# Introducing a New Computer Science Curriculum for All School Levels in Poland

Maciej M. Sysło[1,2] and Anna Beata Kwiatkowska[1]

[1] Faculty of Mathematics and Computer Science, Nicolaus Copernicus University
Chopin str. 12/18, 87-100 Toruń, Poland
syslo@mat.umk.pl, aba@mat.umk.pl

[2] Faculty of Mathematics and Computer Science, University of Wrocław
F. Joliot-Curie str. 15, 50-383 Wrocław, Poland
syslo@ii.uni.wroc.pl

**Abstract.** The first regular informatics lessons in schools in Poland were organised in the second half of the 1960'. Some of them were devoted to programming a mainframe computer (in Wrocław) and some to theoretical models of computers and computations (in Warsaw). Then, for more than last 30 years of formal informatics education in Poland we have been very successful in keeping informatics (as computer science) as a stand-alone subject and in shaping its curriculum according to high standards of the discipline. In this paper, in Section 1 we first discuss terminology related to computers in education and then report on early history of informatics education in Poland. In Section 2, the present curriculum of informatics subjects is described in details together with some comments on using computational thinking in its implementation. Then, as the main contribution of this paper we introduce in Section 3 a new computer science curriculum for all school levels in Poland. To this end, the existing curricula of informatics subjects have been remodeled, extended (e.g. by adding programming to each level), and unified according to the five Unified aims of learning computing. The new curriculum benefits very much from our prior curricula and experience. Finally we discuss some implementation details, supporting activities, and the road map for a successful introduction of the curriculum to all schools.

**Keywords:** education, informatics, computer science, algorithmic thinking, computational thinking, programming.

## 1 Introduction

### 1.1 Terminology

In education, as in the most of other disciplines and areas of computer applications, the terms: computer science, computing, IT (Information Technology), ICT (Information and Communication Technology), and informatics usually bear popular understanding of the discipline related to computers and also to the wide range of technologies. The meaning of these terms in education is formally defined based on the context of their use in the curriculum statements.


A. Brodnik and J. Vahrenhold (Eds.): ISSEP 2015, LNCS 9378, pp. 141–154, 2015.
DOI: 10.1007/978-3-319-25396-1_13

The first regular lessons related to 'computers' were held in Poland in the second half of the 1960' when the terms 'computer' and 'informatics' had not been used yet and a computer was a 'mathematical machine', see [19].

In the first informatics syllabus for schools (PTI, 1985) the school subject was called Elements of informatics.

The first national curriculum appeared in the second half of the 1990'. It contained curricula for informatics subjects and the term 'information technology' appeared for the first time as "the combination of informatics technology with other, related technologies, specifically communication technology" (UNESCO/IFIP). Moreover, computers and information technology appeared as educational technology in the curricula of some other subjects. At that time a new term was coined and became popular – 'informatics education' which embraces stand-alone informatics subjects and the use of technology in other subjects.

A word on 'computing' is in place here. In the IEEE/ACM Computing Curriculum 2001 this term embraced 'computer science', 'software engineering', 'information systems', 'information technology'. Since then, computing has changed dramatically and its scope has broadened so much that it is not a single discipline. However it is used in [6] as 'computing education' in a similar sense we use 'informatics education'. This term is also used in the National Curriculum in England (2014) [4] and it covers both areas, computer science and information and communication technology: "The core of computing is computer science ... Computing also ensures that pupils become digitally literate – able to use, and express themselves and develop their ideas through information and communication technology". In Poland, the term 'computing' has no official counterpart – 'informatics' is used for almost everything what is related to computers. However the educational circles popularize the term 'computational thinking', its meaning and importance for the future education of our students, since it has appeared in [23], see also [15].

Recently, when working on a new core curriculum for computer science, to distinguish between any use of computers within informatics education and classes on rigorous computer science we have introduced a term 'computer science education', although school subjects are still called informatics, see Section 3.

