



Standards for Higher Secondary Education for Computer Science in Germany

Arno Pasternak¹(✉), Lutz Hellmig²(✉), and Gerhard Röhner³(✉)

¹ Fritz-Steinhoff-Schule Hagen, Technische Universität Dortmund, Dortmund, Germany

arno.pasternak@cs.tu-dortmund.de

² Universität Rostock, Rostock, Germany

lutz.hellmig@uni-rostock.de

³ Lichtenberg-Schule Darmstadt, Studienseminar Darmstadt, Darmstadt, Germany

groehner@t-online.de

Abstract. In this paper we will report on the *Standards for Higher Secondary Education for Computer Science in Germany*. After the first results of the PISA-studies in Germany the administration in the central and the federal states switched standards and curricula from setting teaching aims and cognitive learning objectives to describing students' learning outcome with practices.

On the central level this work was only done for the major subjects like Mathematics, Mother Tongue, Foreign Language and Natural Science. This is a disadvantage for the other subjects in school.

Therefore the German CS society—*Gesellschaft für Informatik (GI)*—installed a working group to develop standards for higher secondary education in 2013. By the end of 2015 this group finished the task and in January 2016 the board of the GI published this academic work as the official standards in Germany for higher secondary CS education [2].

It is to be hoped that these standards will have a positive impact on the standardisation of computer science teaching in the various German federal states, just as the standards for lower secondary education had almost 10 years earlier.

Keywords: Higher secondary computer science education
Education standards · CS in Germany

1 Introduction

In the late 1990s OECD has started to monitor and compare the performance of students with the PISA studies [11–13]. Germany's students did not match the expectations. As a consequence, stakeholders in Germany's educational system decided to shift from setting teaching aims and cognitive learning objectives

to describing students' learning outcome with practices. In subsequent years, federal authorities developed standards for a set of important subjects—but not regarding CS education [1, 7].

The German non-governmental professional organisation for CS—Gesellschaft für Informatik (GI)—took this task to empower CS education as a subject in schools. GI published national standards for lower secondary education in 2008 [1, 15]. These standards left an evident impact on the development of curricula in the 16 federal states of Germany. However, there is no compulsory CS in the lower secondary education in many schools.

In 2013 the GI continued by appointing a working group to develop standards for the higher secondary education. This working group finished its work in the end of 2015, and in January 2016 the GI board adopted these standards as the official position of the GI [2].

In this paper we give an account of the key ideas of these standards.

2 Computer Science Education in Germany

2.1 German Educational System in a Nutshell

Due to the responsibility of the 16 federal states of Germany, the German educational system is a 'hexadecimal assortment' of different approaches to establish a general education in each federal state. Nevertheless, there are some key points in common for all federal states: Education starts at the age 6 with primary school from grade 1 to grade 4 or in some federal states to grade 6. In each federal state, three degrees of qualification can be achieved by students: SEK I after grade 9 or grade 10. The range from grade 7 to grade 9 or grade 10 is called the lower secondary level. The third degree of qualification is attainable in the 'Gymnasiale Oberstufe'. It includes the lower secondary level together with the higher secondary level in grades 11 and 12—in some federal states until grade 13—and implies the higher education entrance qualification. In the higher secondary level, students are taught in a few core subjects (i.e. German, Mathematics, English), furthermore they attend lectures of an individual selection of subjects in different scientific domains. Besides that, the students may choose which subject he or she attends on a basic level¹ or on an advanced level².

Efforts to ensure a common, standardized higher education entrance qualification in Germany are conducted by the *Permanent Conference of the Ministers of Education and Cultural Affairs of the States of the Federal Republic of Germany (KMK)*.

2.2 Higher Secondary Education

The history of CS education in schools started in the 1960s. In some schools, voluntary courses took place in physical logic and programming with mini-computers. These were the precursor for CS in school. In the middle of the

¹ In German: Grundkurs or Grundfach.

² In German: Leistungskurs or Hauptfach.

1960s many people were involved in general discussions about the educational system in Germany [3, 14]. The result of these discussions was a new constitution of the higher secondary education in Western Germany with a system with compulsory and elective courses in grade 11 to grade 13 [5]. CS as a full-credit-subject was one of the innovations. The problem in 1972 was the same as today: the lack of CS teachers. Consequently there were many schools without an offer in CS.

