
Abstract

This chapter presents an active-learning-based teaching model for implementation in the Methods of Teaching Computer Science (MTCS) course, which is based on the constructivist approach. This model is used in most of the offered activities in this guide. The chapter starts with motivation and rationale for using active learning in the MTCS course; then, the active-learning-based teaching model is introduced and explained, including a description of the role of the instructor of the MTCS course in the model implementation.

2.1 Introduction

As mentioned in the Introduction (Chap. 1), the main purpose of the Methods of Teaching Computer Science (MTCS) course is to prepare prospective computer science teachers (the students of the course) toward their future career as computer science teachers.

In general, courses about science teaching in the secondary school emphasize curriculum-related issues, addressing topics such as learning theories and pedagogical methods, principles for the development of scientific curricula, laboratory instruction and other investigative learning approaches, professional ethics in science instruction, and the place of science learning in the pupils' general education. All these topics are also relevant and important in the case of computer science teaching to promote the prospective teachers' professional perception.

The recommended teaching methods for the MTCS course, as described in Chap. 1, indicate that the MTCS course should be built as a teaching model. Accordingly, the course should be designed in a way that (a) promotes students' positive learning experience in a supportive teaching environment and (b) enables the

© Hazzan and Lapidot 2004 ACM, Inc. Included here by permission

students to imitate this way in their future computer science classes. To achieve this goal, the MTCS course should be based on constructivist teaching methods and implement active learning. This approach is important not only because we want the prospective computer science teachers to enjoy their learning processes and improve their understanding of computer science concepts, science teaching, and computer science education (by experiencing a variety of learning/teaching methods), but also because we want to inspire their future way of teaching in the high school.

This chapter presents an active-learning-based teaching model for implementation in the MTCS course. This model is used in this guide in most of the offered activities.

2.2 Active Learning

Confucius (551 BC—479 BC) once said:

I hear and I forget,
I see and I remember,
I do and I understand.

Active learning is widely accepted nowadays as a quality form of education. Among the many descriptions of active learning, we highlight Silberman's assertion (1996) according to it "Above all, students need to 'do it'—figure things out by themselves, come up with examples, try out skills, and do assignments that depend on the knowledge they already have or must acquire." Active learning is closely related to inquiry-based learning, problem-based learning, and project-based learning, which are all highly suitable approaches¹ for computer science learning in general and for the MTCS course in particular.

According to constructivist educators (Kilpatrick 1987; Davis et al. 1990; Confrey 1995), learning is an active acquisition of ideas and knowledge construction, rather than a passive process. In other words, learning requires the individual to be active and to be engaged in the construction of one's own mental models. As follows from the above quote by the famous Chinese philosopher, the more active learners are, the more meaningful is their understanding of what they learn. Therefore, in the design of the MTCS course, we propose educators to encourage "learners to be active in their relationship with the material to be learned" (Newman et al. 2003).

There are numerous ways to implement active learning in computer science education (see, e.g., Whittington 2004; Ludi 2005; McConnell 2005; Anderson et al. 2007; Gehringer and Miller 2009; Walker 2011; Zimundi 2012). McConnell (1996), for example, suggests several techniques, such as modified lectures, algorithm tracing, and software demonstration. In this spirit, this guide is based on the implementation of the active-learning-based teaching approach by offering a wide collection of activities to be implemented in the MTCS course in the context of computer science education.

¹ A good resource on active learning, including different types of activities, can be found in <http://www1.umn.edu/ohr/teachlearn/tutorials/active/what/index.html>**

2.3 Why Active Learning Is Suitable for Implementation in the MTCS Course?

In addition to the general argumentation about the suitability of the active-learning-based teaching approach to the MTCS course, we suggest that active learning may also promote the professional development and perception of the prospective computer science teachers, as the following justifications propose.