Concluding comments on computer terminology in education in Poland today: 'informatics education' refers to any use of computers, informatics, and information and communication technology in other non-informatics subjects as educational tools and methods, and 'computer science education' refers to rigorous learning and teaching of computer science and it also contributes to general informatics education.

### 1.2 Early History of Computers in Education in Poland

Two initiatives addressed to high school students appeared in two academic centers in Poland, in Wrocław and in Warsaw in the 1960' – see [19] for details.

In 1964/1965, academic teachers from the University of Wrocław initiated informatics classes in two high schools and the subject was called Numerical methods and programming. High school students were writing programs in Algol

60 for numerical calculations during school hours and then ran them on the Elliott 803 computer (made in the UK) installed at the University.

On the other hand in Warsaw, the informatics lessons, which started in 1970, were run in two university mathematics classes in high schools and were devoted mainly to some theoretical foundations of informatics – models of computers and computations. Other initiatives born in Warsaw in the 1970' were: an informatics syllabus for schools and in-service courses for mathematics teachers on teaching informatics in high schools. In the second half of the 1970', informatics was taught in about 1000 high schools in Poland.

It is worth mentioning that the first informatics lessons in schools in Poland were initiated by leading mathematicians and computer scientists of their time working in academic institutions. Teachers and instructors in schools and students involved in the first informatics classes admit today a great concern and engagement of the initiators of those first classes – professors Stefan Paszkowski in Wrocław and Hanna Szmuszkowicz and Zdzisław Pawlak in Warsaw.

The first informatics syllabus was proposed by the Polish Informatics Society (PTI) and then approved by the Ministry of Education in 1985. The school subject was called Elements of informatics. The curriculum covered the topics related to the use of microcomputer applications (for text editing, creating graphics and sounds, building tables and simple databases, making simulations) and elements of structural programming using Logo mainly for drawing pictures and operations on lists of characters.

For the next 10 years (micro)computers were mainly used in teaching informatics as a stand-alone subject and only occasionally they were used for supporting teaching and learning other subjects. Then the development of user-friendly human-computer interfaces and the Internet begun to influence the way computers were used in schools. In the mid 1990', the term 'information technology', as 'informatics for all students', was accepted by the education policy makers in Poland. The emphasis in education has moved from computer science to information technology, from constructing computer solutions to using ready-made tools, from computer science for some students to information technology for all. In [15] we demonstrated however that learning information technology can enhance algorithmic and computational thinking skills in solving with computers problems which arise in various school subjects and other areas.

In the beginning of this century, the national core curriculum contained the following compulsory subjects: informatics (in 4-6 grades in primary school – 2 hours per week for one year or 1 hour per week for two years), informatics (in middle school – 2 hours per week for one year or 1 hour per week for two years), information technology (in high school – 2 hours per week for one year), and an elective subject informatics (in high schools – 3 hours per week for two years). students could also take an external final examination (matura in Polish) in informatics.

In our approach to informatics education we make a general assumption that informatics (= computer science) deals mainly with creating 'new products' related to computers (such as hardware, computer tools, programs and software, algorithms, concepts, theories, etc.) and information technology is mainly using 'informatics (computer related) products'. Although this distinction does not define either informatics or information technology, it is very useful in describing the scope and methodology of learning and teaching both subjects. It is quite important for students' achievement that information technology, especially its sophisticated tools, may be also used to create highly innovative and involved computer products. However, their novelty and ingenuity contribute to the discipline to which they belong, rather than to computer science.

Regarding information technology [9], [10] we convince our students to elaborate her/his style of working with information. Application software programs, such as editors (text and graphics), spreadsheets, presentation programs, usually have several options which support a user in improving her/his style (e.g., styles, templates, wizards, etc.). Elements of style are also very important when working with information on the Internet, and in searching, publishing and communicating on the web.

