Future Academy®
Offering excellence in Social and Behavioural Sciences

The European Proceedings of
Social & Behavioural Sciences
EpSBS

eISSN: 2357-1330

ERD 2016: Education, Reflection, Development, Fourth Edition

Diagnosis of Major Challenges in Information Technology Education in Lebanese Schools

EL-Zakhem Imad[a]*

* Corresponding author: EL-Zakhem Imad, izakhem@balamand.edu.lb

[a]*Department of Computer Science, University of Balamand, Lebanon izakhem@balamand.edu.lb*

Abstract

http://dx.doi.org/10.15405/epsbs.2016.12.18

In the educational system in Lebanon students are exposed to IT through all their school years, starting with the primary school, but in the first university programming course a high drop or failure rates are noticed. Despite this fact, we still identify high failure rates among students in programming courses. This leads us to the following question: Does this failure has roots in the High School curriculum and the tutoring strategies or is related to the programming curriculum? Our hypothesis derives from an integrated vision of the curriculum as a coherent entity spread all over the student's learning curve despite the diversity of learning periods. The three components involved in the process of IT education are: the curriculum, the instructors responsible for the curriculum implementation and the students, who are the targeted population. To validate our hypothesis we examine the three components of IT education using a specific methodology: an analysis of the IT curriculum in schools, done in order to verify that the curriculum will meet the requirement of the university computer science courses; interviews with the instructors from different educational institution, private and public in order to assure and examine their alignment with the goals of the curriculum and finally an assessment of the students' comprehension acquisition of the basic concepts of IT education is conducted through a simple quiz given to high school students in Lebanon.

Keywords: IT school curriculum; computer education; programming language.

1. Introduction

The high impact of IT in a growing number of fields like industry, communication, health, science and society in general is universally recognized. We now speak of a "digital world", based on two major domains, hardware and software. The development of information technology and digital information of all kinds affects the whole society and its activities. The digital activities and industries manipulate the abstract information, which is different than the concrete products. A program or a computer data weigh nothing, has no transport cost, and can be duplicated easily.

Computer science education gives the key of the future to all citizens in a world which will be even more digitalized and computerized than the current one so they understand and participate to its evolution rather than suffer by simply consume what is being done and decided elsewhere. The education prepares them for new occupations and employment generated by computing. The engineers and scientists, provided by the knowledge and adequate scientific and technical practices, should be able to make the best use of all IT tools to adapt these tools to their business or create new products.

Also, computer science is a subject of considerable scientific research worldwide, covering topics like algorithms, computation, software, and hardware design. The education of computer science in schools is an interdisciplinary topic between education and computer science. Most education researchers don't know what computer science is and also most of the computer science researchers don't focus on education and don't have a vision about how to deliver computer science lessons especially to school students. This paper is analyzing the IT curriculum in schools and aiming to bridge the gap of this important topic of research between education and computer science.

2. IT or Computer Science

The terminology used to describe the IT teaching is not common, also the IT introduction in the schools' curricula vary from one country to another. We can note a great diversity in the naming of this discipline: ICT, IT literacy, Informatics, Computer Science. One must distinguish between two types of education: a teaching centered on familiarization of information technology usage and on the other hand an education specialized more into computer science. The first type is found especially in the early stages of education (IT literacy) and it is more prevalent than the computer science teaching. In general ICT and IT literacy imply the use of applications and knowledge of computer operation, while computer science and informatics are more about computing and programming languages.

The problem of recognition of the need for a real computer education is not limited to a specific country. It is increasingly debated and many countries have already launched new initiatives on the subject. Notably, the United Kingdom has decided to introduce computer science at the Baccalaureate at an equal level to other sciences (Académie des sciences 2013).

In countries where a computer science teaching exists, computational thinking, algorithms and computer programming constitute the core of the curriculum. The programming languages are not always specified, because the interest is mainly about the acquisition of basic concepts. Some countries are introducing networking and management information systems by the end of the secondary level. A comparison of different cases in Finland, Japan, Massachusetts, Ontario and Singapore performed for the Royal Society in England shows that the age of introduction of a computer specialist education varies from 12 to 16 years (royal society 2012). As for the secondary grades many countries have an IT independent discipline, with a formal program, trained teachers, and dedicated weekly hours. Some countries explicitly request the development of programming ability in their learning objectives (notably Germany, Greece, Spain, Italy, Poland, and United Kingdom) (EACEA P9 Eurydice 2011). There is, moreover, a great diversity of the status of IT lessons according to the courses: some countries offer mandatory courses and others optional courses, others now rely on integration skills and computer-related knowledge into existing disciplines. It is important to note a recent trend towards the implementation of specific curricula for computer science education in schools (Académie des sciences 2013). Until now school students consider IT is simply word processing and web browsing, and they have a poor understanding of what a computing career involves (Yardi & Bruckman 2007).

