A Core Informatics Curriculum for Italian Compulsory Education

Luca Forlizzi, Michael Lodi, Violetta Lonati, Claudio Mirolo, Mattia Monga, Alberto Montresor, Anna Morpurgo, Enrico Nardelli

To cite this version:

Luca Forlizzi, Michael Lodi, Violetta Lonati, Claudio Mirolo, Mattia Monga, et al.. A Core Informatics Curriculum for Italian Compulsory Education. Informatics in Schools. Fundamentals of Computer Science and Software Engineering - 11th International Conference on Informatics in Schools: Situation, Evolution, and Perspectives, ISSEP 2018, Oct 2018, St. Petersburg, Russia. pp.141–153, 10.1007/978-3-030-02750-6_11 . hal-01913057

HAL Id: hal-01913057
https://hal.inria.fr/hal-01913057

Submitted on 6 Nov 2018

A Core Informatics Curriculum for Italian Compulsory Education

Luca Forlizzi[1][0000−0002−3923−7668], Michael Lodi[2][0000−0002−3330−3089],
Violetta Lonati[3][0000−0002−4722−244X], Claudio Mirolo[4][0000−0002−1462−8304],
Mattia Monga[3][0000−0003−4852−0067], Alberto Montresor[5][0000−0001−5820−8216],
Anna Morpurgo[3][0000−0003−0081−914X], and
Enrico Nardelli[6][0000−0001−9451−2899]

[1] Università degli Studi dell'Aquila, Italy, luca.forlizzi@univaq.it
[2] Alma Mater Studiorum - Università di Bologna & INRIA Focus, Italy, michael.lodi@unibo.it
[3] Università degli Studi di Milano, Italy, http://aladdin.di.unimi.it
[4] Università degli Studi di Udine, Italy, claudio.mirolo@uniud.it
[5] Università degli Studi di Trento, Italy, alberto.montresor@unitn.it
[6] Università degli Studi di Roma "Tor Vergata", Italy, nardelli@mat.uniroma2.it

Abstract. In order to bring informatics, its ideas and ways of thinking of major educational value to all primary and secondary school students, the Italian Inter-universities Consortium for Informatics (CINI), in collaboration with the academic associations who gather together researchers in informatics (GRIN) and computer engineering (GII), has recently proposed a core informatics curriculum for all the levels of compulsory school. This paper summarizes the proposed curriculum, highlights the key underlying motivations, and outlines a possible strategy to ensure that its implementation in schools can be effective.

Keywords: curriculum · compulsory education · non-vocational programs.

1 Introduction

Informatics[7] is no longer a subject area cultivated only by professionals, but is relevant to every citizen and should be part of general education since the earliest stages in order to develop "students' computational and critical thinking skills" and show "them how to create, not simply use, new technologies." Informatics "provides a fundamental set of concepts and skills needed to prepare students for the 21st century, regardless of their ultimate field of study or occupation."[8] As a matter of fact, the role of computing in school curricula is currently a topical issue of the education policies all over the world. In the US, for instance, the

[7] We adopt here the term "Informatics," more common in continental Europe, instead of "Computer Science" or "Computing"

[8] "Why K-12 computer science?"
http://code.org and http://computinginthecore.org

Computer Science Teachers Association (CSTA) in cooperation with the ACM have proposed comprehensive standards for K-12 education [19], and since 2015 the "Computer Science for All" initiative[9] puts school informatics on a par with other scientific and technological fields. In the UK, following the exhortations from the Royal Society Report "Shut Down or Restart" [22], Computing is a mandatory subject for all instruction levels starting from s.y. 2014–15 — see in particular England's "Computing at School" curriculum [9]. Similar démarches are under way in several other countries, although there is not yet full consensus as to what should be taught in K-12, and the associations Informatics Europe and ACM Europe have jointly put forth to the European Commission the "Informatics for All" proposal,[10] whose aim is to establish informatics as an essential discipline for students in Europe at all levels throughout the educational system. A broad picture of the state of CS education worldwide can be found in [16,1].