In teaching informatics (computer science), an algorithmic problem solving approach is suggested for the systematic development of a computer solution for a problem, which covers the entire process of designing and implementing the solution. This methodology is aimed at generating good solutions, characterised by three fundamental properties: readability (the solution is understandable to anyone who is familiar with the problem domain and computer tools used), correctness (the solution satisfies the problem specification), and efficiency (the solution doesn't waste computing resources, such as time and space). The methodology consists of six stages: (1) Analysis of a problem situation. (2) Development of a specification of the problem. (3) Designing a computer solution of the problem. (4) Coding. (5) Testing a computer solution. (6) Presentation and discussion, see [15] for details. These 6 stages of the algorithmic approach are functionally very similar to the stages in the operational definition of computational thinking, developed recently, see [7], [17].

# 2 Informatics Education in Poland Today

## 2.1 Informatics Education

For a long time formal education in Poland started at the age of 7, which has recently been lowered to 6. Since 1999 the school system at the primary and secondary levels has consisted of three stages:

- primary school – 0-6 grades (age 6 to 13);
- middle school (in Polish: gimnazjum) – 7-9 grades (age 13 to 16);
- high school – 10-12 grades (to 13 in vocational schools) – (age 16 to 19).

We describe here in more details the present curriculum of stand-alone informatics subjects approved at the end of 2008 and introduced to primary schools (1-3) and to middle schools in 2008 and to primary schools (4-6) and to high schools in 2012. The new curriculum described in Section 3 is in some parts extension and modification of the present curriculum statements towards replacing activities within information technology by learning rigorous computer science, including programming.

*Primary Schools*

In primary schools a stand-alone informatics subject is now called **computer activities** and runs through grades 1 to 6. In grades 1-3, computer activities are supposed to be fully integrated with other activities like reading, writing, calculating, drawing, playing etc. At the next stages of education students are expected to use computers as tools supporting learning of various subjects and disciplines, formal, non-formal, and incidental in school and at home

Computer activities in grades 4-6 lay down solid knowledge and skills within information and communication technology to be used at the next levels.

*Middle Schools*

Informatics in middle schools is taught for at least 2 hours per week for one year or one hour per week for two years. The curriculum contains a section on algorithmics, algorithmic thinking and solving problems with computers. Although programming is not included in the curriculum, an introduction to Logo or to another programming language is a popular practice in some schools. Within algorithmics, students are expected to be able to: explain what an algorithm is, provide a formal description (specification) of a simple problem situation and propose an algorithm for its solution; use spreadsheets to solve simple algorithmic problems (e.g. the change making problem); describe, how to find an element in an ordered or an unordered sequence of elements; use a simple sorting algorithm (e.g. by counting); run some algorithms on a computer – either writing a program, using a spreadsheets or running an education software.

Informatics in middle schools is supposed to introduce basic elements of informatics, as computer science, important for at least two reasons: as a starting point for informatics education of all students in high schools and as a pre-orientation for those students who might be interested in choosing a high school which offers a specialization in computer science.

The implementation of the curriculum of informatics in middle schools has some undesirable features – most of the teachers admit that they have no time to cover algorithmic topics. However the truth is that the teachers are afraid of these topics since they are not enough confident in their algorithmic knowledge and skills to touch algorithmics with students who quite often have some experience in programming and running their own programs.

*High Schools*

In the present curriculum for high schools information technology disappeared as a stand-alone subject and informatics has been introduced in its place, for at least 1 hour per week for one year. In this way, informatics for all students in middle schools has been extended to high schools. This new subject is a continuation of informatics from middle school in the area of problem solving and decision making with a computer by applying algorithmic approach and also may serve as a pre-orientation, intended to prepare students for their choice of future study (e.g. informatics as an elective subject), career and jobs in computing related disciplines and areas.

Informatics (understood as a rigorous computer science) remains in high schools as an elective subject and is taught only in some schools 3 hours per week for two years. Students may also take an external final examination (matura in Polish) in informatics.

### 2.2 Computational Thinking

Since the first informatics lessons in Polish schools in the mid 1960', algorithmic thinking has been the main approach for problem solving and systematic development of computer solutions of problems coming mostly from computer science and its applications.