In the GDR, between 1949 and 1990 an independent state of Germany, the government enacted a comprehensive program for CS education from grade 9 in schools up to vocational training and university education in 1985. As a part of this, eleven special schools for STEM education started to teach CS in 1986. Furthermore, curricula for CS education in grades 9/10 and 11/12 at schools for general education were developed and implemented successively. These offers were accompanied by a program of computer science teacher education [9].

Meanwhile in most of the federal states of Germany there are educational programs for CS, but the quantity is too small. The curriculum of CS in schools is different in the federal states. For example databases were not always part of the curriculum in the past. The quantity of lessons with information about hardware is slight. The programming language changed from mostly preferred language Basic in the first years via Pascal to Java.

Students have to fulfill many conditions in other subjects like Mathematics, First Language etc., so they often are not interested in learning CS. It is not easy, to make CS a part of a special permanent education.

In other papers there are more details about CS education in some federal states of Germany [4, 8].

2.3 Lower Secondary Education

CS as a topic in the lower secondary education in German schools followed CS in higher education a few years later. With a few exceptions only the situation in schools was worse than in higher education. There were and are not enough teachers for CS. Often teachers have to devise their own curriculum. Many teachers and administrators mistake CS for ICT. Mostly CS is offered as an elective course for the students. In the last few years there is a new discussion about CS in lower secondary education. In some federal states CS is established with a minimum of lessons and importance. Unfortunately a significant number of influential people struggle again and again with the idea of a compulsory subject in lower secondary education.

3 Standards for Computer Science for Higher Secondary Education

The lack of consideration for Computer Science Education by nationwide educational authorities was the reason for the German Society of Computer Science (GI) to develop recommendations for standards in computer science education,

beginning with standards for lower secondary education in 2008. Since the standards for CS for higher secondary education are continuing the ideas in this paper, it is advised to describe the main ideas of lower secondary standards first in Sect. 3.1.

The situation for higher secondary education is similar to lower secondary education. There are some official standards for some subjects, but not for CS Education yet. Instead of that the *Unified Requirements for Examinations in Higher Secondary Education* [6] describe the demands for the final exams for the higher education entrance qualification. The authors of the higher secondary standards also had to keep in mind the compatibility of the unified requirements with the higher secondary standards.

3.1 Standards for Computer Science Education for Lower Secondary Education

History. From 2003 to 2008 researchers and teachers of CS developed national standards for CS in lower secondary education. After its finalisation they asked the German society for CS (GI) for a takeover of these standards as an official GI-paper [1]. At the beginning of 2008 the GI approved this request.

As a result, in Germany we have on the one hand ‘official’ standards for some key subjects developed by the inter-federal-state official organisation KMK, on the other hand ‘semi-official’ recommendations for standards, including the *Standards for Computer Science Education*, developed by the GI.

While there were no official nationwide curricula in CS and additionally in many federal states there were no experience and no paper either, administration in most federal states orientated towards these GI-standards in the following years. This was not the expectation at the beginning of the work and represents a great success of these standards.

Today CS in lower secondary education is effectively defined by these standards in Germany.

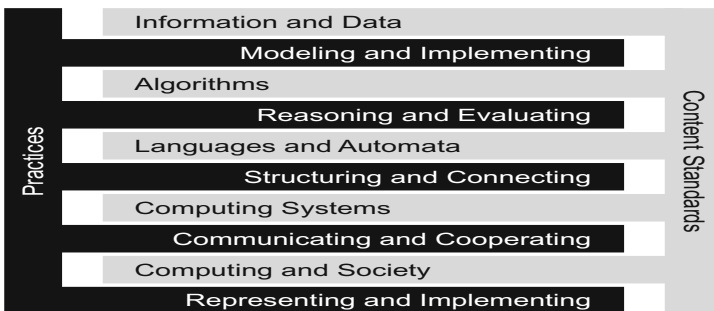


Fig. 1. Matrix of practices in lower secondary education, according to [15]

Structure. The structure of these guidelines is based on the *Principles and Standards for School Mathematics* [10]. The standards are divided in two parts: *principles* and *standards*. The principles are sub-divided in *equal opportunities, curriculum, teaching and learning, quality assurance and the multidisciplinary of CS*. The standards are sub-divided in *contents and practices* respective Fig. 1. Afterwards these standards are first described and made concrete for grades 5 to 7 and secondly for grades 8 to 10.

