Proposal for a national Informatics curriculum in the Italian school

in cooperation with

Gruppo
Ingegneria Informatica

Gruppo Informatici

December 2017

(this translation August 2018)

ACKNOWLEDGMENTS

This document has been authored by: Enrico Nardelli (coordinator), Luca Forlizzi, Michael Lodi, Violetta Lonati, Claudio Mirolo, Mattia Monga, Alberto Montresor, Anna Morpurgo.

We acknowledge the contributions of: Giovanni Adorni, Franco Barbanera, Mario Bochicchio, Alessandro Bogliolo, Claudia Canali, Sara Capecchi, Antonella Carbonaro, Giorgio Casadei, Mario Coppo, Marco Corbatto Pierluigi Crescenzi, Antonina Dattolo, Renzo Davoli, Barbara Demo, Mariangiola Dezani, Gabriella Dodero, Gian-Luigi Ferrari, Filomena Ferrucci, Francesco Fontanella, Ilenia Fronza, Rosella Gennari, Ugo de' Liguoro, Simone Martini, Alfredo Milani, Umberto Nanni, Elena Pacetti, Gianluca Palermo, Susanna Pelagatti, Valentina Poggioni, Guido Proietti, Simona Ronchi della Rocca, Ugo Solitro, Damian Tamburri, Giorgio Ventre.

The Working Group "Informatics and School" (Gruppo di Lavoro "Informatica e Scuola") of **CINI** (Consorzio Interuniversitario Nazionale per l'Informatica – National Interuniversity Consortium for Informatics) is http://consorzio-cini.it/gdl-informatica-scuola

We thank the associations of university professors in Informatics (**GRIN** – Gruppo Informatici) and Informatics Enginering (**GII** – Gruppo Ingegneri Informatici).

Foreword

This proposal of the Italian academic Informatics[1] community aims at contributing to the development of Informatics education in the primary and secondary school in Italy.

The role of Informatics in school curricula is currently a topical issue of the education policy all over the world. In the US, the 2015 the "Computer Science for All" initiative puts Computer Science in schools on a par with other scientific and technological fields. In the UK, starting from s.y. 2014-15 Computing is a mandatory subject for all instruction levels. Similar démarches are under way in several other countries.

This document is the outcome of a long process, promoted by the Italian Informatics community, but that has also benefited from important contributions of pedagogists and experienced school teachers who took part in the discussion. For convenience, the proposal was drafted according to the model of related MIUR documents reporting curricular indications. In addition, it attempts to explain as clearly as possible our cultural and scientific standpoint.

On the one hand, according to our community's view, Informatics is an independent scientific discipline that provides the concepts and the languages necessary to understand and to fully participate in the digital society. On the other one, it is a cross-disciplinary field that offers an additional point of view to interpret phenomena and to approach problems.

We are nevertheless aware that the current outcome is just the first step of a long journey and that our task cannot yet be considered as accomplished. Further opportunities for discussion with teachers, experts in education and people in charge of school policies are still necessary, hence we are ready to continue our dialogue and cooperation with all of them.

[1] In this document we use the term "Informatics," more common in central Europe, instead of "Computer Science" or "Computing"

Preamble

Informatics, the science of computing, will have an ever growing impact on production, economy, health, science, culture, entertainment, communication and society in general. Several of the innovations we are currently witnessing can be ascribed to the remarkable advances in this discipline, which has gained autonomy as a science with its peculiar ways of thinking, of interpreting the world, as well as of approaching problems.

While we are experiencing a rapid evolution of digital devices and of their applications, the scientific foundations of Informatics are nevertheless firmly rested on a homogeneous range of concepts, methodologies and skills.

In order to cope with the ubiquity of information technology, all citizens must learn the foundations of Informatics and acquire the conceptual tools necessary to understand the logic and the processes underlying the digital world in which they are immersed and on which the quality of their life will depend.

Abstraction, planning 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 understand complex systems. Whatever their interests and future vocation, all students need to practice these competencies in order to be able to create with, and not merely make use of, the digital technologies.

The instructional program starts from primary school and is organized into three main learning stages.

In the first stage (primary school) students 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, possibly by drawing inspiration from the history of such ideas.

In the second stage (lower secondary school) students are expected to grow in autonomy. To achieve this educational goal, 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 competencies. In particular, programming tasks can play a key role in this respect.

The first two stages lay the foundations for mastering the concepts and for enhancing the competencies 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.