A much wider view on computing competencies has been proposed by Jeannette Wing in her seminal paper on computational thinking [23]. Earlier, one of the EU directives suggested that traditional skills for everyone known as the 3Rs (i.e. **r**eading, w**r**iting and a**r**ithmetic) should be extended to 3R+TI by adding skills in applying information technology. Wing has taken this step further by extending algorithmic thinking and fluency with information technology to competencies for all learners which can be used across disciplines as a computing methodology for solving problems and improve understanding of the role of computing in the modern society. Moreover, computational thinking may also "inspire the public's interest in the intellectual adventure of the field of computer science" [23] and as a result may also encourage and motivate students to consider a future career in computer science related disciplines.

Computational thinking includes a range of mental tools that reflect the breadth of computer science, for example, reduction and decomposition of a complex problem in order to solve it efficiently, approximation when an exact solution is beyond the reach of the computer, recursion as a method of inductive thinking and its computer implementation, representation and modelling some aspects of a problem to make it tractable, and heuristic reasoning to develop a solution of intractable problems.

One can observe the influence of computational thinking on other disciplines. On the other hand, computer scientists' interest in other disciplines is driven by their belief that other scientists can benefit from computational thinking. For instance, in mathematics, as formulated by R.W. Hamming in 1959, "the purpose of computing is insight, not numbers" – note, there were only a few computers in 1959. In [21] we suggest how to extend and enrich traditional topics in school mathematics by applying computational thinking to obtain solutions which are supported by the power of computer science as a discipline and computers as computing tools. Moreover, our approach to deal with topics in mathematics with computational thinking and computing tools contributes to constructionist learning, to learning by doing and making meaningful objects in the real world – here computer solutions. Mental tools used for this purpose include: data representations, reductive thinking, approximation of numerical and intractable problems, recursive and logarithmic thinking, heuristics.

We have adopted computational thinking (see [15]) as the main learning and teaching methodology for information technology as a school subject when it was compulsory for all high school students. A similar approach has been also used in some outreach activities [16] aimed at a better preparation of school students for their future decisions to study informatics related disciplines and to encourage them to consider a future career in computing.

The main difference between using information technology and thinking computationally is in going beyond using information technology tools and information towards creating tools and information. It reminds our distinction between informatics (as creation of programs, computers, theories, etc.) and information technology (as applying informatics tools), see Sections 1.2. The creation of tools (e.g. programs) and new information requires thinking processes about how to use abstraction and manipulate data and many other computer science and computing concepts, ideas, and mental tools of computational thinking.

## 3 A New Computer Science Curriculum

In this Section we report on an initiative[1] to revise the curricula of informatics subjects (decribed briefly in Section 2) in the Polish National Curriculum so that computer science education will cover all school levels in K-12. Fortunately our job was only to redefine the structure and content of the subject curricula since informatics subjects already cover all school levels in the present curriculum.

In the last decade, several national initiatives have been taken to provide relevant computer science education in schools, for instance in UK [4] and USA [6] – on all levels of K-12 education, and in Denmark [3] and New Zealand [2] only in high schools. Our proposal benefits from these and other initiatives as well as from general considerations such as in [22] and in [12].

In Subsection 3.1 we shortly discuss motivations for our initiative, Subsection 3.2 describes the developed computer science curriculum, and finally in Section 3.3 we comment on some aspects of the new curriculum and activities which will support its implementation in schools.

In what follows when 'informatics' is used it stands for 'computer science'.

### 3.1 Is Computer Science Education in Crisis?

In the 1980'-1990' computer science was confused with computer programming and, as a result, there was a strong opposition among education policy makers and parents to teaching computer science. They argued that only a few high school graduates would choose a career as a programmer. Even today, many

[1] The initiative has been developed by the Council for Informatization of Education, the Ministry of National Education. The authors are members of this Council.

people, among them also teachers and academics, do not consider computer science as an independent science and, therefore, as an independent school subject. Most of them confuse computer science and information technology.

Informatics education in school has not cleared up the myths about computer science and it is again confused with computer programming which has recently become easy accessible even for novice programmers. Students can easily access high-level tools for designing and producing complex applications without any knowledge of fundamentals, such as logic, discrete mathematics, programming methodology, or computability.