- *Constructivism*: Constructivism is a cognitive theory that examines the nature of learning processes. According to this approach, learners construct new knowledge by rearranging and refining their existing knowledge (cf. Davis et al. 1990; Smith et al. 1993; Ben Ari 2001). More specifically, the constructivism approach suggests that new knowledge is constructed *gradually*, based on the learner's existing mental structures and on the feedback that the learner receives from the learning environments. In this process, mental structures are developed in steps, each elaborating on the preceding ones, although there may, of course, also be regressions and blind alleys. This process is closely related to the Piagetian mechanisms of assimilation and accommodation (Piaget 1977). One way to support such gradual mental constructions is by providing learners with a suitable learning environment in which they can be *active*. The working assumption is that the feedback, provided by learning environment in which learners learn a complex concept in an active way, may support mental constructions of the learned concepts. In our case, in order to support the construction of the computer science teachers' professional perception, the prospective teachers participating in the MTCS course must have a learning environment that supports this complex mental construction. It is suggested, therefore, that active learning is naturally suited for use in such situations.
- *Wearing different hats*: In order to support the construction of the prospective computer science teachers' professional perception in the MTCS course, it is important that during the course, the students experience wearing different hats (see Fig. 2.1). At times, the prospective computer science teachers wear the hat of a high school pupil and are asked to perform "pupil assignments"; at other times, they wear the hat of the computer science (future) teacher; and yet at other times they wear the student's or the researcher's hats. As it turns out, active learning enables the switching between such situations in a very natural manner. It is also important to mention that as future computer science teachers they will have to wear different hats in their daily work (role-model, tutor, evaluator, leader, counselor, and decision maker are just a few examples) and the experience they gain in the MTCS course could help them in performing these roles.
- *Wearing the student hat*: Since the computer science material itself is usually still fresh in the student's mind, in addition to learning the content of the MTCS course itself and the construction of the professional perception as computer science teachers, the prospective computer science teachers continue, in parallel, with their mental construction of the computer science body of knowledge. From

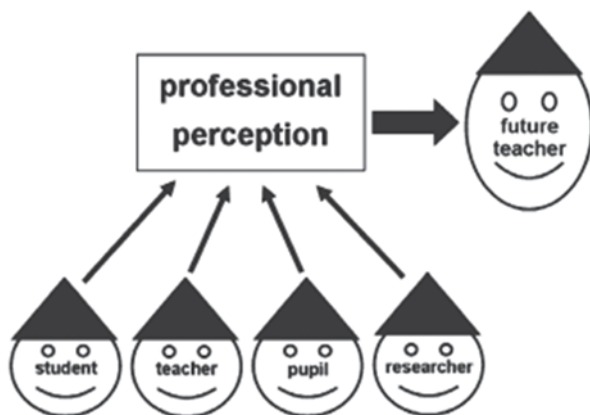


Fig. 2.1 Wearing four hats in the construction process of the prospective computer science teachers' professional perception

a constructivist perspective, in such situations, active learning is preferred over lecture-based teaching.

- *Reflection*: The prospective computer science teachers can improve the construction of their professional perception also by incorporating reflective processes into the construction process (Ragonis and Hazzan 2010). That is, by becoming reflective practitioners (Schön 1983, 1987), their comprehension of the profession of computer science education may be improved. Reflective practitioners are professionals who continuously improve their professional skills based on their on-going reflection with respect to their professional performance. Active learning is compatible with the reflective practice perspective since it provides learners with an opportunity to reflect on the activities they perform as part of their active learning.
- *Teaching methods*: Active learning enables the illustration of different teaching methods. Consequently, it enables to expose the prospective computer science teachers to different teaching methods and class arrangements. Based on the constructivist approach, the prospective computer science teachers' experience of different teaching methods in an active learning fashion, promotes their understanding of the methods' advantages and disadvantages.
- *Bridging gaps*: Active learning can bridge gaps in the teaching experience and computer science background that exist among the students participating in the MTCS course. Some of them may have stronger backgrounds in computer science; others may have more teaching experience. Since active learning enables each student to continue with the construction of his or her professional perception from his or her current professional stage, active learning can help instructors of MTCS courses overcome these variations that exist among the students.
- *High-order thinking tasks*: Last, but not least, active learning enables to offer the prospective computer science teachers tasks that enhance higher-order thinking, such as analysis, synthesis, and evaluation tasks.

2.4 Active-Learning-Based Teaching Model

So far, we have explained the rationale for the implementation of an active-learning-based teaching approach in the MTCS course. We now propose a model for active-learning-based teaching to be employed when teaching MTCS courses. This model is used in most of the activities presented in this Guide.

