X World Conference on Computers in Education
July 2-5, 2013; Toruń, Poland

Teaching Secondary School Students Programming Using Distance Learning: A Case Study

Marek Nowicki, *faramir@mat.umk.pl*
Michał Matuszak, *gruby@mat.umk.pl*
Anna Beta Kwiatkowska, *aba@mat.umk.pl*
Maciej M. Sysło, *syslo@mat.umk.pl*
N. Copernicus University, Faculty of Mathematics and Computer Science,
Chopina 12/18, 87-100 Toruń, Poland

Piotr Bała, *bala@mat.umk.pl*
N. Copernicus University, Faculty of Mathematics and Computer Science,
Chopina 12/18, 87-100 Toruń, Poland
University of Warsaw, ICM, Prosta 69, 00-878 Warszawa, Poland

Abstract

Regional Computer Science Course is a novel approach to increasing programming skills among secondary school students in Poland. It is fully based on distance learning and individual work of students after regular school hours. In this project the achievements of students were monitored on-line through weekly programming contests. The standardized tests have been developed to monitor students' skills. In result large number of students increased their programming skills. In this paper we present idea of the project as well as its advantages and pitfalls.

Keywords

distance learning, programming, secondary school

INTRODUCTION

Programming is one of the most difficult elements in the computer science teaching. Traditionally, school curriculum focuses on the algorithmic aspects of programming, leaving the practical implementation to the student's own work. This is strongly reflected by the number of teaching hours proposed for this topic in the curriculum. In addition, the programming skills of teachers are rather weak, and thus they lack adequate support for the students. On the other hand, high school requirements and requirements for computing contests participants assume proficiency in implementation of efficient algorithms. As a result, we observe the decreasing number of students taking exams in computer science or participating in programming contests. A small number of students reaching programming proficiency make it usually on their own with little support from schools and teachers.

In order to stop these trends, and to increase number of students potentially interested in learning programming at the Faculty of Mathematics and Computer Science N. Copernicus University for number of years we have been organizing classes for high school students. In the school year 2009/2010 these activities were implemented in an innovative form based entirely on distance learning. This was basically the first project of this kind implemented in the country and Europe on a massive scale among secondary school students.

In this article we present how the project has been implemented, the advantages and disadvantages of such teaching, and we present the results obtained.

REGIONAL COMPUTER SCIENCE COURSE

Regional Computer Science Course consists of various forms of learning, the majority of them is based on distance learning. The Course is divided into three major activities which vary in terms of subject and the degree of difficulty, and thus with different prerequisites:

- Course I. Programming - 15 on-line lectures,
- Course II. Algorithms - 15 on-line lectures,
- Course III. Advanced algorithms - 20 meetings in the computer lab.

Courses I and II were carried out entirely over the Internet using a distance learning platform (OLAT 2013). Every week, on Monday at 8:00 a new lesson consisting description, the sample code in C and Java, and verification questions was made available to students. Contact with students was implemented by a dedicated discussion forums where students were reporting their problems and solutions and the lecture instructors were posting hints and explanations. The instructors were not posting ready solutions in order to motivate students to work by their own.

Each lesson has attached programming problems to check in practice the knowledge acquired by the students. Solutions in the form of source code were sent by the students to the dedicated internet system called WebContest (ZawodyWeb) (Kluszczyński et all. 2010), which automatically validates solution. Validation has been performed through compilation, execution and then check of the program results on the prepared input data sets. For the correct solution student received points depending on the difficulty of the task and the correctness of the solution.

The received points were the basis for the creation of the best students ranking. Those who gained the highest scores were invited to workshops conducted at the University.

Students' solutions for courses could be sent at any time by 10 days from the time it were published, eg. from Monday to the next Wednesday (23:00). After deadline sending of the solutions was still possible but the points gained were not counted in the classification. During the holidays and breaks the deadline was automatically extended.

Time for student's activities scheduled for one week is about 90 minutes (2 lessons), but it depends on an individual skills. Participation in classes was open to all students. They had to register in the distance learning system providing the basic data such as name, age and school. The students from the Kujavia-Pomerania Region were invited primarily to participate in the classes, but distance learning courses could be attended by the students and not students from outside of the province. Number of such students was however limited.

The syllabus of courses was adapted to the computer science curriculum, agreed with the content applicable to the graduation in computer science and Olympiad in Informatics. We have encouraged teachers to use the materials for the lessons or at the out of class activities. For this reason, access to material was not limited in time - after the release of the lessons they were available for students and teachers throughout the duration of the school year.

Course I was devoted to the computer programming and presentation of simple algorithms. Course II was carried out at the end of Course I and was devoted to learning of the basic algorithms.

Course III was conducted in parallel with the activities of Course I and II and was intended for students who have basic programming and algorithmic skills and would like to broaden their knowledge of algorithms. Course III activities were carried out using traditional classes in the computer labs of the Faculty of Mathematics and Computer Science N. Copernicus University. We have conducted two groups: one for students from Torun (Tuesdays evening) and second one for students from outside of Torun (Saturdays). Group size was limited to about 20 people due to the capacity of the laboratories.