Since most students are fluent in using computers to play, search the web and communicate they have no interest in pursuing computer science as a career choice. One of the goals of computer science education should be to motivate students to go 'beyond the screen' and investigate how computers work and how software is designed so they can create their own solutions. Computer science lessons should prepare students for further study instead of leaving them satisfied with the knowledge and skills they have already learned.

The society needs a continuous inflow of good students to be educated and trained as professional specialists for computer related jobs in order to sustain the development and achievements that are necessary to meet the expectations of the information society and its citizens.

The White Paper by the CSTA [13] lists a number of challenges and requirements that must be met if we want to succeed in improving computer science education – our new curriculum has been designed to meet these challenges:

- students should acquire a broad overview of the field of computer science;
- instruction should focus on problem solving and algorithmic (computational) thinking;
- computer science should be taught independently of specific application software, programming languages, and environments;
- computer science should be taught using real-world problem situations;
- computer science education should provide a solid background for the professional use of computers in other disciplines.

### 3.2 The New Computer Science Curriculum

The new computer science curriculum benefits very much from our experience in teaching informatics in schools for more than 30 years (see Sections 1 and 2). In particular, it takes from the present curriculum the hours assigned to informatics subjects and unifies the names of all stand-alone informatics subjects as informatics. Therefore, according to the new curriculum, informatics is a compulsory subject in primary schools (1-6 grades, 1 hour a week for 6 years), middle schools (7-9 grades, 1 hour a week for two years), and high schools (10 grade, 1 hour a week). Moreover, informatics is also an elective subject in high schools (11-12 grades, 3 hours a week for two years) and high school students may graduate in informatics taking the final examination (pl. *matura*) in informatics.

We have been very lucky that the present curriculum already includes informatics subjects on each education level and we have only had to modify and extend their curricula. The same applies to the hours of instruction. Needless to say that otherwise it would be very difficult to impossible to convince the education authorities that the national curriculum needs such changes in the area which is not of the highest priority on the official agenda, unfortunately.

The new computer science curriculum begins with an introduction explaining the importance of computer science for our society in general and for our school students in particular (see Section 3.1 for some general comments). Then follow the curricula for each level of education. Each curricula consists of three parts:

- Second part is the same in all curricula. It includes **Unified aims** which define five knowledge areas in the form of general requirements – see below.
- First part is a description of **Purpose of study**, formulated adequately to the school level.
- The third part consists of detailed **Attainment targets**. The targets grouped according to their aims define the content of each aim adequately to the school level. Thus learning objectives are defined that identify the specific computer science concepts and skills students should learn and achieve in a spiral fashion through the four levels of their education.

The **Unified aims** are as follows:

1. Understanding and analysis of problems – logical and abstract thinking; algorithmic thinking, algorithms and representation of information;
2. Programing and problem solving by using computers and other digital devices – designing and programming algorithms; organizing, searching and sharing information; utilizing computer applications;
3. Using computers, digital devices, and computer networks – principles of functioning of computers, digital devices, and computer networks; performing calculations and executing programs;
4. Developing social competences – communication and cooperation, in particular in virtual environments; project based learning; taking various roles in group projects.
5. Observing law and security principles and regulations – respecting privacy of personal information, intellectual property, data security, netiquette, and social norms; positive and negative impact of technology on culture, social life and security.

Two very important comments regarding the new computer science curriculum are in order. Although any curriculum defines the aims and targets to be included in any school syllabus, the curricula for particular school levels in the new curriculum contain some optional attainment targets which can be freely added to a subject syllabus or assigned only to a group of students. This is a novelty in our national curriculum and leaves some room for **personalized learning** of gifted students as well as students who have particular interests in specific areas of computer science and its applications (such as mathematics, science, arts).

