Applying Learn Team Coaching to an Introductory Programming Course

C.B. Class, H. Diethelm, M. Jud, M. Klaper, P. Sollberger
Hochschule für Technik + Architektur Luzern
Technikumstr. 21, 6048 Horw, Switzerland
{cclass | hjdiethelm | mjud | mklaper | psollberger}@hta.fhz.ch

Abstract – **In this paper we describe the didactical method "learn team coaching" and its application to an introductory programming course offered in winter term 2005/06. We present course evaluation results and conclusions.**

Keywords – curriculum development, learn team, learn team coaching, programming course

Introduction

In Switzerland as in many European countries the Bologna reform of higher education is implemented. In response to this, a majority of courses at the Lucerne School of Engineering and Architecture (HTA Luzern) have undergone a complete redesign. In doing so we had four principal goals in mind:

- First, as with the Bologna reform self studying gets a higher share in total study time, we had to provide structure and content for focused self studying.
- Second, working in interdisciplinary teams becoming customary, our students should get used to this kind of work.
- Third, an increasing number of modules attended by students following different curricula require taking into account the individual differences regarding prior knowledge.
- Fourth, students shall not only acquire profound knowledge but also skills and excellence.

Among the courses that have been designed is an introductory programming course. As a central didactical method, learn team coaching has been applied. Section 1 describes the prerequisites as well as the structure and the didactical design of this introductory programming course. In section 2 learn team coaching and its application to the programming course are presented. Section 3 summarizes evaluation results and conclusions. Section 4 concludes this paper.

1 The Introductory Programming Course

The introductory programming course gives an introduction to the object oriented programming paradigm and the Java programming language as well as to algorithms. It is attended by students of information and communication technology and by students of electrical engineering. Prior knowledge and experiences of the students range from programming experiences in different languages to rudimentary skills using computer applications. The course values 6 ECTS (European Credit Transfer System) points, corresponding to a total workload of about 180 hours. This workload includes lectures, assignments, self-study times, tests as well as a final examination of 4 hours.

Principal learning targets of the course are:

- Students know basics of object-orientation, fundamentals of Java as well as basics of javadoc and unit testing.
- Students can understand, assess and extend existing code.
- Students can understand application programming interface (API) documentation and acquire new knowledge on their own.
- Students know different basic algorithms and can translate them to Java programmes.
- Students can play a responsible part in a team.

The course is followed by a sequel introducing further concepts of Java as well as data structures.

In winter term 2005/06 the course has been held four times in parallel by four different professors.

1.1 Teaching Java

Teaching a programming language means conveying *knowledge* about programming concepts and the language's syntax and semantics as well as programming *skills*. To convey skills, the learner must get the possibility of practising. A classical way of teaching programming is to start with a simple "Hello World" application and then incrementally introduce language elements like control structures and data types. Following this approach, students build increasingly complex applications from the scratch. This approach has an inherent pitfall: students risk to loose sight of the underlying concepts and often get lost in problems related to the syntax of the programming language. Therefore, different authors have proposed new ways to teach Java, e.g. [1], [2] and [3]. We decided to base our course on the approach that described in [4]. This approach is based on the BlueJ environment (www.bluej.org) and is summarized in the following guidelines:

- "Objects first.
- Don't start with a blank screen.
- Read code.
- Use "large" projects.
- Don't start with "main".
- Don't use "Hello world".
- Show programme structure.
- Be careful with the user interface." [4]

Based on this approach, David J. Barnes and Michael Kölling have published a study book [5]. We decided to use this study book for student's self study assignments.

1.2 Didactical Method

To foster self study as well as team work, learn team coaching has been chosen as the basic didactical method for the introductory programming course. This method allows for the combination of accompanied and autonomous self study [6]. In a first step, students prepare content by studying learning material on their own. In a second phase students meet in teams and try to solve all problems that arose during the self study phase. Then the team meets with the professor. During this meeting remaining questions are discussed and solved together. The next section describes how this didactical method has been implemented in the introductory programming course.