The active-learning-based teaching model consists of four stages—trigger, activity, discussion, and summary, focusing on a particular topic addressed in the MTCS course. The model is illustrated in Fig. 2.2 and is described in what follows. Needless to say that, when appropriate, (a) this model can be implemented in additional computer science teaching situations and (b) variations in the model implementation can be made.

First Stage: Trigger. Following the constructivist perspective, the objective of this stage is to introduce a topic with a worthwhile assignment in a nontraditional fashion (Brooks and Brooks 1999). For this purpose, the prospective computer science teachers are presented by a challenging active-learning-based trigger, an open-end activity of a kind with which they are not familiar. Specifically, a trigger should enhance and foster meaningful learning and should have the potential to raise a wide array of questions, dilemmas, attitudes, and perceptions. Following Newman et al. (2003), it is proposed that a trigger should be realistically complex and relevant for the learners. Depending on the trigger's main objective, the activity can be worked on individually, in pairs or in small groups.

In the MTCS course, a trigger can be based on different kinds of activities, such as *analyzing* a class situation, *debugging* a given computer program, *composing* a test on a specific computer science topic, *designing* an exhibition poster about a particular computer science concept, *following* a visualization or animation display for a given computer program, and so on.

One of the main objectives of introducing a new topic using a trigger is to train the prospective computer science teachers how to face and deal with open-ended and unfamiliar situations. Such situations, which are so predominant in teaching in

Fig. 2.2 Active-learning-based teaching model



general, and in computer science education in particular, require teachers to consider multiple reaction options. In order to achieve this objective, it must be possible to approach a trigger in more than one way. Furthermore, a well-designed trigger exposes the students, while working on the trigger itself, to a rich and varied mix of computer science and pedagogical aspects. Throughout the model stages, this vast collection of ideas is discussed, elaborated, refined, and organized.

Second Stage: Activity. In this stage, the students work on the trigger presented to them. This stage may be short, or it may be longer and take up the majority of the lesson. The specific period of time dedicated to this stage naturally depends on the kind of trigger used and on its educational objectives.

Third Stage: Discussion. After the required period of time, during which the students work on the trigger either individually, in pairs, or in small groups, the entire class is gathered. At this stage, products, topics and thoughts that originated during the activity stage are presented to the entire class and are discussed. At this stage, the students refine their understanding of concepts, attitudes, and ideas, as part of the construction process of their professional perception.

The instructor highlights important ideas presented by the students and emphasizes principles derived from these ideas. In order to convey the notion that no unique solution exists for most teaching situations in general, and for the specific activity presented by the trigger in particular, the instructor does not judge students' positions and opinions. At the same time, however, classmates *are* encouraged to react and express their opinions and their constructive criticism with respect to the different ideas or materials presented.

Fourth Stage: Summary. This stage of the model puts the topic into the context of the course and emphasizes the concepts that were discussed. It is managed differently than the three previous stages. First, it is significantly shorter. Second, while in the first three stages the students are the main actors, in the Summary stage, the MTCS course instructor takes front stage. The instructor wraps up, summarizes and highlights central concepts, teaching ideas, conceptual frameworks, and other related topics that were raised and discussed during the previous three stages.

The summary can be expressed in different forms, such as a framework formulation, listing connections between the said topic and other topics, concept map, and so on.

2.5 The Role of the Instructor in the Active-Learning-Based Teaching Model

The term “instructor” refers to the lecturer teaching the MTCS course. In what follows, we explain the significant role of the instructor during each stage of the proposed teaching model.

In general, the instructor has to create a supportive intellectual and emotional environment that encourages students to be fully active.

In the first stage (Trigger), the instructor constructs and presents the trigger. As mentioned earlier, a trigger must be designed very carefully as it constitutes the basis for the entire model.

In the second stage (Activity), the instructor circulates between the different groups working on the trigger, listens to their opinions, is sensitive to what they say, and encourages them to deepen their thinking. When needed, the instructor guides the students in their discussion. Though the guidance should encourage alternative thinking approaches, the instructor is advised not to dictate any position.

In the third stage (Discussion), the instructor must act as a good listener and be sensitive to crucial points suggested by the students. Specifically, the instructor should encourage the students to explain why and how they developed their suggestions, suggest exploring different options, foster reflection processes, all without passing judgment on the students' opinions. Since well-designed triggers lead to rich discussions and debates, instructors may, at this stage, find themselves navigating through various disagreements. When needed, the instructor highlights the important facets of each opinion and presents possible connections between different ideas.