Personalization in the new curriculum is a means to encourage and motivate students to make **personal choices** of a range of computer science topics and areas in middle and high schools what may lead them towards computer science specialization in the next steps of education and in professional career. Personalization is aimed at increasing students' interests in learning then in studying computer science as a discipline, or at least in better understanding how computers and their tools work and can be used in solving problems which may occur in various areas.

Another facet of personalization is a curriculum of compulsory informatics in vocational high schools for computer, electronic, and electric technicians which is fully devoted to learning computer programming.

It should be noted, that as in the UK curriculum [4], our curriculum recognises the value of computer science as the underlying academic discipline and expects students to understand and use the basic concepts and principles of computer science, analyse and solve problems computationally, programming their solution, on one side (see Unified aims No 1 and 2), and on the other side, students are still expected to apply information technology and to be competent, creative, and responsible users of technology in other school subjects, disciplines, and areas of computer applications (see Unified aims No 2 and 5).

### 3.3 Implementation Comments, Supporting Activities

**Role of Programming.** From one hand, A. Perlis wrote in 1962 that everyone should learn to program and Mark Prensky declared a few years ago that "The True 21st Century Literacy Is Programming". From the other hand, we should avoid 'the equation': computer science = programming which is accused of killing interest in computer science among school students in 1990'. Not all students will become professional programmers but writing own programs, individually or in a group, they practice creative and computational thinking, and gain skills of the digital era useful for professional and personal life. They should also get some experience in programming other digital instruments, such as toys, robots, vending machines.

Traditionally, a programming language is a language of computer science in a sense that it is a tool for expressing algorithms and communicate them to computers and also to other programmers. However, we should remember that "informatics should be taught independently of specific application software and programming languages and environments", [13].

In [15] and [17] we have extended the meaning of the terms 'program' and 'programming' to see them in a wider context of using computers to solve problems which are not necessarily algorithmic in nature and introducing all students to computational thinking. There are plenty of opportunities to communicate with a computer by means of programs which are created by other programs, not necessarily writing own programs. The following objects are computer programs: spreadsheet, data base, interactive and dynamic presentation, website, and also documents and graphics and they can be used to 'program' a computer. This meaning of programming with no use of a programming language has a psychological advantage over programming in the traditional sense since the majority of students and their parents consider learning a programming language as the first step to a computer science career, but our goal is only to expose all students to computational thinking. In general, computational thinking is not equivalent to thinking process which leads to computer programming.

The above comments put programming in a right position in computer science education in schools – it is not the only way to communicate with computers and to use them for problem solving. However computer programming can enhance students' problem solving skills in a constructivist way by constructing programs as 'real objects' and then use them in learning by doing (solving problems, making experiments with data, verifying hypothesis, proving statements).

Implementing the curriculum we advise to use programming language for visual programming with novice programmers in primary schools and then to switch for textual language in middle and high schools. No particular language is recommended – the choice is left to teachers and students.

**Computer Science Unplugged.** Computer science unplugged (CS Unplugged) activities introduce children[2] to fundamental computing concepts and do not require a computer, see [5]. Such activities promote creativity, problem solving skills, and cooperation in groups.

We have extended this approach by introducing some computer activities and integrating them with what children are doing and used it at a children's university to introduce very young students to some concepts in computer science, see [20]. Children work in a number of environments which consist of two stages: first they are engaged in cooperative games and puzzles that use concrete objects (like in CS unplugged), and then they move to computational thinking about the objects and about the concepts they are learning. In this way we introduce our young students to such computer science concepts as: calculations using mechanical tools, complexity issues (the Tower of Hanoi, Fibonacci numbers, and binary search), and graph models (of real world situations). Our approach contributes to constructionist learning, to learning by doing and making meaningful objects in the real world, computational models of real-world situations. Our learning environments are extensions of 'unplugged' ones by encouraging children to purposely and properly use computers for certain activities.