2 Self Study and Learn Team Coaching

2.1 Teaching Elements and Structure

The introductory programming course is built around four teaching elements. Relevant elements are introduced in a classical *lecture*. Students prepare and deepen the content introduced in the lecture during the *self study* phase. In the *learn team* questions and problems of the self study phase are discussed. Besides, the team works on additional team study assignments. As a result of the learn team session, a protocol summarizing remaining problems and relevant results is prepared. The protocol then serves as the base for the *learn team coaching* session with the professor. The content of this session is, thus, directly linked to the personal learning situation of the team members. To allow for sufficient self study time, the learn team meetings take place in the week following the input lecture. Based on 8 lessons per week, this results in the structure depicted in Fig. 1.

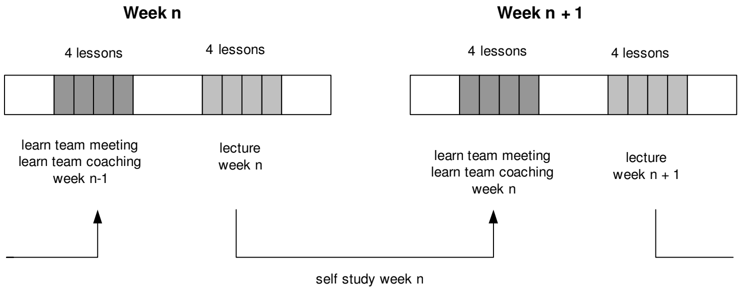

Figure 1 – Structure of the study elements

2.2 Teaching Material

As the learn team coaching teaching method has been new to most of our students, we provided an instruction document to all students. In the document we described the motivation for the choice of the learn team coaching method as well as main aspects of self study phase, learn team meetings and learn team coaching. At the end of the document we included a form to reflect the self study phase and prepare relevant results and questions for the learn team meeting.

Relevant concepts have been presented in form of lectures. The choice and sequence of the elements has been closely related to the book [5]. The book's terminology has been used for the presentations as well as some examples. Nevertheless, the presentations have been prepared to be quite independent of the book, to have their own emphasis and sometimes to go more into detail than the book. Being one important element for the self study phase, the slides of the lectures had to be of a high quality. To achieve this, we decided that each professor only prepared the lectures of some weeks and provided the slides to the others. In order to guarantee for a high quality, each input had been reviewed in the lecturer team. To do so we had one meeting per week. This meeting also provided a good possibility to synchronize the different parallel courses.

Self study assignments must be based on high quality study material. As mentioned earlier in this paper, we based our module on [5]. This book has been set for the course, thus, all students had to buy their personal copy. The book is of a high didactical quality and provides numerous self study assignments. Java projects need to be studied, modified and small programming tasks have to be fulfilled. Consequently the book provides a very good starting point for the self study phases requiring supplements only. Every supplement consisted in a self study assignment document including:

- precise learning targets of the week (operational; formulated as “you know, you can, …”).
- listing of pages to be read and exercises to be solved, as well as
- some additional small assignments or questions.

Students used the preparation form provided in the instruction document to prepare the learn team meetings.

A first task of the learn team meeting has been the collection and discussion of open questions of the team members. Then the students worked together on learn team assignments. For these meetings group rooms with access to the wireless LAN have been reserved, so the students could work in a quiet place providing the necessary infrastructure. In each session, the learn team prepared a protocol, whose structure has been defined by the professors. In this protocol they wrote open questions, problems and further topics they wanted to discuss.

The time for the learn team coaching has been limited to one lesson per group. Therefore, the topics mentioned in the learn team protocol had to be prioritized. Though the learn team coaching method proposes to work out the prioritization together with the students, the professors mostly proposed a prioritization in order to accelerate this process. The learning targets that have been formulated in the self study assignments provided a good means to conclude the learn team coaching session.