Course I: Programming

Programming course has been prepared in the form of 15 lessons covering basic programming starting from the installation of development environment and first program up to more complex language structures. The course was developed in parallel for the two main programming languages: C (using CodeBlocks (CodeBlocks 2013) as programming environment) and Java (using NetBeans (NetBeans 2013)). Due to the particularities of each language lessons were developed in parallel covering the same or very similar range of material. The presented topics relating to the basics of programming, variables and expressions, indexed variables, control instructions (if, for, while), function calls, logical variables and expressions, operations on the texts, write and read data from a file. Course completed with two lessons devoted to the computer graphics.

During this course students get familiar with recursion, loops, different ways to finish the loop, divide and conquer method and to the selected sorting algorithms. The course was designed in such a way that the students could benefit from the previously presented material and did not have to use external information.

Course II: Algorithms

For the algorithms course we have required student's knowledge of the programming for Cycle I. In this course the main emphasis has been placed on knowledge of the basic algorithms and methods for their implementation. In this case, each lesson was illustrated with fragments of code in C and Java simultaneously.

Course was opened by lesson on the algorithms complexity. In subsequent lessons there were presented algorithms for numbers divisibility (including the Euclid's algorithm), sorting, recursive and iterative ones. One lesson was dealing with the text algorithms (string comparison, pattern search, etc.). The course presented the most important data structures such as stack, queue and array including sample applications. The last three lessons involved graphs, their representation in the form of lists and arrays, graph searches and finding paths in the graph (including Dijkstra's algorithm).

Course III: Advanced algorithms

For the selected pupils attending on-line classes we have organized learning of advanced algorithms in the traditional way of meetings in the computer lab. These lessons were devoted to solving problems from the programming contests and the Olympics in Informatics. The tasks were selected thematically, so that the various activities included a well-defined range of problems. Most of the lessons were devoted to graphs (graph representation, graph search, the coherence graphs, dichotomy, cycles in graphs, the shortest path in the graph). Separate classes were devoted to trees and their applications (the minimum tree, offset) and their applications. These themes were complemented with the problems such as Huffman codes, RSA, dynamic programming, pattern matching algorithms and random numbers and their applications. The cycle ended on dynamic and geometric algorithms.

Programming workshop

To participate in the workshop there were invited all persons who have achieved the best results in the distance learning. Participants were suggested four subjects and there was conducted a poll that allowed us to select the most interesting ones. 41 people completed the questionnaire and the results pointed to two big issues the widely popular (and 3D graphics applications on mobile phones). Finally we ended up with 3 workshops:

- 3D Graphics
- Design of dynamic websites using PHP, HTML and MySQL
- Creation of mobile applications with Java JME.

We have almost 60 pupils attended the workshops, some of them were participating in two different courses. The results were evaluated using anonymous pools and feedback forms and we have received only positive and very positive marks.

Support for traditional learning

The distance learning tools are also used as support for traditional computer classes in the secondary school. In particular we have used it as tool to gather homework assignments for the students of the dedicated class with special programme in mathematics, physics and computer science. The assignments were correlated with the problems described during the classes. Students were given 4-6 weeks to complete the homework. The solutions were submitting programs through ZawodyWeb system getting on line verification of their solution.

DISTANCE LEARNING PLATFORM

We have used OLAT distance learning tools because of the expected number of participants (a few hundred people, with 30-40 participants using the service simultaneously). The another reason was ability to use HTML for lesson preparation and storage. Educational platform was installed on a regular server (2 dual core processors, 8 GB RAM) which allowed handle users activities without any problems. Courses I and II were available as separate courses which required self-enrolment. Each course was complemented by a discussion forum and surveys (1-2 at the course). Registered participants were notified of new lessons via emails sent once or twice a week by the course coordinator.

Registrations for the courses were performed by the pupils themselves using the forms provided by the OLAT platform. This process required to have an active e-mail account because of the need to confirm registration. The registration resulted in few individual issues, but generally was intuitive for students.

Lessons were prepared in the form of HTML files using the editing tools provided by the platform OLAT. Fragments of code in C and Java included in the lessons were prepared as separate blocks using ConTEXT editor (ConTEXT 2012) which allowed for the syntax highlighting in a manner of the programming environments.

Based on the information received from the participants there were no major problems in the use of the e-learning environment, for the course authors the system was intuitive as well.

For the distance learning course special care has to be given to the volume of the lesson. The lesson length was verified by observation of selected students pursuing specific topics and by surveys conducted among participants. In particular, the survey results presented in Table 1 show that the length of the lessons, range of material and the difficulty of the tasks were chosen as planned.

The results from Table 1 confirm that lessons in the Course I were easier for students and required less time than in the Course II. In both cases the programming tasks required a comparable effort, but we must remember that in Course I, students had to solve 2 or 3 programming problems, while in Course II it was just one but more difficult and laborious task. Appearing in the poll long time required to solve the programming task can be explained as natural situation of trouble finding the error, either logical or programming. The data confirms that, in average, learning activities required about 90 minutes per week as it was assumed.

