

CSpathshala: Computational Thinking Curriculum for K-12

- Vipul Shah



CSpathshala material is licensed under the Creative Commons Attribution 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/> or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

1. Preamble

A principal goal of school education is to prepare the child for not only participation in the world outside but also to interrogate it, and contribute to changing and shaping the world. Literacy and numeracy provide the means for doing this and hence modern states enact compulsory school education that achieves universal literacy and numeracy. Language, Mathematics, Science and Social studies form the core of school curricula world over, as they provide the basic means for studying the world, and in changing it.

Computing is ubiquitous and people from every walk of life will need to be familiar with computing in some form or the other. This will give rise to a huge demand for computing skills leading to a talent shortage unless computational thinking is introduced in schools right now. As we participate in the digital revolution, we need to train our children on skills for the digital age. We need to teach our children to be socially responsible citizens and train them to be creators, inventors and adopters of technology. It is therefore imperative to teach computing, which promotes problem solving, computational thinking and critical reasoning skills, at par with Mathematics and Sciences.

Although teaching computers has already been introduced, it focuses primarily on digital literacy and a bit of programming. Students are taught use of word processors and presentation applications as well as some programming language such as Logo, Scratch, C++ or Java. Computational thinking is wider in scope and involves understanding a problem, designing a solution and expressing it in a form that a human or a machine can execute. US and UK have already moved beyond teaching digital literacy through Computer Science Teachers Association ([CSTA](#)-US) and Computing at School ([CAS](#)-UK). Other countries such as France, Spain and South Africa too are in the process of adopting a national computing curriculum. If India does not transition to a computing based curriculum now, she will be left behind!

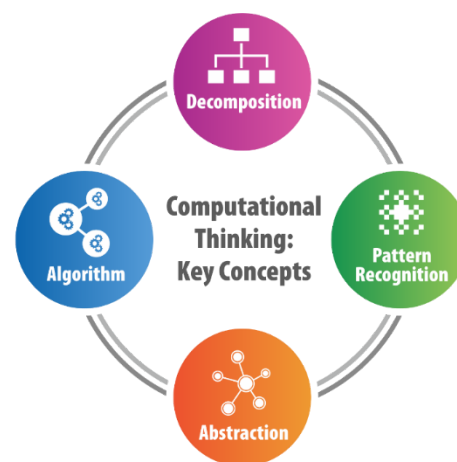
The Association for Computing Machinery ([ACM](#)) India has started a national education initiative, [CSpathshala](#), to develop a computing curriculum and initial sample material for K-12. CSpathshala has evolved the curriculum and contents by piloting these across schools around the country.

2. Acknowledgements

We are grateful to the curriculum committee comprising Dr. Rajeeva Karandikar (CMI), Dr. Madhavan Mukund (CMI), Dr. R Ramanujam (IMSc), Dr. Venkatesh Raman (IMSc), Dr. Ramesh Hariharan (IISC), Dr. K Narayan Kumar (CMI), R Venkatesh (TRDDC), Vipul Shah (TRDDC), Sonali Gogate, Nikhil Karkare (Walnut School) and Kumar Madhukar (TRDDC) for developing the curriculum. We also thank to Kumar Madhukar, Vipul Shah, Sonia Garcha (DST), Nikhil Karkare, Sonali Gogate, Arun Kadekodi (Soft Corner) as well teachers from many schools for inputs on implementation plan. This document has been put together based on various documents created and provided by the curriculum committee.

3. What is Computational Thinking?

Computational Thinking is the process of formulating a problem, finding solution to the problem and expressing it in such a way that humans or machines can understand the solution. It involves the use of problem solving methods to decompose the problem into smaller manageable sub problems, identifying the right abstractions so as to deal with scale and complexity, finding existing patterns or models that can be adapted, building an algorithm to solve the problem and in case of multiple solutions, analysing the solutions on multiple parameters to identify the one that best meets the given situation.



Computational thinking is an analytical thinking skill that draws on concepts from computer science but is a fundamental skill useful for everyone (Wing [2006]). Computational thinking as a practice is central to all sciences and not just computer science. There is hardly any profession where computers and computational methods are not used. New disciplines have come up in the past decades or so due to the advancement in computational methods. Computational thinking concepts are used in other disciplines including biology, health, retail, transportation, history, journalism, finance, and archaeology. Students who can think computationally are able to conceptualise, understand and use technology, helping them to be better prepared for the future.

It is said that by middle of 21st century Computational skills would be essential skills like reading, writing and arithmetic! This is why it becomes imperative that we teach Computational Thinking in schools.