Whatever the school level, the teaching of Informatics can, by its nature, be approached through active learning methods, teamwork and laboratory activities (including "unplugged" activities). Programming projects, possibly integrating or re-using third-party products, as well as the reasoning to justify program correctness are crucial to the development of creativity, critical thinking, and therefore personal autonomy.

Informatics is often misrepresented as the mere use of digital technologies, but this is a distorted view. By contrast, advances in this knowledge field foster new and meaningful ways to observe, understand and act on the world around us. The general term "computational thinking" is commonly used to refer to these new ways of characterizing natural systems (e.g. living systems) as well as artificial systems (e.g. networks of social relations). To be able to take a thorough "computational thinking" perspective, students need an adequate Informatics education.

Data, information, computation, algorithm, computing machine and formal language are key unifying concepts of the discipline. Although such concepts were already known, to some extent, before the birth of Informatics as an independent field, they have been significantly clarified and deepened precisely because of the major role they play in Informatics.

In addition to its conceptual tools, Informatics provides us with a wide range of methodological tools that enable us to model and to master the complexity of the faced problems faced. This conceptual and methodological baggage is also fundamental for a purposeful and creative use of information technology.

Thus, essential aims of an Informatics curriculum are to develop students' ability: to collect, represent and organize data; to conceive algorithms; to model problems; to reduce the complexity of a problem by breaking it down into smaller parts that are easier to approach; to think at multiple levels of abstraction; to identify recurring patterns; to reuse available solutions for solving similar problems; to describe data, problems and solutions in abstract (artificial) languages.

As Duchâteau pointed out (1992), Informatics is a relentless endeavor to disclose meaning from form and to confine meaning within form. Moreover, the conceptual understanding of the scope of its tools revealed, according to Mazoyer (2005), 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 related limits can be expressed and understood formally.

The major goal of the curriculum is to give all students the opportunity to develop basic competences in informatics. In particular, at the end of compulsory school each student should be able:

- to understand and to apply basic concepts and principles of the field;
- to tackle problems by means of the tools and methods of Informatics;
- to analyze problems by devising formal representations, by designing algorithmic solutions 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, competent and creative way.

Primary school

Competence goals at the end of primary school

The student:

T-P-1. understands that an algorithm describes a procedure that can be automated in a precise and unambiguous manner

T-P-2. understands how an algorithm can be expresses by means of a program written using a programming language-

T-P-3. read and write structurally simple programs;

T-P-4. explain, using logical reasoning, why a structurally simple program achieves its goals or fails;

T-P-5. begins to recognize the difference between information and data

T-P-6. explores the possibility of representing data of various kinds (numbers, images, sounds, ...) using different formats, even arbitrarily chosen ones;

T-P-7. starts recognizing the presence of computers in technological devices of everyday life;

T-P-8. recognizes the Internet as a communication infrastructure, distinguishing it from its services (eg search engines, e-mail, WWW) and the contents transmitted;

T-P-9. understands the basic rules for the safe and socially responsible use of information technology;

T-P-10. uses information technology systems to choose and use digital content;

T-P-11. develops a positive attitude towards computer-based applications, recognizing their potential as tools for personal expression in everyday life.

Knowledge and skills at the end of the third grade of primary schoola

Area of Algorithms

O-P3-A-1. to recognize algorithmic aspects in routine operations of everday life: e.g., brushing one's teeth, dressing, leaving the classroom

O-P3-A-2. to understand that difficult problems can be solved by breaking them down in smaller parts

Area of Programming

O-P3-P-1. to notice errors in simple programs and act to correct them;

O-P3-P-2. to order the sequence of instructions correctly;

O-P3-P-3. to use loops to concisely express that a certain action has to be repeatedly executed a prefixed number of times;

O-P3-P-4. to use one-way selection to make decisions within simple programs.

Area of Data and Information

O-P3-D-1. to select and use objects to represent data one is familiar with (e.g. colors, words, ...)

O-P3-D-2. to define the interpretation of the objects employed to represent data (i.e., the legend)

Area of Digital Awareness

O-P3-N-1. to recognize the uses of informatics and digital technologies in everyday life;

O-P3-N-2. to understand the concept of private data and the need to keep them confidential;

O-P3-N-3. to understand the importance of respecting others when using digital technologies;

O-P3-N-4. to be able to ask for help in case of problems related to downloaded materials or to contacts in which one is involved with on the Internet or through other online technologies.

Area of Digital Creativity

O-P3-R-1. to create elementary digital content

O-P3-R-2. to select and to use digital content for expressive purposes, using computer-based applications and digital devices in a simple way.