3.2 Underlying Views on Computer Science

In Germany, CS in schools is generally referred to as the science of theoretical analysis, conception and the concrete implementation of complex IT systems. We consider an IT system as a specific assembly of hardware, software and components of networks to solve a problem. This assembly contains the non-technical aspects of embedding IT systems in social contexts. Therefore CS contains elements of Mathematics, Engineering, Social and Natural Science as well as Humanities.

The core products in CS are immaterial in contrast to the traditional engineering sciences. The impact of these products is very powerful. The development of an IT system is based on strategies that are not learnable from experiences from real life.

We need a new additional view for the information society inside the general education – the computational education. The basic science of this computational education is computer science.

CS makes aware of the regularity of information processing in society, nature and technology. Additionally CS and computational thinking completes the professional spread of other subjects. Computational development and problem solving is a creative process with covering theory, abstraction and design. Computational thinking and tools are involved in all parts of science, economy and technology. Everybody is concerned with CS at least as a user.

CS lessons give students many opportunities for the increase of competences, which are important for a self-determined life in a computational society. Students cannot extrapolate the comprehension of the modern computational society with everyday experiences. The qualification in knowledge, methodical, social and self competences happens in CS lessons in a mutual and a holistic way with situations of the students' everyday life. The use of and the interaction with concepts of modeling and structuring, software tools and programming languages is essential in a CS education. Learning and working in teams lead to a better qualification in study and profession.

3.3 Model of Competences for Higher Secondary Education

The model of competences for higher secondary education (respective Fig. 2) is based on the *Tyler matrix* of practices and contentual competences [16] and on the standards of computer science education for lower secondary education

(see Fig. 1). The structure is the result of a longer discussion in the community of scientists, educators, and teachers in the field of computing.

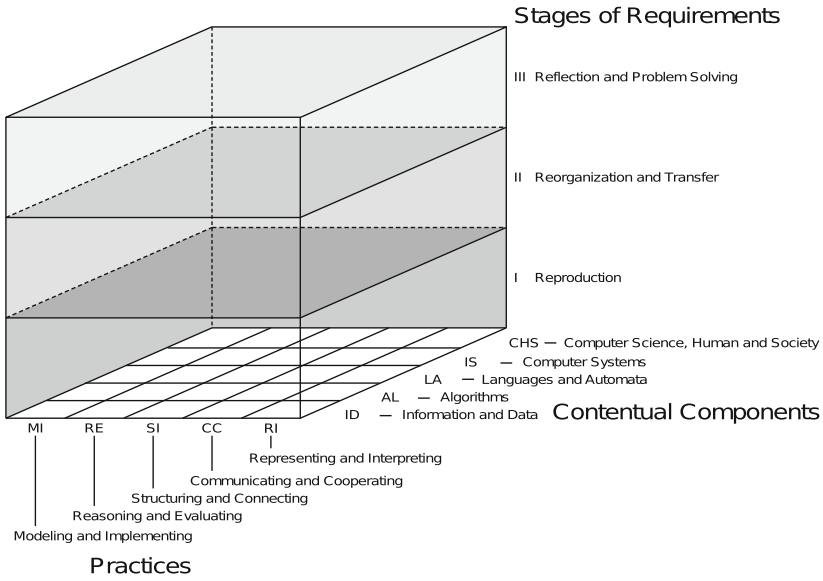


Fig. 2. Competence model of the standards for computer science in higher secondary education

Competences are the result of the amalgamation of contentual and practical components. Upholding this system is a key concept for the acceptance and the use of the standards by stakeholders. Regarding the framework conditions of higher secondary education in Germany, it was necessary to modify the system.

3.4 Practices

Practices describe the way in which students should deal with specific content. Thus the inherently cognitive abilities and skills, which are specific to the subject, are linked, but are not tied to special computer science contents. They can be used by students only in active involvement with the content and enable them to apply their acquired knowledge and skills for solving problems in new situations.

The following five different practices are used according to the educational standards for the lower secondary education:

Modeling and Implementing

These are the central parts of the modeling cycle: to analyze a problem, to design and implement a CS-model, test and evaluate the implementation on a computer system.

Reasoning and Evaluating

Reasoning is the idea of verification of a given statement or fact by rationally comprehensible arguments. The evaluation is to formulate a value judgement by considering the context and using transparent and appropriate criteria.