4. Computational Thinking in K-12 Education

The National Research Council (NRC) (NRC [2010]) has highlighted the importance of exposing students to computational thinking notions early in their school years and helping them to understand when and how to apply these essential skills. Computer Science Teachers Association (CSTA <https://www.csteachers.org/>) and the International Society for Technology in Education (ISTE <https://www.iste.org/>) have presented a computational thinking framework for K-12 schools in 2011 with core computational thinking concepts and capabilities, including data collection, data analysis, data representation, problem decomposition, abstraction, algorithms and procedures, automation, parallelization, and simulation (CACM [2017]).

Algorithms are central to both computer science and computational thinking. Algorithms underlie the most basic tasks everyone performs, from following a simple cooking recipe to providing complicated driving directions. There is a general misconception that algorithms are used only to solve mathematical problems and are not applicable in other disciplines whereas there are enough examples from daily lives that require us to use algorithms. For example, the steps involved in brushing teeth, getting ready for school, steps performed during a lab experiment etc. Understanding and articulating algorithms as a sequence of precise steps helps us think logically.

Students can be exposed to the computational thinking concept of abstraction by creating models in physics (such as a model of the solar system). Abstraction helps students learn to strip away complexity and unnecessary detail to focus only on the important parts of the problem.

A number of leading research, educational, and funding organizations have argued for the need to introduce K-12 students to these core constructs and practices.

5. Teaching Computing without using computers

The primary goal of teaching Computational Thinking is to develop ability to solve problems and express the solution in the form of an algorithm that is precise and can be understood by a human or a machine. Experts feel that the above does not require the use of computers and can be taught as a series of interesting, engaging and fun activities. We want to convey that computing concepts and fundamentals do not depend on particular technology or software or programming languages. Technology has a short shelf life and will serve our children for the next few years, whereas, fundamentals will stay with them for several decades.

This avoids confusing Computer Science with programming or learning application software, makes the activities **available to those children who do not have access to computers**, and overcomes the hurdle of having to learn programming before being able to explore ideas and articulating solutions as algorithms. We want students to realize that they are capable of finding solutions to problems on their own, rather than being given a solution to apply to the problem. For example children play the game of Tic-Tac-Toe and then develop the rules of the game as well as winning positions.

6. Computational Thinking: Curriculum

The objectives of computing education in school can broadly be listed as follows:

- Acquiring an acceptable level of comfort with technology (like mobile phones and ATM machines) and tools (like word processing, numeric, internet and other software) in general,
- The use of computers for other subjects (like art, mathematics, sciences and geography),
- The ability to program a computer to solve a complex task that involves computational thinking skills like abstraction, decomposition, pattern recognition and solving in incremental steps.

Keeping the above objectives in mind, a national curriculum committee comprising computer scientists from top academic institutes as well as IT industry experts have explored CSTA K-12 curriculum framework and recommendations, CAS UK curriculum, code.org lessons, CS unplugged material and have developed a computing curriculum for CSpathshala.

The principles of computational thinking viz. decomposition, pattern recognition and generalization, algorithms and programming are explained in CSpathshala curriculum through the following themes that have been suggested in [8]:

- Systematic listing, counting and reasoning
- Iterative patterns and processes
- Information processing (Data)
- Discrete Mathematical Modelling
- Following and Devising Algorithms (Algorithms)
- Programming
- Digital Literacy

Students should use a variety of strategies to **systematically list and count** the number of different ways there are to complete a particular task. For example, junior school students should be able to count all items without missing any, make a list of all possible outcomes of a simple situation such as the number of outfits that can be worn using two coats and three hats. Middle school students should be able to systematically list and count the number of different four-block-high towers that can be built using blue and red blocks, or the number of possible routes from one location on a map to another, or the number of different "words" that can be made using five letters. Senior school students should be able to determine the number of possible orderings of an arbitrary number of objects and to describe procedures for listing and counting all such orderings.

An **Iterative patterns and process** involves repeating a single step or sequence of steps many times. For example, junior school students should understand patterns in daily life (weekdays, months, seasons, festivals), identify simple patterns in numbers and words, visual patterns like Rangoli. Middle school students should understand that multiplication corresponds to repeatedly adding the same number a specified number of times. They should explore repeating patterns called tessellations. Senior school students should understand how processes describing the change of physical, biological, and economic systems over time can be modeled by simple equations applied repetitively, and use these models to predict the long-term behavior of such systems.