In this paper we present a recent proposal on behalf of our academic informatics community, which is meant to contribute to the development of informatics education in the Italian primary and secondary school [7]. It is the outcome of a long process, which has also benefited from important contributions of pedagogists and experienced school teachers who took part in the discussion. We are nevertheless aware that it is just a step in a longer path to make all people in charge of school policies aware of what is at stake for the students' future. As stated in the preamble to our document, informatics is having, and will continue to have, a growing impact on the development of production, economy, health, science, culture, entertainment, communication and society in general. In order to cope with the ubiquity of information technology, all citizens must acquire the conceptual tools necessary to understand the science underlying the digital world in which they are immersed and on which the quality of their life will depend. Although we are experiencing a rapid evolution of digital devices and of their applications, its scientific foundations are firm and rest on a homogeneous range of concepts, methodologies and skills. Informatics should then be considered as an independent knowledge field with its peculiar ways of thinking, of interpreting the world, of approaching problems. As Duchâteau pointed out [11], informatics is a relentless endeavor to disclose meaning from form and to confine meaning within form. The conceptual understanding of the scope of its tools revealed, according to Mazoyer [15], the "miracles" that combining a large number of times a small set of elementary operations can achieve a huge potential; that this potential is not specific to some particular type of operations; that the limits to this potential can be expressed and explored formally.

2 Context, process and background of the proposal

In the following subsections we briefly outline the Italian context of primary and secondary education; then we summarize the overall process which has led to the current version of the proposal [7]; finally we review its relevant background.

[9] http://cs4all.nyc

[10] http://www.informatics-europe.org/news/434-inf4all.html

2.1 The Italian school system

Starting from 2007, the Italian school system has undergone a broad reform process, aimed at renewing both the educational approach and the curricular organization. Compulsory education spans now over 10 years, usually corresponding to the age range 6–16, and is subdivided into three main stages: primary school (grades 1–5), lower secondary school (grades 6–8) and early upper secondary school (grades 9–10). In 2012 the Italian Ministry of Education, University and Research (MIUR) issued the curricular recommendations for the primary and lower secondary levels, that are common to all schools. As opposed to the previous instructional programmes, where the content of each subject area was mainly arranged in temporal sequence, the new framework aligns with the recent European trends in pre-tertiary education, focusing on skills and competences to be acquired in broad areas.[11] The upper secondary level, on the other hand, is characterized by a variety of strands, whose curricula are substantially differentiated from the outset, as appears from the list of specific documents. See [4] for a more comprehensive summary of the secondary school system in Italy. In the outlined picture, primary school teachers do not have subject-specific qualifications.

According to the current national curricular recommendations, computing-related topics and digital technologies should pertain to two rather broad areas:

– A cross-disciplinary key citizenship *digital competence* area:[12] proficiency and critical attitude in the use of ICTs for work, life, communication; use of computer to retrieve, assess, retain, produce, present, share information as well as to cooperate through the Internet.
– A general *technology* subject area (grades 1–8) or a specific informatics/IT-related subject (grades 9–10 for some types of schools, to be taught by qualified teachers), which partly overlaps with the above area, but *may* also include some computer and/or robot programming.

Moreover, with regard to the basic competences at the end of lower secondary education for the scientific-technological area, the national recommendations refer to some general awareness of the implications of using ICTs (for society, environment, health, etc.) and just add that "whenever possible, students can be introduced to simple and flexible programming languages in order to develop a taste for creation and for the accomplishment of projects [...] and in order to understand the relationships between source code and resulting behavior."[13]

However, the actual implementation of the curricular recommendations is to a large extent responsibility of each school, in accordance with the degree of autonomy introduced by the reform — autonomy that may occasionally be exploited by self-motivated teachers to propose valuable initiatives also in informatics education.

[11] In particular, the Italian Ministry has adopted the "Recommendation of the European Parliament and of the Council" of 18 December 2006 on key competences for lifelong learning (2006/962/EC).

[12] *Digital competence* is one of the seven broad areas listed in 2006/962/EC.

[13] http://www.indicazioninazionali.it/J (in Italian)