**Integration of Computer Science with Other Subjects.** We have reviewed the curricula for all other school subject on all school levels and all topics (attainment targets) which are appropriate for augmenting by including and using computer science concepts, skills, and mental tools, have been annotated with

[2] The first author of this paper used the CS Unplugged approach in 1970' when introducing the concept of a stable marriage or pairing (due to Lloyd Shapley, the Nobel Prize winner in Economics in 2012) to university students by 'playing' the algorithm with groups of students.

comments how to apply computational thinking to enhance knowledge and skills in the other subjects. We expect a cooperation with teachers of other subjects in implementing our ideas how to integrate computer science with other areas.

**Teacher Preparation: Standards, Training, and Evaluation.** Preparation of teachers is a crucial factor for the success of introducing the new computer science curriculum to classrooms. The computer science education standards for teacher preparation, which are similar to the ISTE standards [11], have been developed. This standards are operational and include what teachers should be able to do to inspire, motivate and engage students and to promote students' ability to learn effectively. Moreover, the new standards focus also on teachers' engagement in professional development – candidate teachers may come from various pedagogical and subject areas and they may need personalized professional development and training, see [8].

We have also developed a certification procedure, which evaluates in the classroom teacher's preparation for effective and successful managing of learning computer science by her/his students. This procedure is similar to that proposed in [18] for preparation of any teacher to use information technology in the classroom. It is worth to mention that the main purpose of evaluation of teacher's work is to help and support her/him in better preparation for teaching computer science. We are glad to find out that our approach to teachers' certification has many elements in common with the teacher evaluation and improvement system discussed and proposed in the Measures of Effective Teaching Project supported by the Bill and Melinda Gates Foundation.

**Outreach Activities.** Various outreach initiatives and activities in the area of computer science education are organized in Poland nationwide or locally. They range from formal and informal lectures, courses, and workshops run by public or private institutions. Such activities contribute to school education by increasing motivation and preparation of school students for their future decisions to study computer science or related fields and become computer specialists. Informatics + was one of such projects (see [16]). More than 15 000 students from five regions of Poland took part in this project during 3 years. The project Informatyka + was awarded The Best Practices in Education Award by Informatics Europe in 2013 in "recognition of outstanding European educational initiative that improves the quality of informatics teaching and the attractiveness of the discipline."

Extracurricular activities of students, such as described here and below, promote computer science concepts, involving mainly mental tools of computational thinking. Such activities contribute to formal computer science education as well as to broaden informal and incidental learning.

**Competitions.** Competitions are typical outreach activities, they are usually run by institutions external to schools. These educational events require knowledge and skills exceeding what is taught in schools. They engage and develop skills necessary in the future professional activities such as: constant self-development, self-discipline, hunger for knowledge, ability to work in a team.

Olympiads in Informatics, which in fact are competitions in algorithmics and programming, attract most skillful students. However, the Bebras competition [1] promotes interest in computer science as well as in information and communication technologies to all school students of all grades. Bebras tasks are on concepts coming from information comprehension, logical and algorithmic thinking, games and puzzles, graphical representations of notions and objects, computer and software functions, etc. They aim at developing computational thinking in the contexts coming from various areas and school subjects.

### 3.4 The Road Map

The new computer science curriculum as described in Section 3.2 has been accepted by the Ministry of National Education and has been made available to teachers and schools in July 2015. Formally it will be included in the National Curriculum, what needs the Parliament approval, in 2016. In the meantime teachers will take part in various in-service courses on how to develop school syllabi based on the new curriculum and to develop educational materials for their instruction and for students. On the other hand, we expect that computer science departments at tertiary institutions will offer continuous in-service training for teachers, partly in the form of blended learning, based on the preparation standards for teachers, see [18].