Structuring and Connecting

In structuring, facts are analyzed from a CS point of view. Objects and process as well as their interaction are systematically noted. In networking, existing contexts, effects and analogies in and outside of computer science are recognized and used.

Communicating and Cooperating

Communication in oral and written form uses the technical language as well as the use of methods for information dissemination from different sources.

Cooperation is necessary for teamwork in the field of computer science, especially for a project. Students use network based platforms for communication and cooperation and reflect their opportunities and risks.

Representing and Interpreting

Concepts and facts of computer science are presented in a variety of forms. Formal representations enable automatic information processing and serve for professional communication. Representations of the modeled reality are interpreted. Interpreting is a prerequisite for assessing facts. Data are obtained by interpretation of information.

3.5 Stages of Complexity in the Dimension of Practices

Since the *standards* are only recommendation up to now, the *Unified Requirements for Examinations in Higher Secondary Education* are obligatory in all federal states. They are based on a system of three stages with increasing complexity. Since the construct is in common for all subjects in every federal state of Germany, it has been included in the standards for higher secondary education for CS. The authors of the standards have labeled the stages.

They have been labeled in the standards for higher secondary education for CS.

1. **Reproduction** contains the display of known facts of a definite area in a trained context; the description and representation of procedures, methods, and principles of computer science as well as the use of skilled procedures in well-known situations.
2. **Reorganization and Transfer** connotes the self-reliant use of known facts for answering further questions in familiar contexts; the transfer of knowledge to slightly modified questions, situations, and methods; and the use of known techniques, methods and principles for solving new problems in familiar situations.
3. **Reflection and Problem Solving** is represented by a systematic approach to understand, represent and judge complex conditions and to draw conclusions. Furthermore, the students purposely choose and modify their acquired knowledge to solve new problems.

The levels of complexity relate to the competences of the students to be developed. Therefore, they were transferred into the section of practices. For each of the five components Modeling and Implementing, Reasoning and Evaluating, Structuring and Connecting, Communicating and Cooperating as well as Representing and Interpreting, three stages of complexity are defined separately and cumulatively.

We describe the levels of practices by the example of *Modeling and Implementing*. Modeling and Implementing are the key elements of the circle of modeling, including the analysis of a problem and the design of a computational model, which will be implemented, tested and evaluated.

At the lowest stage of *reproduction*, the students are able to display an already known model in a familiar representation; discover a given model or investigate a given implementation and test an implementation with given cases. The medium level—*reorganization and transfer*—has been achieved if the students prove the applicability of a given model for solving a problem, accomplish modeling and implementing—according to an analysis of a problem—with a familiar method of modeling. Furthermore, they use IDEs in an appropriate way for implementing a model and prove an implementation on functionality and correctness, including special cases, systematically. Reflection and Problem Solving takes place, if students—according to an analysis of a problem with a higher grade of complexity—model and implement with a method (not necessarily with a familiar one) of modeling. They revise their own work, considering aspects of efficiency, generality, and reusability. Students reflect their practice of problem solving and use their insights for further work.

3.6 Fundamental and Extended Level of Requirements in the Contentual Dimension

Each subject in higher secondary education can be offered on a basic level or on an advanced level with twice the time. For that reason, the standards have two stages for each contentual section. The first stage is mandatory for both levels, while the second stage describes advanced contents and concepts in addition to the first level.

There are five areas of contents: Computer Science and data, algorithms, languages and automata, Computer Science systems, Computer Science and society. Each content area is described with about five notes in each stage.

In the next paragraph we will extensively outline three examples of content areas and the other two briefly.

Selected Examples of Content

Information and Data. Information is the context-related sense of a statement, instruction or message. Computer science is characterized by the systematic representation and automatic processing of data as a carrier of information.

Data is a representation of information in a formalized way for communication, interpretation and processing, represented by strings that follow in a suitable syntax. Data become information again when interpreted in a concrete context.

Fundamental Level. The students ...

- distinguish between characters, data and information as well as between syntax and semantics,
- analyze data on their structure,
- form information as data with data types and data structures,
- use, model and implement operations on static and dynamic data structures,
- create a data model and implement a data model as a database,
- investigate and organize data while regarding redundancy, consistency and persistence,
- use a query language to display and manipulate data and interpret the data.

