# Introducing Computer Science to Educationally Disadvantaged High School Students - The Israeli Experience

Dalit Levy

*Department of Education in Science and Technology*

*Technion, Israel Institute of Technology, Haifa 32000 Israel*

dality@tx.technion.ac.il

## 1. Background

The social and educational issue of making the discipline of computer science accessible to the public, including high school students who grow up in low socio-economic status families and in rural areas, have been shaping our thinking for over a decade. In the phrase "our thinking" I refer to the Israeli community of computer science (CS) educators, educational researchers and curricular developers in general, and to the "Migvan" R&D team I belong to in particular[1]. In what follows I will shortly describe the current high school CS curriculum in Israel, suggest principles for a course that is intended to introduce basic CS concepts and ideas to educationally disadvantaged high school students, and reflect upon our experience with teachers and students taking that course.

## 2. The CS Curriculum in Israeli Schools

Within the Israeli educational system, CS entered high schools as a discipline in itself in the mid-1970s [2]. At the beginning of the 1990s, a new CS and information technology (IT) curriculum had been developed, a new set of curricular materials (books, presentations, educational software) had been published, and more and more high schools opened CS classes and IT-oriented tracks. The new curriculum that is nowadays mandatory in all Israeli high schools emphasizes the foundation of algorithmics [1], and integrates the 'abstract' introduction to CS concepts and problem solving methods with the implementation of these concepts and methods using a real programming environment.

The national curriculum suggests three levels of learning CS and five learning modules (called "units") from which each school can construct its unique CS curriculum. The more prestigious CS curriculum is the highest level, where students take all five units throughout their three-year high school. The intermediate level is constructed of three units over two years, and in the lowest level students usually take only one course (called "first unit") during one school year at the tenth grade. While in the highest level the students usually have more or less homogenous mathematical, scientific and thinking skills background, the populations that learn the intermediate and the lowest level are heterogeneous with diverse students' background and ability. In special, those students who take only the first unit course are often referred to as disadvantaged – whether their weaknesses emerge from their families socio-economic status, their individual learning and schooling history, behavioral factors, or some diagnosed learning disabilities. These 'first unit' classes have been at the core of Migvan's R&D activities for the last twelve years, when we have strived for designing, implementing and putting into work a curricular unit in CS that could be disciplinary sound and meaningful for these challenging classes at the same time. The main educational principles for the 'first unit' of the Israeli CS curriculum are presented below.

## 3. Principles of the first CS Unit

The goal of the overall CS curriculum, as has been stated by the ministry of Education disciplinary committee[2], is to deal with basic CS concepts and with the design of computerized systems. This goal is supplemented with four basic principles:

1. Integrating theory and practice.
2. Adjusting the conceptual and intellectual challenge to the ability of high school students.
3. Modularity of contents and units.
4. Balance between mandatory and optional subjects.

Both the goal and the basic principles have been adapted in designing the first unit and in developing the learning environment. For example of implementing the goal, the first unit deals with basic CS concepts like algorithmic patterns and structured programming. For example of the last principle, teachers of the first unit can choose between two programming environments – using Pascal or Microworlds Project Builder[3], and although the

---

[1] "Migvan – research and development in CS education" works at the Technion, Israel Institute of Technology and managed by Tami Lapidot and prof. Uri Leron. 'Migvan' is the Hebrew term for 'variety'.

[2] The Hebrew version is available online http://www.csit.org.il/NCCS/TestProg/Mavo.html

[3] an LCSI product, See www.microworlds.com

first half of the first unit is mandatory (called 'introduction to programming I'), teachers can also choose what to deal with in the second half ('introduction to programming II' OR software tools).

The four general educational principles hold for all the modules of the CS curriculum, including the first module. In addition, three unique principles seem appropriate when designing a CS unit for the specific population of learners that take the 'first unit' course and the special community of teachers who teach that course.