The computer science carries a specific reasoning process appeared in the early 1970s. Some of the attributes of this science are algorithmic, modeling and simulation. According to many researchers computer science is actually a very old science, which moreover does not necessarily require a computer (computer unplugged). The unplugged computing project outlined in (Bell 2009) exposes school students to education ideas without having to use computers. It is based on fun activities and competitions to show kids the kind of reasoning expected from a computer. The project has seen extensions abroad and support in the industry. It has been translated into 12 languages and has become popular.

Computational thinking is based on different primitive concepts. When children are learning these concepts they will be more familiar to implement their acquired knowledge using computing tools like programming language. The main use of computing is for problem solving by using either common sense or logic or even a well structured algorithmic method. This ability should be learned by all students like all other sciences mathematics, physics chemistry and biology despite that not all the students will select one of these sciences for their higher education or professional career (Wing, J. M., 2008). The introduction of computational thinking in education in the early stages of education is a fact in several countries (Mannila, 2013). Examples, activities and lesson plans may help those involved in the training of teachers and policymakers to make decisions regarding when and how computational thinking can be included in the educational institutions (Mannila, 2013).

3. Curriculum Analysis

The Lebanese national IT curriculum has set 10 objectives spread over few topics like IT literacy, computing and communication.

- Some of the objectives are already outdated specially those referring to “positive attitudes toward computer”, “interact with other cultures and civilizations through various programs and computer networks”, because nowadays students are living in the era of mobile connectivity and social networks.
- Five objectives referring to information management and navigation, database management, acquaintance of basic computer concepts and their use in various domains, manipulation of computer’s peripherals for various purposes. These 5 objectives are not inter-related. The topic of computer components is just about historic information of the machines’ evolution. It is a narrative and passive part which does not need a lot of interaction from the students. Word processing, spreadsheet and presentations techniques are didactic and instructive lectures.

Students will learn and memorize the steps to achieve the required objectives. If a student misses a lesson, he can continue in the following one because the instructions are not inter-related and not depending on each others.

- Two out of 10 objectives referring to computing and problem solving. Table 1 shows these objectives set across grades 8 till 12. When it comes to programming, writing a paragraph in a natural language is similar to writing a program to solve a problem. A student has to express his thoughts, to recall his vocabulary and apply the grammar rules while writing his sentences. If the student doesn't know the needed word and the grammar rule he will not be able to express his idea, the same happens in programming, a program should be well written using the proper keywords and proper syntax rules. Programming language is like any formal language needs to be practiced extensively to be learned. It is not something that can be learned by-heart (Kaasboll 2002), (Du Boulay, 1989).

Table 1. Programming skills set from grade 8 till grade 12

Class:	**Grade 7**	**Grade 8**	**Grade 9**
No of hours:	12	8	8
Contents:	• Programming environment • Fundamental instructions • Repetition statement • Arithmetic operators (+,-,*, /) • Simple Procedures • The Editor	• Words and Lists •Variables • Procedures with variables • Procedures management	• Interactive programming • Logical operators (NOT, AND, OR) • Conditional statement (If ...Then) • Applications: constructing interactive programs.
Class:	**Grade 10**	**Grade 11**	**Grade 12**
No of hours:	18	16	18
Contents:	• Computer programming and problem solving. • Programming environment. • Program structure and elements. • Simple programming applications.	• Programming techniques: comments, scrolling, dialog boxes, menus. • Loops and decision statements. • Student's projects.	• Programming techniques: graphics, multimedia, OLE. • Students' projects.

At the time being, teaching IT is optional and it not included in the national exams. Some schools don't have the infrastructure and the needed teachers. The curriculum is now undergoing a major review and more weight is going to be given to computing and programming. Also it is being discussed to make it mandatory and to be included in the national exams. This trend can be seen in France and in England and many other countries (Académie des sciences, 2013), (Royal Society, 2012).