2.2 Writing and revision process

The proposal of a core informatics curriculum for all the levels of compulsory education is a recent initiative of the Italian Inter-universities Consortium for Informatics (CINI), promoted by its interest group for "Informatics and School", and carried out in cooperation with the academic associations who gather together researchers in informatics (GRIN) and computer engineering (GII). The process started in June 2017, when the assembly of the interest group charged an editorial board to write a first draft of the proposal. The draft was subjected to discussion and refinement during several distance meetings of the editorial board, held between June and early August. The document was then made available to all members of the CINI interest group for Informatics and School, who contributed some valuable feedback and were eventually able to reach an agreement on it in a meeting convened in September 2017. The next step was to publish the proposal in the form agreed upon by the interest group and to invite the whole informatics community (represented by CINI, GRIN and GII) to provide comments and suggestions. In the meanwhile, further contributions were collected through discussions with experienced teachers, pedagogists and other experts of school policies. By the end of October 2017 the editorial board examined the feedback from the informatics community and edited the current revision of the document. The revised proposal obtained an official status in November, after formal approval from the assemblies of CINI, GRIN, GII. Finally, in December 2017 it was presented within an initiative of the Italian Chamber of Deputies. For convenience, the proposal[14] has been organized in conformity with the competence-based model of MIUR documents reporting curricular recommendations. It attempts to introduce the general educational motivations as well as to explain as clearly as possible our community's cultural and scientific perspective. The discussion is currently continuing on practical issues, such as the need for instructional material and the involvement of in-service teachers. In March 2018 the document was also submitted to the chair of the MIUR Scientific Committee for the National Curricula for the primary school.

2.3 Background of the proposal

Informatics has recently been introduced in the official curricula for compulsory school of several countries [12]. Regardless of the approach or tools used, as pointed out by Hubwieser *et al.* [13], "there is a convergence towards computational thinking as a core idea of the K-12 curricula" and "programming in one form or another, seems to be absolutely necessary for a future oriented" informatics education. As far as the learning of programming is concerned, we should also note that its scope is now broader than it used to be. Indeed, practicing programming can be seen as a means of self-expression and social participation [14,18], as a component of a new form of literacy [6,24], as a way to widen

[14] An English translation of the official proposal [7] is avalaible at https://www.consorzio-cini.it/gdl-informatica-scuola

experience and experiment with personal ideas [5], and maybe also as an instrument to foster children's metacognition [17].

However, to set the background of the present proposal, we can mention, in particular, three noteworthy curricular models:

1. The CSTA/ACM K-12 standards [19], which identify five major strands: *computational thinking*, *collaboration*, *computing practice and programming*, *computers and communication devices*, and *community, global and ethical impacts*.

2. The English implementation of the new computing subject [10], structured into three components emanating from the Royal Society report [22]: *computer science*, *information technology,* and *digital literacy.*

3. The report of the French Academy of Science [8], which distinguishes between three ways of learning informatics, each appropriate for a different instruction level: *discovery* (primary school), *acquisition and autonomy* (lower secondary), and *mastering concepts* (high school).

3 A core informatics curriculum

Informatics at school is often misrepresented as mere use of digital technologies, but this is of course a distorted view. Its real educational value, both as an independent *scientific* discipline and as a cross-disciplinary field, lies in that it offers new and meaningful ways to interpret the world around us and to approach problems. The general term *Computational Thinking* (CT) [25,26] is commonly used to refer to such new ways of characterizing natural as well as artificial systems. However, students need an adequate informatics education to actually take a CT perspective as well as to fully participate in the digital society.

The proposed curriculum has been conceived in a two-dimensional framework. The former dimension, starting from grade 1 of primary school, is characterized by three main learning stages:

1. In the first stage (primary school) pupils are encouraged to *ask questions*, as well as to *discover* in their everyday life and to *explore* some basic ideas of informatics. They can be engaged either in *plugged*, i.e. implying the use of computing devices, or *unplugged* activities, i.e. without using digital technologies [2].
2. In the second stage (lower secondary school) students are expected to *grow in autonomy*. To achieve this educational objective, they have to learn more about the organization of data and the concept of algorithm. Moreover, they should be offered opportunities to develop *abstract thinking* and to acquire new specific as well as *cross-disciplinary* skills. In particular, programming tasks can play a key role in this respect.
3. The first two stages lay the foundations for mastering the concepts and for enhancing the skills at the core of the third stage (upper secondary school), at the end of which students should be able to *model problems* and to *design algorithms*. Abstraction, organization and accuracy are essential traits of the problem solving approach in the informatics field, that foster the development of critical thinking and provide helpful keys to master complexity.

The latter dimension concerns the content, which is organized into five key areas: *algorithms*, *programming*, *data and information*, *digital creativity* and *digital awareness*. Each such area will be the subject of a separate subsection. Overall, as reported in the list of general learning goals, at the end of compulsory school each student should be able:

- to understand and to apply basic concepts and principles of informatics;
- to approach problems by exploiting tools and methods of the field;
- to solve problems by devising formal representations, by designing algorithms and by coding the algorithms in a programming language;
- to evaluate the potential benefits as well as the limits of applying a range of digital technologies to achieve a given task;
- to use digital technologies in a conscious, responsible, confident, purposeful and creative way.