2.3 Other Course Parameters

As the book [5] is written in English and as this is not the mother tongue of our students, an English teacher gave a short instruction on “How to work with an English text book”. This instruction has been specifically prepared for working with [5]. In the first half of the term, the lecture has been restricted to two lessons to provide more “regular” study time for learn team meetings and self study.

Two written tests have been carried out to offer feedback on the students’ learning progress. The tests have been corrected and graded normally. To fulfil the final examination’s admittance requirements students had to participate at least in one test. Further admittance criteria have been to participate in 80 % of the learn team meetings as well as to lead at least one meeting. All material has been provided electronically to the students.

3 Evaluation

At the end of the term the course has been evaluated electronically. Approximately 70% of the students participated in this evaluation. As the evaluation has been carried out anonymously, it is not possible to relate evaluation results to the different programs of study (information and communication technology or electrical engineering respectively). Evaluation results have been very positive. Many students valued the didactical method while some students mentioned that they would have preferred courses that were closely guided. For many students the self study assignments have been ambitious and sometimes too time consuming. Some students mentioned that input lectures of up to 4 hours simply are too much. Many students praised the possibility of writing tests during the term.

In the final examination students of both programs of study had comparable results. This indicates that the didactical method and the material were suitable for students with differing prior knowledge.

Also for the professors the structure and the didactical method of the course have been a new experience. In fact, there have been some doubts, whether this method would work in a programming course in the first year of a bachelor programme. Professors have not been present during the learn team meetings and it was unaccustomed for them to “give up” so much control over what happens during lessons. But the experience has been very encouraging and most students have been very committed. Students have been very interested in the discussions during the learn team coaching sessions, especially as we answered *their* own questions and problems. An analysis of the final examination results has shown that students acquired a profound knowledge of object-oriented concepts whereas coding exercises were less successful. We accredit this to (a) the concept of teaching Java that puts code understanding and object-orientation into the centre and to (b) the didactical method that does not provide the possibility of fulfilling programming assignments under the observation of a professor, allowing to identify lacks related to design and coding issues as well as skills that need more training. For this reason, programming assignments as well as the implementation of a small software project are, next to self study and learn team meetings, important elements of the sequel programming course introducing more elements of object orientation and Java, basic data structures as well as basics of software engineering.

Based on the evaluation and examination results we decided to keep learn team coaching as the basic didactical method of the introductory programming course. In the next step we will concentrate on improving the teaching material with a special focus on the time required for self studying.

4 Conclusion

In this paper we describe an introductory programming course that uses learn team coaching as one central didactical element. We present the course structure as well as the teaching materials that have been used in the course. Evaluation results indicate that the didactical design of the course has been appropriate. Examination results have shown that students reached the defined learning targets. In short, the course has been successful and we could show, that learn team coaching can be applied to a programming course.

References

[1] D. Clark, C. MaNish and G. F. Royle, "Java as a teaching language – opportunities, pitfalls and solutions", Proceedings of the 3rd Australasian conference on Computer science education, 1998, pp. 173-179.

[2] R. Duke, E. Salzman, J. Burmeister, J. Poon and L. Murray, "Teaching programming to beginners – choosing the language is just the first step", Proceedings of the Australasian conference on Computing education, 2000, pp. 79-86.

[3] S. H. Moritz and G. D. Blank, "A design-first curriculum for teachning Java in a CS1 course", SIGCSE Bull., vol. 37, no. 2, June 2005, pp. 89-93.

[4] M. Kölling and J. Rosenberg, "Guidelines for teaching object orientation with Java", SIGCSE Bull., vol. 33, no. 3, September 2001, pp. 33-36.

[5] D. J. Barnes amd M. Kölling, "Objects first with Java A practical introduction with BlueJ", 2nd ed., Pearson, 2005.

[6] B. K. Pfäffli, "Lehren an Hochschulen", Haupt, 2005.