## References


1. Bebras competition, `http://bebras.org/`
2. Bell, T.: Establishing a Nationwide CS Curriculum in New Zealand High Schools. Comm. ACM 57(2), 28-30 (2014)
3. Caspersen, M.E., Nowack, P.: Computational Thinking and Practice - A Generic Approach to Computing in Danish High Schools. In: Carbone, A., Whalley, J. (eds.) CRPIT, vol. 136, pp. 137-143. ACS (2013)
4. `http://computingatschool.org.uk/`
5. Cortina, T.J.: Reaching a Broader Population of Students through "Unplugged" Activities. Comm. ACM 58(3), 25–27 (2015)
6. CSTA: K -12 Computer Science Standards (2011), `http://csta.acm.org/Research/sub/CSTAResearch.html`
7. CSTA: Computational Thinking Task Force, `http://csta.acm.org/Curriculum/sub/CompThinking.html`
8. Gal-Ezer, J., Stephenson, C.: Computer Science Teacher Preparation is Critical. ACM Inroads 1(1), 61–66 (2010)
9. Gurbiel, E., Hardt-Olejniczak, G., Kołczyk, E., Krupicka, H., Sysło, M.M.: Informatics. Textbook for middle school. WSiP, Warszawa (2009) (in Polish)
10. Gurbiel, E., Hardt-Olejniczak, G., Kołczyk, E., Krupicka, H., Sysło, M.M.: Informatyka to podstawa. Textbook for all students in high school. WSiP, Warszawa (2012)
11. ISTE, `http://www.iste.org`
12. Settle, A., Franke, B., Hansen, R., Spaltro, F., Jurisson, C., Rennert-May, C., Wildeman, B.: Infusing Computational Thinking into the Middle- and High-School Curriculum. In: ITiCSE 2012, Haifa, Israel, pp. 22–27 (2012)

13. Stephenson, C., Gal-Ezer, J., Haberman, B., Verno, A.: The New Education Imperative: Improving High School Computer Science Education, Final Report of the CSTA Curriculum Improvement Task Force, CSTA. ACM (February 2005), http://csta.acm.org/Publications/White_Paper07_06.pdf
14. Sysło, M.M., Kwiatkowska, A.B.: Informatics *Versus* Information Technology – How Much Informatics Is Needed to Use Information Technology – A School Perspective. In: Mittermeir, R.T. (ed.) ISSEP 2005. LNCS, vol. 3422, pp. 178–188. Springer, Heidelberg (2005)
15. Sysło, M.M., Kwiatkowska, A.B.: The Challenging Face of Informatics Education in Poland. In: Mittermeir, R.T., Sysło, M.M. (eds.) ISSEP 2008. LNCS, vol. 5090, pp. 1–18. Springer, Heidelberg (2008)
16. Sysło, M.M.: Outreach to Prospective Informatics Students. In: Kalaš, I., Mittermeir, R.T. (eds.) ISSEP 2011. LNCS, vol. 7013, pp. 56–70. Springer, Heidelberg (2011)
17. Sysło, M.M., Kwiatkowska, A.B.: Informatics for All High School Students. In: Diethelm, I., Mittermeir, R.T. (eds.) ISSEP 2013. LNCS, vol. 7780, pp. 43–56. Springer, Heidelberg (2013)
18. Syso, M.M., Kwiatkowska A.B.: E-Teacher Standards and Certificates. In: Reynolds, N., Webb, M. (eds.) Learning while we are connected. WCCE 2013, vol. 2, pp. 145-151. UMK Toruñ (2013)
19. Sysło, M.M.: The First 25 Years of Computers in Education in Poland: 1965-1990. In: Tatnall, A., Davey, B. (eds.) History of Computers in Education. IFIP AICT, vol. 424, pp. 266–290. Springer, Heidelberg (2014)
20. Sysło, M.M., Kwiatkowska, A.B.: Playing with Computing at a Children's University. In: WiPSCE 2014, Berlin, Germany, pp. 104–107. ACM (2014)
21. Sysło, M.M., Kwiatkowska, A.B.: Learning Mathematics supported by computational thinking, In: Futschek, G., Kynigos, C. (eds.) Constructionism and Creativity, pp. 258–268. Österreichische Computer Gesellschaft, Vienna (2014)
22. Webb, M.: Considerations for the Design of Computing Curricula. In: Brinda, T., Reynolds, N., Romeike, R. (eds.) KEYCIT 2014, Berlin, pp. 163–173 (2014)
23. Wing, J.M.: Computational thinking. Comm. ACM 49, 33–35 (2006)