3.1 Area of algorithms

Algorithms are at the core of informatics. They predate programming, as several noteworthy algorithms have been designed well before the advent of computers. Pupils should meet the concept of algorithm since the early years of primary school, in an informal and playful way at first. Starting from the lower secondary school, the level of formality is progressively increased and the concept of algorithm is linked to other school subjects. By the end of the curriculum, pupils are expected to master the notion of algorithm and the related scientific concepts. To achieve these general goals, the curriculum tackles four main topics.

Algorithms as procedures. Pupils first encounter algorithms in grades 1-3, as a way to describe the procedures representing the activities of everyday life; *e.g.,* brushing one's teeth, dressing, leaving the classroom in an emergency drill. The initial approach could be unplugged, to later evolve into plugged activities, for example to solve coding puzzles. In grades 6–8, the collection of processes that are studied algorithmically is extended to include examples taken from other disciplines, such as mathematics, science and technology. Towards the end of the curriculum, pupils should know a selection of simple algorithms that solve fundamental informatics problems such as search and sorting. Apart from studying existing ones, pupils should progressively grow in autonomy and start to design their own algorithms, a skill that should be accomplished by the end of the curriculum.

Interpretation and disambiguation. The process of learning and designing algorithms should be accompanied by an increasing understanding that algorithms need to be described in a precise and unambiguous manner. In grades 1–5, this goal could be accomplished by having the pupils perform the role of the executor, in an unplugged way. In grades 6–8, the need of precision is reinforced, by making pupils reflect on the instructions performed by the automatic executor and how they are always completed in the same way. Pupils are thus expected to reflect on the ambiguities hidden in an algorithm described using natural language.

Decomposition. By grade 3, pupils should understand that difficult problems could be solved by breaking them down in smaller parts; by grade 5, such understanding should become operational, *i.e.,* pupils should be able to actually solve simple problems in such way. These concepts are later reinforced by the concept of modularity introduced in Section 3.2.

Reasoning about algorithms. During the second and third stage of the curriculum (lower and upper secondary school), pupils are introduced to a larger spectrum of issues related to algorithmics. They should move away from the concept of algorithms solving specific instances of a problem, and understand that algorithms should solve problems in their generality. By grade 8, pupils should be able to reflect on the correctness of their solutions, in particular by detecting and describing the conditions under which these processes can terminate. By grade 10, pupils should also be able to evaluate, in simple terms, the efficiency of basic algorithms and use logical reasoning to evaluate different algorithms that solve the same problem. They should also be able to understand that not all problems can be solved by algorithms in an efficient way.

3.2 Area of programming

Starting in the early grades pupils should get familiar with writing computer programs. In primary schools pupils write structurally simple programs, that possibly react to events, within a friendly — *e.g.,* visual — programming environment. By grade 8, pupils are expected to design, write and debug, using easy-to-use programming languages, programs that apply selection, loops, variables and elementary forms of input and output. By grade 10, pupils are expected to comply with syntax while writing simple programs in a textual programming language; moreover they should be able to define, implement and validate programs and systems that model or simulate simple physical systems or familiar processes, that occur in the real world or are studied in other disciplines. Overall, pupils should be able to operate on a program in order to understand its behaviour, modify it, identify and fix flaws. In primary schools, pupils first observe errors in programs and act spontaneously to correct them, then they start examining programs in order to detect and fix errors and should be able to use logical reasoning to understand why a simple program fails; by grade 8 they should intentionally experience small changes in a program to understand and modify its behaviour; by grade 10 they should recognize how the various parts of a program contribute to its functioning, and be able to predict the outcome of a program without running it. To achieve these general goals, the following specific skills should be progressively developed from grade 1 to 10.

Sequencing, selection and iteration. In early primary school pupils are expected to sort a sequence of instructions correctly, use one-way selection to make decisions within simple programs, and explore the use of two-way selection to implement mutually exclusive actions. The first use of loops in primary school is to concisely express that a certain action has to be repeated a given number of times; then loops can be used to repeat a certain action while an easy-to-test condition holds. By grade 8 pupils should be able to nest selection and loops as above, and start using variables in the conditions of selections and loops; by grade 10, they should be able to write conditions that use a logical operator, and use conditionals/selections within loops to describe the repetition of parametric actions.