Knowledge and skills at the end of the fifth grade of primary schoola

Area of Algorithms

O-P5-A-1. to use logical reasonsing to explain how simple algorithms work

O-P5-A-2. to solve complex problems by breaking them into smaller parts.

Area of Programming

O-P5-P-1. to examine the behavior of simple programs to understand and possibily correct them;

O-P5-P-2. to write loops to repeat a certain action while an easy-to-test condition stays true;

O-P5-P-3. to recognize that a sequence of instructions can be considered as a single action, which can be repeated or selected;

O-P5-P-4. to write simple programs that react to events;

O-P5-P-5. to explore the use of two-way selection to implement mutually exclusive actions within simple programs.

Area of Data and Information

O-P5-D-1. to use combinations of symbols to represent non-elementary data one is familiar with (e.g. secondary colors, sentences, ...);

O-P5-D-2. to use symbols to represent simple structured data (e.g. bitmap images)

Area of Digital Awareness

O-P5-N-1. to know the main hardware and software components of the devices one uses;

O-P5-N-2. to understand the distinction between the communication network and the services accessible through it;

O-P5-N-3. to understand how the privacy of digital data can be protected by "secret" codes;

O-P5-N-4. to recognize acceptable / unacceptable behavior in the use of information technology and of content obtained through it;

O-P5-N-5. to know how to report problems or concerns regarding content obtained or contacts established on the Internet.

Area of Digital Creativity

O-P5-R-1. to create simple multimedia content

O-P5-R-2. to create simple computer applications for expressive purposes (eg. stories, games, music, ...) using suitable environments;

O-P5-R-3. to select, modify and combine digital content for expressive purposes, using computer applications and IT in a simple way.

Lower Secondary School

Competence goals at the end of lower secondary school

The student:

T-M-1. understands the need for precision, so that instructions are always interpreted in the same way by an automatic executor;

T-M-2. algorithmically describes simple processes, such as those encountered in nature and everyday life, or those studied in other disciplines;

T-M-3. understands the importance and the need to reflect on the correctness of the algorithmic descriptions;understands the use of variables to represent data within the program;

T-M-4. designs, writes and debugs, using easy-to-use programming languages, programs that apply selection, loops, variables and elementary forms of input and output;

T-M-5. rearrange programs in order to improve them, by structuring them in modular components as functions and procedures;

T-M-6. recognizes input and output to computer-based applications

T-M-7. understands the different roles of data in a program: input, program state representation, output;

T-M-8. classifies data according to their nature and purpose;

T-M-9. knows the main (physical and functional) architectural principles of a computer-based system;

T-M-10. recognizes the hardware and software components of computer-based systems;

T-M-11. recognizes the fundamental mechanisms by which computer-based systems communicate and provide services on the Internet;

T-M-12. knows the appropriate / inappropriate safe / dangerous ways responsible / irresponsible to use computer-based technologyselects and uses, even in a combined way, computer programs and services to achieve a specific goal

T-M-13. experiments the potential of computer-based applications and digital devices as a tool for personal expression.

Knowledge and skills at the end of lower secondary school

Area of Algorithms

O-M-A-1. to detect the potential ambiguities hidden in the description of an algorithm when natural language is used;

O-M-A-2. to describe the algorithms according to the capabilities of the automatic executor and reflects on their correctness;

O-M-A-3. to write algorithms, even based on conventional notations, to describe simple processes inspired from the natural world, from the everyday life, or studied in other disciplines;

O-M-A-4. to detect and describe the conditions under which the above mentioned processes may terminate.

Area of Programming

O-M-P-1. to try small/simple changes in a program to understand and modify its behavior, identify and fix its flaws;

O-M-P-2. to write programs that use nested loops and selections;

O-M-P-3. to use in a simple way modular mechanisms, such as functions and procedures

O-M-P-4. to write programs using also typed variables of a simple kind;

O-M-P-5. to use variables that represent the state of the program and allow tracing the progress of computation;

O-M-P-6. to use variables in the conditions of loops and selections;

O-M-P-7. to re-organize programs to improve their comprehensibility.

Area of Data and Information

O-M-D-1. to recognize whether two representation of the same simple data are interchangeable for the intended purpose;

O-M-D-2. to perform simple manipulations of symbols that represent structured data (e.g. binary numbers, bitmap images);

O-M-D-3. to use variables to represent the state of a computation;

O-M-D-4. to use structured variables to represent collections of homogeneous data (e.g. vectors, lists, ...).