Advanced Level. The students ...

- use, model and implement operations on complex datastructures,
- develop a real part of the life with complex relationships in a database.

Algorithms. Algorithms are finite descriptions of processes for solving problems and result in a clearly defined sequence of actions during execution. An execution on a computer requires the formulation in a programming language. Complex problems can be solved, if—in addition to the algorithmic basic structures—design methods and data structures are developed. The implementation of an algorithm requires sufficient testing and, if necessary, revisions.

Fundamental Level. The students ...

- use algorithmic basic structures (sequence, alternative, repetition) and implement an application using a programming language,
- analyze given programs respectively the basic concepts including variables, references, nesting and functional decomposition,
- implement algorithms and describe them in adequate form,
- use modularization for structuring of algorithms and use them for their implementation,
- use software libraries or available modules to implement an algorithm,
- test and revise a program systematically.

Advanced Level. The students ...

- model and implement iterative and recursive algorithms and data structures,
- compare and assess algorithms for solving a problem, including efficiency,
- analyze examples of the complexity of algorithms
- assess the practical and theoretical boundaries of algorithmization.

Languages and Automata. Formal languages are the basis for communication with automata and are used in a variety of application scenarios in computer systems. In contrast to natural languages, formal languages have a definitive syntax defined by grammar, syntax diagrams, or language descriptions.

Automata are state-based systems that read and handle an input character after character. Automata types can be differentiated according to the design of their memory and thus according to their basic possibilities and limits. (Following ditto the fundamental and the advanced level.)

IT Systems. A computer system is a specific combination of hardware, software and network components to solve an application problem. Also included are non-technical aspects that are relevant by e.g. embedding in a sociocultural system, inclusion of the potential users in the development process, the economic and environmental consequences. Competent use, design and evaluation of computer systems require a fundamental understanding of their structure and functioning. For the development of computer systems, machine-processable facts of the real world are identified and modeled. Typical fields of application of computer systems are data management, communication, graphics, simulation, robotics, process control and regulation or speech processing. (Following ditto the fundamental and the advanced level.)

Computer Science and Society. Information systems shape our information society and interact with people and society. Based on socially relevant questions or own experiences in dealing with IT systems, interactions between IT systems and their social embedding are analysed. In confrontation with normative, legal, ethical and social aspects, an orientation framework and a sense of responsibility in dealing with modern information technology are developed. Freedom of choice in dealing with IT systems in accordance with social norms and standards and appropriate responses to risks when using computer systems are reflected. (Following ditto the fundamental and the advanced level.)

3.7 Illustrating Tasks

Presenting tasks is an effective way to illustrate the generally formulated items and make them vivid. Depending on the situation, different kinds of tasks are needed to acquire new knowledge, to consolidate knowledge or examine the adopted competencies. Learning is a process which includes phases of performing as a subset. Performing can be a self-examination to reflect the improvement of the learning progress by the learner. If somebody else (e.g. a teacher) judges the result of the learning process by giving marks, we call it scoring. A model of the relationship between Learning, Performing and Scoring is given by Fig. 3.

Regarding the model, three types of tasks illustrate the standards: A collection of eight complex tasks from written exams of different federal states of Germany show an impression of different approaches to prove students' competences. Furthermore, the document includes four tasks for oral exams developed

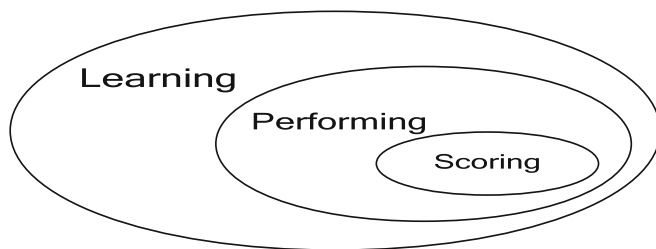


Fig. 3. Relationship between learning, performing and scoring

by experienced teachers. Last but not least, five tasks show examples how knowledge can be developed and trained according to the standards.

Every task is complemented with an anticipated typical solution and an assignment to the related contentual and processual sections, and to the stage of complexity.

Finally a list of suitable operators is part of the document to ensure a clear understanding of the required demands of the tasks.