Use of variables. In primary school, variables are used to represent input and output data, or to represent data computed during the execution of a program; by grade 8, simply typed variables are used to represent the state of a program and track the progress of the computation; by grade 10, students should write programs with structured variables, and be able to use variables in loops to define exit conditions or parametric actions.

Modularity. By grade 5, pupils recognize that a sequence of instructions can be considered as a single action subject to repetition or selection; by grade 8 they should be able to re-arrange a program to improve it, by organizing it in modular components as functions and procedures; by grade 10 they should design and develop modular programs using procedures and functions.

3.3 Area of data and information

The possibility of representing information through symbols, which can be stored and manipulated by an automatic processing system, lies at the very foundations of computing and then must be part of any informatics curriculum. Representing information is inherently connected to an abstraction process. Therefore concepts and methods in this area are acquired throughout the whole span of the curriculum, following the progression of pupils' abstraction abilities. Starting at grade 5, the curriculum aims at developing awareness that computers deal merely with raw data, encoded as symbols, and that information pertains only to the sphere of meaning, intrinsic to the human mind, what necessarily implies some degree of subjectivity. The main goals of the curriculum in this area can be classified by theme as follows.

Data Representation. Starting from grade 1, pupils gradually explore potential representations of various kinds of data (*e.g.,* numbers, images, sounds), using different formats, possibly even some of their own conception. By the end of grade 3, they should be able to select and use suitable items to represent simple data they are familiar with (*e.g.,* colors, words). At the end of grade 8, pupils should realize whether two alternative representations of the same data are interchangeable for a given purpose. The conventional character of any data representation, relative to what it is meant to describe, should be fully understood in grades 9–10. As a consequence, pupils become aware that different ways of representing data may affect both the effectiveness and the efficiency of a computation on such data. This achievement is also a prerequisite for the subsequent development of the ability to identify and choose the data representations best suited to an intended purpose.

Structure and organization of data. At grades 4–5 pupils start to represent simple structured data (*e.g.,* bitmap images) as well as, through combinations of symbols, a little more complex data familiar to them (*e.g.,* secondary colors, sentences). From grade 6 to 8 they develop the ability to classify data according to their kind (*e.g.,* numerical, textual), that leads to the data type concept. At the same time, they should learn to perform simple manipulations of symbols that represent structured data (*e.g.,* binary numbers, bitmap images), and to use structured variables to represent collections of homogeneous data (*e.g.,* vectors, lists). At the end of grade 10, pupils should know the features of basic data structures (*e.g.,* lists, vectors, matrices, dictionaries) and learn how to select an appropriate structure to approach a given problem.

Roles of data. The perception that data can be used in fundamentally dissimilar ways is to be developed in parallel with the programming skills, in particular those in connection with the use of variables. At the end of grade 8, pupils should be capable to distinguish the different roles played by the data within a program. Starting with the identification of input and output data, pupils should become familiar with the representation of the state of a computation carried out by a computer program. The idea of metadata should be introduced, in some specific context such as HTML or a simple data description language, in grades 9–10.

3.4 Area of digital creativity

Information Technologies (IT) are a very powerful means of self-expression and creativity. Starting from grade 1, pupils should become aware that they *can* use IT to express themselves, whereas too often they are just passive consumers of ready-to-use technological products and applications. As their ability to program improves, they are encouraged to engage with actively creating digital content and computer programs (in suitable environments), progressively using and combining different media, technologies and services. Moreover, they should start to reflect, to decide whether to use or not available technologies, and possibly to select appropriate technologies for different expressive purposes or to solve small problems they are personally interested in.

Use and creation of digital content. Using digital content and computer applications is just a first step: in grades 1–5, pupils should learn how to create simple and multimedia digital content; in addition, they start to select appropriate content, as well as to modify and combine it in simple ways. In grades 6–8 pupils should experiment with different ways of processing digital content (*e.g.,* bitmaps versus vectorial images), while learning how to effectively present it.