4. Textbooks and Instructors

The national books for IT have many units to reflect the different axes in the national curriculum: Computer Components, Operating System, Word Processing, Electronic Drawing, Spreadsheet, Programming, and Electronic Presentation Techniques. Figure 1 and Figure 2 show only the unit of programming lessons for grades 8 and 9 respectively. There are 8 lessons in grade 8 to be delivered in 8 sessions of 50 minutes each. Many drawbacks are noticed and listed below:

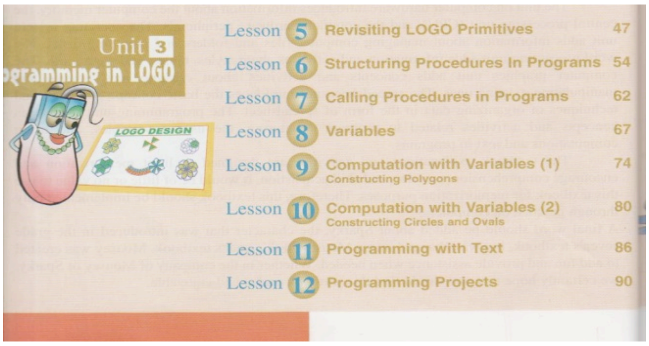
- The students have to move to the lab, wait to start the machines and the application, and at the end get back to their classrooms. All instructors agreed that the time is not sufficient at all for the programming lessons.
- The students introduced to some programming concepts in grade 8 have to wait one year to learn more new concepts in grade 9 which is a long period of time. After one year almost all students will not be ready for the new concepts.
- In all the students spend 21 hours spread over 3 years (grade 7, 8, 9)
- Memory concept and assignment operator are not introduced first.
- Introducing many concepts in the first year in 12 hours including procedure which is not a primitive concept.

The computer science is after all a science and it is based on computing and expressed through programming. Since students begin to learn science in the primary school, so it should have its place in the schools' curriculum like all other sciences afterward the primary classes. For example, in mathematics the students start by learning numbers then addition and subtraction. They begin to understand these elementary concepts of mathematics and in each grade a deeper look of the same concepts is acquired. The concepts of the programming science are to be learned the same way like other sciences. The main topics in programming are the memory concept and the variables, the assignment, the selection structures, the repetition structures, the classes and objects. Since many of these topics can be viewed as concepts, these should be learned at the smaller ages, knowing that at theses ages the acquiring of new concepts is easier (Ben-Ari M, 1998), (Hamidi et al., 2011).

The instructors follow the content of the books which are aligned with the curriculum. Some effort is done to seek new books where programming has more weight as described in figures 3 and 4.

5. Students' Assessment

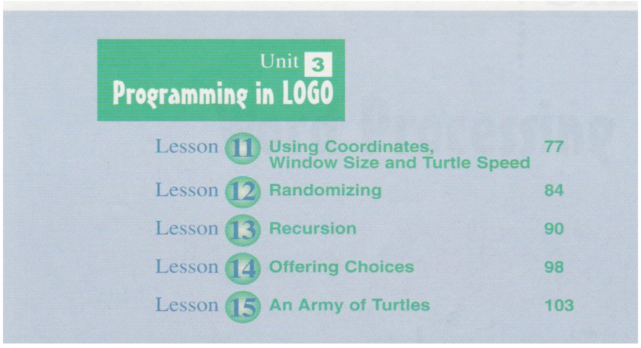
An assessment is conducted to show the different types of skills between sophomore students in the beginning of their academic year. All students are registered to undertake a programming language introductory course. The results showed that almost all students could use applications as word processing, spreadsheet calculation, and electronic presentations to a certain extent. On the other hand, the students failed to prove any knowledge of the primitive concepts of programming like assignment, variables, selection structures and repetition structures.



Unit 3
ogramming in LOGO

LOGO DESIGN

Lesson 5	Revisiting LOGO Primitives	47
Lesson 6	Structuring Procedures In Programs	54
Lesson 7	Calling Procedures in Programs	62
Lesson 8	Variables	67
Lesson 9	Computation with Variables (1) Constructing Polygons	74
Lesson 10	Computation with Variables (2) Constructing Circles and Ovals	80
Lesson 11	Programming with Text	86
Lesson 12	Programming Projects	90

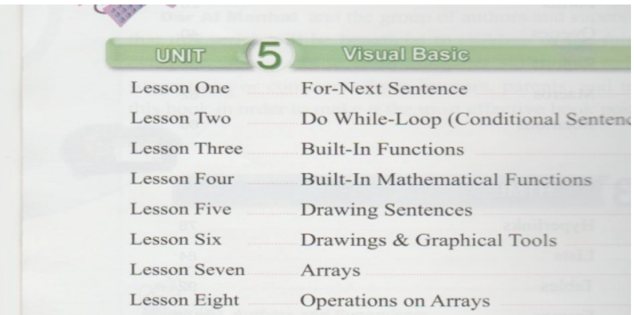


Unit 3
Programming in LOGO

Lesson 11	Using Coordinates, Window Size and Turtle Speed	77
Lesson 12	Randomizing	84
Lesson 13	Recursion	90
Lesson 14	Offering Choices	98
Lesson 15	An Army of Turtles	103