4 Conclusion

CS in Germany today is not a substantial part of education in lower and higher secondary education. During the last 10 years the *Permanent Conference of the Ministers of Education and Cultural Affairs of the federal states in the Federal Republic of Germany* have developed standards in subjects like Mathematics, Languages and Natural Sciences. In the majority of the federal states of Germany curricula are introduced, the contents are more or less different. The German CS organization GI undertook the task and developed CS standards for higher secondary education in Germany. In January 2016 these standards were adopted by the board of the GI. These standards can be seen as an extension of the standards for lower secondary education, written in 2008.

Better than expected the standards for lower secondary education have been involved in the development of curricula in the federal states of Germany. We hope the standards for higher secondary education fulfill the same role.

Acknowledgments. In Germany standards were usually developed in working groups installed by administration organizations without a considerable discussion with teachers or other people. These standards for CS were developed without involving any administration structure. The goal was to involve many teachers in this process. Experts were invited by the working group to discuss part of these standards. In September 2015 these standards were reviewed on the German conference *INFOS* (Informatik und Schule). Additionally an online survey was conducted in November 2015. Thanks to all the people involved in this process.

References

1. Arbeitskreis 'Bildungsstandards' der GI: Grundsätze und Standards für die Informatik in der Schule. LOGIN-Verlag, Berlin (2008)
2. Arbeitskreis 'Bildungsstandards SII' der GI: Bildungsstandards Informatik für die Sekundarstufe II. LOGIN-Verlag, Berlin (2016). http://informatikstandards.de/docs/Bildungsstandards_SII.pdf. Accessed 09 June 2016
3. Dahrendorf, R.: Bildung ist Bürgerrecht. Plädoyer für eine aktive Bildungspolitik. Nannen-Verlag, Hamburg (1965)
4. Hubwieser, P.: Computer science education in secondary schools – the introduction of a new compulsory subject. *Trans. Comput. Educ.* **12**(4), 16:1–16:41 (2012)
5. KMK (Ständige Konferenz der Kultusminister der Länder der Bundesrepublik Deutschland): Vereinbarung zur Gestaltung der gymnasialen Oberstufe in der Sekundarstufe II (1972)
6. KMK (Ständige Konferenz der Kultusminister der Länder der Bundesrepublik Deutschland): Einheitliche Prüfungsanforderungen in der Abiturprüfung Informatik. Luchterhand, Neuwied (2004)
7. KMK (Ständige Konferenz der Kultusminister der Länder der Bundesrepublik Deutschland): Bildungsstandards im Fach Physik für den mittleren Bildungsabschluss - Beschluss vom 16.12.2004. Wolters Kluwer Deutschland - Luchterhand, Neuwied (2005)
8. Knobelsdorf, M., et al.: Computer science education in North-Rhine Westphalia, Germany. A Case Study. *Trans. Comput. Educ.* **15**(2), 9:1–9:22 (2015)
9. Merkel, G.: Über Informatikkenntnisse des gebildeten Bürgers. In: Stetter, F., Brauer, W. (eds.) *Informatik und Schule 1989: Zukunftsperspektiven der Informatik für Schule und Ausbildung*, pp. 56–68. Springer, Heidelberg (1989). https://doi.org/10.1007/978-3-642-75163-9_5
10. NTCM: Principles and Standards for School Mathematics. National Council of Teachers of Mathematics, Reston (2000)
11. OECD: Measuring Student Knowledge and Skills A New Framework for Assessment: A New Framework for Assessment. Measuring Student Knowledge and Skills A New Framework for Assessment, OECD Publishing (1999)
12. OECD, Adams, R., Wu, M., Program for International Student Assessment, Organisation for Economic Co-operation Development: Pisa 2000 Technical report. OECD (2002)
13. OECD, Statistics, U.: PISA Literacy Skills for the World of Tomorrow Further Results from PISA 2000: Further Results from PISA 2000. OECD Publishing (2003)
14. Picht, G.: Die Deutsche Bildungskatastrophe. Deutscher Taschenbuch Verlag, München, 2 edn. (1965), 1. Auflage 1964, Walter-Verlag Olten
15. Puhlmann, H., Schulte, C., Brinda, T.: Bridging ICT and CS: educational standards for computer science in lower secondary education. *Source ACM SIGCSE Bull. Arch.* **41**(3), 288–292 (2009)
16. Tyler, R.W.: Basic Principles of Curriculum and Instruction. University of Chicago Press, Chicago (2010)