Active creation of software applications. From grade 3, pupils should start to create simple computer applications like stories, games, music, using environments designed for their age (*e.g.,* visual programming languages with blocks). In grades 6–8 they should be able to take advantage of their increasing experience with programming to create applications. In grades 8–10, pupils should use more advanced environments (*e.g.,* text-based programming languages) to create more elaborate content. Moreover, they should combine programming and on-line services to solve problems and to achieve tasks.

3.5 Area of digital awareness

Computer-based devices have pervaded everybody's life and it is important to develop awareness in pupils, since early years at school, with respect to their use and how they affect life and relations. This goal is pursued by our curriculum along two paths: a first one focusing on expanding pupils' knowledge of the information technology systems and devices, and a second one where students reflect, in increasing depth, on the personal and social impact of digital technologies.

Knowledge of Information Technology. In primary school pupils progress from recognizing the presence of IT devices all around them to being able to identify their main components and the main services they provide, while becoming aware of the importance of protecting personal data also in their various digital instantiations. In lower secondary school this knowledge is deepened with a comprehension of the main physical and functional principles at the core of computing systems and their communication networks, and through first experiences of interconnecting computer-based systems and input-output peripherals, and collecting and analyzing data. In the early upper secondary school pupils understand the importance of taking into account enduser requirements for the development of computer-based applications and deepen their experience with using computers to interact with and control external devices.

Social impact. In primary school pupils are progressively sensitized to the importance of interacting respectfully with others, even when using digital platforms, and to identifying and reporting problems in social interactions mediated by information technology. In lower secondary school they grow in their understanding of the value of data, both from a personal viewpoint and from a general one, and of how the collection and processing of large quantities of data affects society. In the early upper secondary school pupils arrive at critically reflecting on the multifaceted relations between information technology and society, spanning any domain of interest, and on the importance of keeping human beings in control of critical steps whenever computer-based decisions affect people.

4 The future of our proposal in the schools

In order to have a real impact on schools we need to face several challenges and operational difficulties. A major challenge is to cope with the general shortage of teachers with sufficient familiarity with the basic concepts of informatics, an issue emerging also in other countries [21,23]. Even in vocational schools it is common to find teachers with a poor background in the field [3] and, given the current state of the recruitment process, we can hardly hope that the situation will improve in the next few years. Thus, the teachers at all levels, but especially at the earliest grades, need support to cope with the need to reshape their teaching practice. A national effort is then required to identify and validate suitable instructional methodologies and learning materials that can support effective learning by distilling and formalizing what in essence amounts to the so called *Pedagogical Content Knowledge* — *i.e.,* the "the knowledge of teachers to help others learn", including "the ways of representing and formulating the subject that makes it comprehensible to others" [20]. We believe this could be achieved by designing and carrying out small-scale action-research projects in cooperation between school teachers and researchers in computer science education, as well as by documenting the results, strengths and weaknesses of a variety of teaching approaches. However, the next crucial steps to be successful in introducing informatics in the Italian school are to design a systematic way to disseminate best practices and to make available appropriate textbooks.

The recommendations outlined here may still sound a little utopian to anyone who knows the current state of the Italian schools. In fact, our main goal was to raise awareness among stakeholders and, in particular, policy makers by offering a comprehensive view of how informatics could be introduced in schools as well as by pointing out the potential benefits for every citizen of the 21st century. Our proposal has gained the endorsement of the reference informatics communities and is based on a sensible pedagogical analysis of its major themes. Moreover, as we have tried to explain in this paper, the informatics curriculum has been conceived as a whole, built up of strongly interconnected parts: the learning goals of later grades would appear much more plausible in light of the achievements expected at earlier stages. Finally, we are well aware that the strategy we envisage presupposes a strong political commitment towards informatics education in schools with a focus, beyond the hype of digital competences, on the scientific principles underlying the development of a digital society.