1. Decomposition: organizing the learning material in small and focused units of contents.
2. Experience-oriented learning environment: using students' activities in the computers laboratory as the ground from which conceptual understanding can emerge (as opposed to using computers for exercising per-se).
3. Project-based learning: the learning is accompanied by the need to design, implement and present some 'mini' programming projects.

These specific principles were implemented in two variants of the first unit, both developed in the mid-1990s in Migvan – one using the Pascal programming language and the other using the Hebrew version of Logo programming language. Lately, the learning environment has been updated and new learning material have been developed using the above principles. Examples of implementing these principles in the new learning environment are presented in the next section.

## 4. TEVEL - the Israeli experience

As an example of implementing the above three principle let me take the TEVEL learning environment for the first CS unit[4]. The development of TEVEL (the Hebrew word for 'universe') has been recently completed and the first year of using the learning environment in high schools has almost finished. The first unit course is one school year long, ninety class hours, three hours per week.

Using the first principle of decomposition, we selected five main subjects for the year long unit and designed the curriculum accordingly in five modular periods, 5-6 weeks each. The first period is fully devoted to 'setting a shared base-line' taking into account the diverse background students might have both in computer literacy and in their practices of learning. Thereafter, each period is devoted to one main CS concept – constructing programs using modular procedures (2$^{nd}$ period); structured programming (3$^{rd}$ period); global variables and interactive programming using variables (4$^{th}$ period); and basic algorithmic patterns, including basic conditional algorithms (5$^{th}$ period). We implemented the decomposition one step further, by dividing each period to sub-concepts and by designing a focused set of learning activities for each sub-concept. For example, during the forth period dealing with global variables, students first engage with activities focusing at the meaning of variables (using the box metaphor); then they focus on tracing programs with assignments instructions; and then they focus on interactive programming using variables.

The second principle of experience-oriented learning environment has been already clued above, when describing the students as "engaged with activities". But TEVEL is not only an experience-oriented learning environment; it is designed in such a way that students could begin their learning in each period, in each sub-concept and in each lesson with inquiry activities, using the computer as a tool and the programming language as a medium for these inquiries. TEVEL is not only an experience-oriented, but also an experience-first environment. All class discussions are based upon and derived from students' personal experiences and experiments in the computers lab. It should be noted here that in some classes with educationally disadvantaged students, the notion of class discussion is meaningless – a single teacher can rarely manage a real discussion with most or even some students while the others are just listening. But even in such cases, students might still experience the learning environment and individually discuss their experiences with the teacher afterwards.

The third principle calls for integrating software design processes into the learning environment. There seems to be an overall agreement concerning the profits of engaging students in software projects. The issue we have faced with was what software project can we expected from novices in general, and from educationally disadvantaged high school students in particular. The programming environment we use in TEVEL, MicroWorlds Project Builder, has been chose for the first unit especially because of its advantages in the area of simple project building. MicroWorlds allows students to create dynamic and interactive programming projects even in their first steps in the world of CS, and thus lets students become active designers and active learners. Those students who still has no computer at home; those who are still unconfident in their ability to deal with technology; those who experience major difficulties in 'regular' schooling practices like writing, reading, math – can still design a simple project as early as the first period (with no programming at all but with a variety of media), then design a simple programming project using few procedures upon the middle of the learning unit, and then program a 'real' project with graphics and multimedia, present it to her or his friends and family, and touch the power of programmability.

## 5. References

[1] Y. Gal-Ezer, K. Beeri, D. Harel, and A. Yehudai, "A High school program in computer science", *Computer* 28(10), 1995, pp. 73-80.

[2] B. Haberman, "Frames and boxes: a pattern-based method for manipulating binary trees", *SIGCSE Bulletin Inroads 34 (4)*, 2002, pp. 60-64.

[4] TEVEL project began on December 2001, as a curricular project with the support of MALAM – the Israeli Center for Science Education. The author is the project leader and the project team includes Tami Lapidot, Yosefa Har-Zion and Shai Israeli.