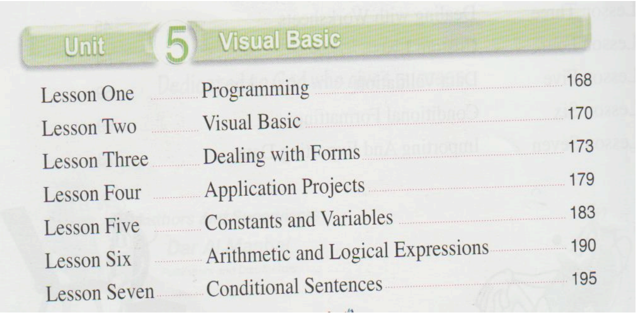
Figure 1. Grade 8 programming lessons

Figure 2. Grade 9 programming lessons



UNIT 5 Visual Basic

Lesson One	For-Next Sentence
Lesson Two	Do While-Loop (Conditional Sentenc
Lesson Three	Built-In Functions
Lesson Four	Built-In Mathematical Functions
Lesson Five	Drawing Sentences
Lesson Six	Drawings & Graphical Tools
Lesson Seven	Arrays
Lesson Eight	Operations on Arrays



Unit 5 Visual Basic

Lesson One	Programming	168
Lesson Two	Visual Basic	170
Lesson Three	Dealing with Forms	173
Lesson Four	Application Projects	179
Lesson Five	Constants and Variables	183
Lesson Six	Arithmetic and Logical Expressions	190
Lesson Seven	Conditional Sentences	195

Figure 3. Grade 8 programming lessons in VB

Figure 4. Grade 9 programming lessons in VB

6. Conclusion

The IT curriculum of schools should be revised. Computational thinking still too limited to scientists, has a special interest because it enables the development of analytical thinking and should be taught to young children. Computer literacy should be introduced in the small grades but not to constitute a core from grade 7 to grade 12. Pupils don't learn to use *Whatsapp* and different complex mobile applications in school although they don't face difficulties using them. If it is decided to include computational thinking, algorithms and programming it should be treated like a science. At the end, the new curricula should be tailored to the diversity of interactions with stakeholders, but without never forget the profound unity of concepts.

References

Académie des sciences (2013). L'enseignement de l'informatique en France, Il est urgent de ne plus attendre.

Bell, T., Alexander, J., Freeman, I., & Grimley, M. (2009). Computer science unplugged : School students doing real computing without computers. The New Zealand Journal of Applied Computing and Information Technology, 13(1), 20. 1

Ben-Ari M, (1998). Constructivism in Computer Science Education, 29th ACM SIGCSE Technical Symposium on Computer Science Education, Atlanta Georgia: ACM press

Du Boulay B., (1989) Some Difficulties of Learning to Program, Studying the Novice Programmer, Lawrence Erlbaum Associates. pp. 283-299

EACEA P9 Eurydice (2011). Key Data on Learning and Innovation through ICT at School in Europe 2011, ISBN 978-92-9201-184-0 doi:10.2797/61068, Education, Audiovisual and Culture Executive Agency, 2011, http://eacea.ec.europa.eu/education/eurydice

Hamidi et al., (2011) Information Technology in Education, Procedia Computer Science 3, 2011, pp 369–373

Kaasboll J. (2002). Learning Programming, University of Oslo

Mannila, L., Dagiene, V., Demo, B., Grgurina, N., Mirolo, C., Rolandsson, L., & Settle, A. (2014). Computational Thinking in K-9 Education. Proceedings of the Working Group Reports of the 2014 on Innovation ; Technology in Computer Science Education Conference, ITiCSE-WGR '14 (p. 1–29). New York, NY, USA : ACM. Doi :10.1145/2713609.2713610

Royal Society (2012), Shut down or restart? The way forward for computing in UK schools Issued: January 2012 DES2448, http://royal.society.org/education/policy

Wing, J. M. (2008). Computational thinking and thinking about computing. Philosophical Transactions of the Royal Society of London A : Mathematical, Physical and Engineering Sciences, 366(1881), 3717-3725. doi :10.1098/rsta.2008.0118

Yardi, S., Bruckman, A (2007). What is computing?: Bridging the gap between teenagers' perceptions and graduate students' experiences. In Proceedings of the 3rd International Workshop on Computing Education Research (ICER'07). Atlanta, GA