Students should explore different methods of **arranging, organizing, analyzing, transforming, and communicating information**, and understand how these methods are used for information processing. Junior school students should investigate ways to represent and classify data according to attributes such as color or shape, and to organize data into structures like tables. Middle school students should be able to read, construct, and analyze tables, and maps. Senior school students should understand the application of methods to problems of information processing and computing such as sorting, codes, and error correction.

Discrete mathematical models such as graphs can be used to represent and solve a variety of problems based on real-world situations. Junior school students should recognize that a street map can be represented by a graph. Middle school students should be able to understand that routes can be represented by paths in the graph; and high school students should be able to find cost-effective ways of linking sites into a network using spanning trees.

Students should be able to **follow and devise lists of instructions, called "algorithms," and use them to find the best solution to real-world problems** - where "best" maybe, most cost-effective or most equitable. For example, junior school students should be able to carry out instructions for

getting from one location to another, should discuss different ways of dividing a pile of biscuits, and should determine the shortest path from one site to another on a map laid out on the classroom floor. Middle school students should be able to write precise instructions for adding two two-digit numbers, devise algorithms and express them in English, use flow charts to describe algorithms and simulate the algorithm to find possible outcome. Senior school students should be conversant with fundamental strategies of optimization, simple sorting and searching techniques.

Programming is a concrete implementation of the algorithm created to solve the given problem. Students in junior school should understand the programming principles: sequencing, conditional and loops using any fixed instruction set. In middle school, students should be able to write simple programs in languages like Scratch or Logo. In senior school, students should be able to implement simple algorithms they have studied in mathematics.

Digital Literacy: Junior school students should build general familiarity with computers. ICT can be integrated into other subjects to enable reports to be created using computers.

Technology is changing at rapid pace and algorithms are increasingly playing an important role in the world around us how the technology impacts our day to day life. Understanding this world as consumers and more so as creators of such technology will require a strong foundation in computational thinking. Providing an exposure to the science of data and information organization, and getting our children tuned to such thinking expands their mathematical abilities and prepares them better for the digital era. This is important for our children to eventually contribute to the information revolution, and not grow up only as its consumers [12].

A detailed syllabus for grades 1-8 is presented in the CSpathshala Syllabus document. CSpathshala has created 25 lessons for each grade. Each lesson has teaching aids in the form of a slide deck for the teacher to be used in the classroom, a lesson plan for the teacher that serves as a script for the slide deck and a worksheet for the students.

7. References

1. A. Tucker, F. Deek, J. Jones, D. McCowan, C. Stephenson, and A. Verno. 2006 A Model Curriculum for K-12 CS (2nd ed.).
2. <http://csta.acm.org/Curriculum/sub/CurrFiles/K-12ModelCurr2ndEd.pdf>
3. Jeannette M. Wing, Computational Thinking, Communications of the ACM, March 2006, Volume 49 (3) 33-35. <https://www.cs.cmu.edu/15110-s13/Wing06-ct.pdf>
4. CAS-UK. Computing at School Working Group <http://www.computingatschool.org.uk>
5. Computer Science Unplugged: csunplugged.org/
6. Sridhar Iyer, Farida Khan, Sahana Murthy, Vijayalakshmi Chitta, Malathy Baru and Usha Vishwanathan. 2013. CMC: A Model Computer Science Curriculum for K-12 Schools
7. code.org
8. New Jersey Mathematics Curriculum Framework
9. Noa Ragonis and Orit Hazzan. 2008. Disciplinary-Pedagogical Teacher Preparation for Pre-service Computer Science Teachers: Rational and Implementation. In Proceedings of the 3rd international conference on Informatics in Secondary Schools - Evolution and Perspectives: Informatics Education - Supporting Computational Thinking (ISSEP '08), Roland T. Mittermeir and Maciej M. Sysło (Eds.). Springer-Verlag, Berlin, Heidelberg, 253-264.
10. Raghu Raman, Smrithi Venkatasubramanian, Krishnashree Achuthan, and Prema Nedungadi. 2015. Computer Science (CS) Education in Indian Schools: Situation Analysis using Darmstadt Model. Trans. Comput. Educ. 15, 2, Article 7 (May 2015)
11. Department of School Education and Literacy Ministry of Human Resource Development Government of India, National Policy on Information and Communication Technology (ICT) In School Education 2012
http://mhrd.gov.in/sites/upload_files/mhrd/files/upload_document/revised_policy%20document%20of%20ICT.pdf
12. Tamil Nadu Curricular Framework 2017, A statement on Mathematics Curriculum