)**;		Through the work of ALGORITHM DESIGN carried out by the teacher, the students appropriate strategies (by way of example) to apply them autonomously in the future;
LOGICAL REASONING	The process of applying rules to problem solving - **LOGICAL REASONING**;		Through the work of LOGICAL REASONING carried out by the teacher, the students appropriate strategies (by way of example) to apply them autonomously in the future;

VERIFICATION **EVALUATION and SELF EVALUATION** **MONITORING**	<u>**CHECK (EVALUATION)**</u> - construction of the first questionnaire: presentation to students of strategies for this construction and definition of the objectives of a questionnaire (this one in particular);	Construction of questionnaire(s) for later phases of the project;	Answering to questionnaires;
ACTION, REFLEX and REACTION		<u>**ACT**</u>: starting from the response and analysis of the results present in the surveys, take reflection notes (these notes will be fundamental for corrective actions on the lesson plans or for designing next ones);	

| VERIFICATION EVALUATION and SELF EVALUATION MONITORING QUALITY STANDARDS OVERALL GOALS | GENERAL EVALUATION OF THE RESULTS OF THE APPLICATION OF THE PEDAGOGICAL MODEL (application - or reading of the results - of the evaluation instruments defined throughout the project) - **QUALITY CRITERIA** pedagogical model's appropriateness for students: - "Did the model fit on students' skills, knowledge level, age, etc.?"; - "The intent is to contribute to the spread of innovative teaching practices applying Computational Thinking approaches to the promotion of art and cultural heritage at school."; - "Teachers: facilitate the learning processes and at the same time guide and supervise this | Evaluation and self-assessment of the general objectives of applying the Pedagogical Model: - "Strong relationship between teachers and learners."; - "Pedagogical approaches addressed include storytelling, problem solving, deductive and inductive pedagogies with a focus on computational models and simulation. Often, training activities are designed specifically to be hands-on so that teachers can more easily transfer their new skills to their classroom."; - "The involvement of each learner in the teaching and learning process as co-creator."; | Evaluation and self-assessment of the general objectives of applying the Pedagogical Model: - "Be able to meet and find solutions to current and future challenges."; - "Move the students from being users of technology to producers of information technology."; - "Furthermore, Essential Attitudes are required to solve problems as a computational thinker: a) empowering pupils with the confidence needed to tackle uncertain problems and persist through challenges requiring iteration and experimentation; b) strong communication skills to facilitate collaboration and presentation; c) general curiosity |

	learning process.";		across all disciplines."; - "In the end students will be able to: understand the digital transformation and its effects on the society; use digital tools and media; be critical and develop responsible approach to digital technology; learn to put one's own creative ideas into action; learn how to solve problems.";

5. Implementation and Usage of CULT-TIPS Pedagogical Model

The CULT-TIPS Pedagogical Model helps provide a framework for those working in cultural heritage and arts and teaches how to introduce elements of computational thinking into their practice.

By surveying and drawing from best practices from both computational thinking teaching and from arts and cultural heritage teaching to create the model, and translating this into the core features, skills and roles which are needed to implement an interdisciplinary teaching and learning method, we intend to offer a framework which teaching professionals can use to develop their own learning materials and programmes.

Alongside this, the CULT-TIPS Pedagogical Model also lays the basis for a number of practical tools which will be developed within the CULT-TIPS project itself, turning the model into hands-on activities for teachers and pupils.

An Online Teacher Training Programme for primary school teachers will be developed in order to support them in teaching computational thinking through art and cultural heritage, with the pedagogical model forming the basis.

This will be supplemented by the CULT-TIPS toolkit, which will help teachers to apply innovative tools to combine computational thinking and arts and cultural heritage in the classroom. The toolkit will be designed to be useful and accessible for all teachers and educators (e.g. museums educators) and not only for those specialized in informatics or those with a high level of digital skills.

6. Activities example

Partners collected personal classroom experiences of educators at national level regarding the application of CT to humanities in Primary Schools.
Annexed to this document there are examples of four countries:

1- ITALY
2 - SPAIN
3 - PORTUGAL
4 - THE NETHERLANDS

In addition, we suggest to have a look at the SCRATCH project done by the Spanish Partner IES Hermanos Machado (2019/2020)
https://sites.google.com/ieshnosmachado.org/cometoknowme/products/scratch-projects?authuser=0

7. References

Mishra, P., & Yadav, A. (2013). Of art and algorithms: Rethinking technology & creativity in the 21st century. *TechTrends, 57*(3), 11.

Yadav A., Mayfuekd C., Zhous N. Computational Thinking in Elementary and Secondary Teacher Education (2014). From https://dl.acm.org/doi/abs/10.1145/2576872

Yadav A., Mayfuekd C., Zhous N., Hambrusch S. Computational Thinking in elementary and Secondary Education (2014). Fromhttps://www.researchgate.net/publication/262350755

CULT-TIPS Project (2019-1-NL01-KA201-060488) Compendium on Computational Thinking Methods (IO1 – A1)

Segredo E., Valladares G., Leon C. (2017) Towards the Education of the Future: Computational Thinking as a Generative Learning, from https://www.researchgate.net/publication/318592554

Yadav A. Computational Thinking in Teacher Education (2017). SprongerLink. From https://link.springer.com/chapter/10.1007/978-3-319-52691-1_13

Yadav, Hai Hong & Chris Stephenson (2016) Computational Thinking for All: Pedagogical Approaches to Embedding 21st Century Problem Solving in K-12 Classrooms

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., Engelhardt, K. (2016). Developing computational thinking in compulsory education –Implications for policy and practice.

Yadav et al., (2018) Computational Thinking in K-12: In-service Teacher Perceptions of Computational Thinking: Foundations and Research Highlights.

Europe Code Week https://codeweek.eu