Area of Digital Awareness

O-M-N-1. to understand the main architectural and functional concepts of the Internet and the Web;

O-M-N-2. to understand the main architectural and functional concepts of computer-based systems and devices, distinguishing between hardware and software;

O-M-N-3. to use the most common computer-based systems and devices to organize and manage the data of interest;

O-M-N-4. to connect computer-based devices with each other and with peripheral devices, also with the purpose to realize simple experiences of data collection and analysis and of control of external devices;

O-M-N-5. to recognize the value of any personal data, not only of sensitive data, and be aware of issues related to identity on the network;

O-M-N-6. to understand the social risks connected to the systematic collection of data and the inherently public dimension of social networks;

O-M-N-7. to critically evaluate the content found on the web.

Area of Digital Creativity

O-M-R-1. to experiment during the creation of digital content various digital tools and multiple processing methods, so as to express themselves at their best;

O-M-R-2. to choose the most appropriate digital tools for their expressive goals;

O-M-R-3. to create software applications for expressive purposes (eg. stories, games, music, ...) using suitable environments;

O-M-R-4. to select and organize digital content for an effective presentation.

First Biennium of Higher Secondary School

Competence goals at the end of the first biennium of higher secondary school

The student:

T-S-1. understands the need to refer to the capabilities of an automatic executor in order to express algorithms in an unambigous way;

T-S-2. recognizes that algorithms are able to solve problems in their generality, and not for single instances;

T-S-3. is able to discuss the correctness of an algorithm with respect to the above mentioned generality;

T-S-4. understands the nature of the problems that are worth an algorithmic solution;

T-S-5. is able to evaluate the efficiency of simple algorithms;

T-S-6. defines, implements and validates 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;

T-S-7. understands when programming can provide a convenient way of tackling a problem;

T-S-8. understands the conventional nature of the representation chosen for the data, according to the described information,

T-S-9. recognizes that the way data are represented and organized affects the effectiveness and the efficiency of computation

T-S-10. selects and recognizes in computer programs the representations of problem data, obtained results and elements tracing the intermediate state of a computation;

T-S-11. recognizes the universal and multi-purpose nature of computer-based systems and understands the role of programs in transforming them into machines for specific purposes;

T-S-12. understands the importance of user needs for the implementation of computer-based applications;

T-S-13. is aware that the Internet and computer-based systems and devices influence the economy and the organization of society;

T-S-14. is aware that the diffusion and use of information technology has ethical and social consequences and learns to evaluate them critically;

T-S-15. selects, uses and combines computer programs and software services to develop well-structured informatics projects;

T-S-16. selects, combines and extends computational artifacts to express its own creativity

Knowledge and skills at the end of the first biennium of higher secondary school

Area of algorithms

O-S-A-1. to know a selection of simple algorithms that solve fundamental problems (for example, search, sorting, maximum common divisor, etc.);

O-S-A-2. to use logical reasoning to evaluate different algorithms that solve the same problem;

O-S-A-3. to understand that not all problems can be solved algorithmically;

O-S-A-4. to take into account, when designing an algorithm, the characteristics of the automatic executor and the limits of its resources;

Area of Programming

O-S-P-1. to recognize how the various parts of a program contribute to its functioning;

O-S-P-2. to predict the result of a program without running it;

O-S-P-3. to use conditions that use a logical operator;

O-S-P-4. to use loops with conditions to describe parametric actions;

O-S-P-5. to write programs using structured variables;

O-S-P-6. to design and develop modular programs using procedures and functions;

O-S-P-7. to write simple programs in a textual programming language, respecting their syntax;

Area of Data and Information

O-S-D-1. to evaluate the advantages and disadvantages of alternative representations of the same data

O-S-D-2. to know the characteristics of the fundamental data structures (eg: lists, vectors, matrices, dictionaries, ...) and to know how to select the most suitable one to accomplish the task at hand;

O-S-D-3. to recognize the difference between data and metadata in some simple context (eg HTML, simple data description languages, ...).

Area of Digital Awareness

O-S-N-1. to realize experiences of data collection and analysis through sensors and of control of external devices;

O-S-N-2. to take into account the requirements of end-users in the implementation of computer-based applications;

O-S-N-3. to identify if and how digital programs and contents can be reused, modified, disseminated;

O-S-N-4. to be aware of the multifaceted relations between the protection of the privacy of individual data and the protection of company security (eg anonymity in the network, ...);

O-S-N-5. to evaluate the reliability of content found on the web, examining and double checking sources

Area of Digital Creativity

O-S-R-1. to use programming environments for expressive purposes (eg animations, sound tracks, games, ...);

O-S-R-2. to combine programming and online services to achieve its own goals