References

1. Barendsen, E., Mannila, L., Demo, B., Grgurina, N., Izu, C., Mirolo, C., Sentance, S., Settle, A., Stupurienė, G.: Concepts in K-9 computer science education. In: Proc. of the 2015 ITiCSE on Working Group Reports. pp. 85–116. ACM (2015)
2. Bell, T., Alexander, J., Freeman, I., Grimley, M.: Computer Science Unplugged: School students doing real computing without computers. The New Zealand Journal of Applied Computing and Information Technology **13**(1), 20–29 (2009)
3. Bellettini, C., Lonati, V., Malchiodi, D., Monga, M., Morpurgo, A., Pedersini, F.: La formazione degli insegnanti della classe 42/A–informatica: l'esperienza dell'università degli studi di milano. In: E questo tutti chiamano informatica: L'esperienza dei TFA nelle discipline informatiche, Collana Manuali, vol. 14, chap. 4, pp. 53–76. Sapienza Univ. Ed. (2015), in Italian
4. Bellettini, C., Lonati, V., Malchiodi, D., Monga, M., Morpurgo, A., Torelli, M., Zecca, L.: Informatics education in Italian secondary schools. Trans. Comput. Educ. **14**(2), 15:1–15:6 (2014)
5. Boyatt, R., Beynon, M., Beynon, M.: Ghosts of programming past, present and yet to come. In: du Boulay, B., Good, J. (eds.) Proc. of the 25th Annual Workshop of the Psychology of Programming Interest Group – PPIG 2014. pp. 171–182 (2014)
6. Burke, Q.: The markings of a new pencil: Introducing programming-as-writing in the middle school classroom. J. of Media Literacy Education **4**(2), 121–135 (2012)
7. CINI: Proposta di indicazioni nazionali per l'insegnamento dell'informatica nella scuola. https://www.consorzio-cini.it/index.php/it/component/attachments/download/745 (December 2017), in Italian

8. Committee on Science Education: Teaching computer science in France: Tomorrow can't wait. Tech. rep., Institut de France – Académie des Sciences (2013), http://www.academie-sciences.fr/en/Advice-Notes-and-Reports/teaching-computer-science-in-france-tomorrow-can-t-wait.html
9. Computing at School: Computer science: A curriculum for schools (2012), http://www.computingatschool.org.uk/data/uploads/ComputingCurric.pdf
10. Department for Education: National Curriculum for England: Computing programme of study. Tech. rep., Department for Education (2013), https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study/national-curriculum-in-england-computing-programmes-of-study
11. Duchâteau, C.: Peut-on définir une "culture informatique"? Journal de Réflexion sur l'Informatique (Institut d'Informatique, FUNDP, Namur) **23-24**, 34–39 (1992)
12. European Schoolnet: Computing our future: Computer programming and coding: Priorities, school curricula and initiatives across Europe – Update 2015 (2015)
13. Hubwieser, P., Armoni, M., Giannakos, M.N., Mittermeir, R.T.: Perspectives and visions of computer science education in primary and secondary (K-12) schools. Trans. on Computing Education **14**(2), 7:1–7:9 (2014)
14. Kafai, Y.B., Burke, Q.: The social turn in K-12 programming: Moving from computational thinking to computational participation. In: Proceeding of the 44th ACM Technical Symposium on Computer Science Education. pp. 603–608. ACM (2013)
15. Mazoyer, J.: Universalité de la notion de calcul. L'enseignement de l'informatique de la maternelle à la terminale, Académie des sciences – workshop (2005)
16. McCartney, R., Tenenberg, J., Eds.: Special issue on computing education in K-12 schools. Trans. on Computing Education **14**(2) (2014)
17. Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., Kafai, Y.: Scratch: programming for all. Comm. of the ACM **52**, 60–67 (2009)
18. Schulte, C.: Reflections on the role of programming in primary and secondary computing education. In: Proc. of the 8th Workshop in Primary and Secondary Computing Education. pp. 17–24. WiPSE '13, ACM, New York, NY, USA (2013)
19. Seehorn, D. (ed.): K-12 Computer Science Standards – Revised 2011: The CSTA Standards Task Force. ACM (2011), http://csta.acm.org/Curriculum/sub/K12Standards.html
20. Shulman, L.S.: Those who understand: Knowledge growth in teaching. Educational Researcher **15**(2), 4–14 (1986)
21. The Committee on European Computing Education (CECE): Informatics education in europe: Are we all in the same boat? (2017), http://www.informatics-europe.org/component/phocadownload/category/10-reports.html?download=60:cece-report
22. The Royal Society: Shut down or restart? The way forward for computing in UK schools. London: The Royal Society (2012)
23. The Royal Society: After the reboot: Computing education in UK schools. London: The Royal Society (2017)
24. Vee, A.: Understanding computer programming as a literacy. Literacy in Composition Studies **1**(2), 42–64 (2013)
25. Wing, J.M.: Computational thinking. Comm. of the ACM **49**(3), 33–35 (2006)
26. Wing, J.M.: Computational thinking: What and why? The Link Magazine (2011)