In the fourth stage (Summary), the instructor sums up the ideas presented during the previous stages. This summary is organized logically so as to highlight the main messages that were raised and discussed during the lesson. When needed, the instructor adds ideas and clarifications that were not suggested by the students themselves.

As mentioned before, the active-learning-based teaching model is used throughout this Guide in many opportunities to support the construction process of the prospective computer science teachers' professional conception as computer science teachers.

References

- Anderson R, Anderson R, Davis KM et al (2007) Supporting active learning and example based instruction with classroom technology. SIGCSE'07, Covington, Kentucky, USA, pp 69–73
- Ben Ari M (2001) Constructivism in computer science education. *J Comput Math Sci Teach* 20(1):45–73
- Brooks MG, Brooks J (1999) The courage to be constructivist. *Educ Leadersh* 57(3):18–24
- Confrey J (1995) A theory of intellectual development. *Learn Math* 15(2):36–45
- Davis RB, Maher CA, Noddings N (eds) (1990) Constructivist views on the teaching and learning of mathematics. *J Res in Math Educ. Monograph 4*, Reston, VA: The National Council of Teachers of Mathematics, Inc
- Gehringer EF, Miller CS (2009) Student-generated active-learning exercises. SIGCSE'09, 3–7 March 2009, Chattanooga, Tennessee, USA, pp 81–85
- Hazzan O, Lapidot T (2004) Construction of a professional perception in the “Methods of Teaching Computer Science” course. *Inroads—SIGCSE Bull* 36(2):57–61
- Kilpatrick J (1987) What constructivism might be in mathematics education. In: Bergeron JC, Herscovics N, Kieran C (eds) *Proceedings of the 11th International Conference for the Psychology of Mathematics Education (PME11)*, vol I, pp 3–27
- Ludi S (2005) Active-learning activities that introduce students to software engineering fundamentals. *ITiCSE'05*, Monte de Caparica, Portugal, pp 128–132

- McConnell JJ (1996) Active learning and its use in computer science. *SIGCSE Bull* 28:52–54
- McConnell JJ (2005) Active and cooperative learning: tips and tricks (Part I). *Inroads—SIGCSE Bull* 37(2):27–30
- Newman I, Daniels M, Faulkner X (2003) Open ended group projects a ‘Tool’ for more effective teaching. *Proceedings Australasian Computing Education Conference (ACE2003)*, Australian Computer Society, Inc, Adelaide, Australia
- Piaget J (1977) Problems of equilibration. In: Appel MH, Goldberg LS (eds) *Topics in cognitive development*, vol 1. *Equilibration: theory, research and application*. Plenum Press, New York, pp 3–13
- Ragonis N, Hazzan O (2010) A reflective practitioner’s perspective on computer science teacher preparation. *ISSEP2010*, Zurich, Switzerland: 89–105. http://www.issep2010.org/proceedings_of_short_communications.pdf. Accessed 3 Sept 2010
- Schön DA (1983) *The reflective practitioner*. BasicBooks, New York
- Schön DA (1987) *Educating the reflective practitioner: towards a new design for teaching and learning in the profession*. Jossey-Bass, San Francisco
- Silberman M (1996) *Active learning: 101 strategies to teach any subject*. Pearson Higher Education, Boston
- Smith JP, diSessa AA, Roschelle J (1993) Misconceptions reconceived: a constructivist analysis of knowledge in transition. *J Learn Sci* 3:115–163
- Walker HM (2011) A lab-based approach for introductory computing that emphasizes collaboration. *Proceedings Computer Science Education Research Conference (CSERC’11)*, 7–8 April 2011, Heerlen, Netherlands, pp 21–31
- Whittington KJ (2004) Infusing active learning into introductory programming courses. *JCSC* 19(5):249–259
- Zimundi E (2012) Active learning for problem solving in programming in a computer studies method course. *Acad Res Int* 3(2):284–292

Guide to Teaching Computer Science
An Activity-Based Approach

Hazan, O.; Lapidot, T.; Ragonis, N.

2014, XXIV, 296 p. 10 illus., Hardcover

ISBN: 978-1-4471-6629-0