Table 1: The results of the survey (response rate) on the time required to familiarize pupils with the lesson and to solve programming tasks in the Course I (Programming) and Course II (Algorithms).

Time required to complete task	Course I lesson	Course I programming tasks	Course II lesson	Course II programming task
0-30 minutes	66%	21%	21%	13%
30-60 minutes	19%	35%	26%	43%
60-90 minutes	5%	27%	16%	15%
90 -120 minutes	1%	5%	5%	13%
120-240 minutes	8%	8%	3%	5%
Over 240 minutes		13%		8%

RESULTS

Participants

The activities were attended by over 900 participants. For that 776 people registered on the Course I and 190 to the Course II. The Course III classes regularly attended about 30 pupils selected based on the results of tests carried out on-line in the 3rd week of Course I.

In the case of educational classes carried over the Internet we have to take care of the possession of computer and Internet access, especially for students living in smaller cities and rural areas. Independent research has shown that over 80% of students in secondary schools have at home a computer with the Internet access therefore we have assumed that some students will participate in on-line classes using computers at school.

Survey conducted during the course showed that 91% of the participants attended classes at home, 23% during the regular school class, 13% at school and 10% during the additional classes organized at school. As it can be easily detected some students were using on-line access at school and at home. This observation is also supported by the statistics of the OLAT server showing the largest number of participants in the afternoon and evening. In individual cases, on-line lessons were used by teachers during regular or additional classes.

Geographical distribution of participants pointed that most pupils is from the cities with well-established educational traditions. Where education in average is less performing, students activity was reduced. This rule applied both to large cities in the region as well as small towns. Data obtained during the registration pointed to the significant higher participation rate in urban areas, but in many cases, young people living in rural areas were attending schools in the large cities and during registration they provided the location of the school rather than place they live. The geographical distribution of pupils is certainly interesting and should be better recognized.

Number of students actively participating in class Cycle I and Cycle II, namely those who have sent solved tasks, accounted for 35-40% of students who participated in the classes, ie., logged on to the system and got familiar with the content of lessons. This result is quite natural, similar figures of active pupils can be given for activities carried out with traditional methods.

In the implementation of the on-line educational activities the participation of students in the classes becomes important. For Regional Computer Science Course the attendance was on voluntary basis. We had no direct method to motivate the participation in classes. Moreover the results obtained in the programming contest were not recognized by the school authorities.

Number of students participating both actively solving problems and inactively (just getting familiar with lesson) decreased with time in a manner generally corresponding to the exponential decay. The participation is shown in Figure 1 for Course I and II. This phenomenon is natural and indicates that the course was not significant moments in which the number of students decreased significantly, eg because of difficult lesson or difficult programming task. In the Course I slightly more difficult for the students was Lesson 5, while for the Course II, a significant reduction in the number of participants is associated with the Easter break (lesson 4) that the start of baccalaureate examinations (lesson 10).

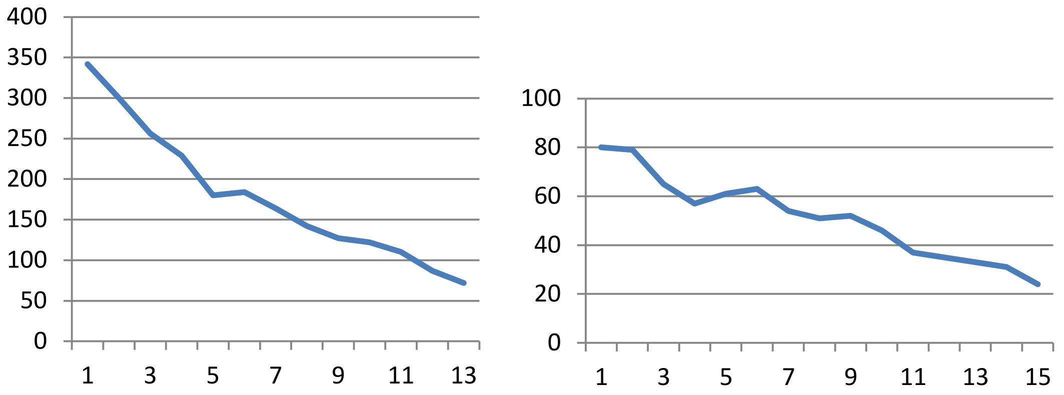

Figure 1: Number of students solving the problems of Course I.(left) and Course II (right). On the horizontal axis the lesson number is given.

Students achievements – on line learning

To check progress in the programming skills we have designed number of programming contests to test students. The tests were constructed to check the basic programming skills of students, namely:

a) the ability to write a simple program, compile it and run
b) the ability to load data from input file, a simple analysis of downloaded data and operations on them
c) the ability to use simple language structures, such as loop and conditional instructions
d) the ability to use simple data structures such as the list, or stack.

The first test was performed during the 3rd lessen to identify students participating in Course III, therefore two programming contests were given to test skills c). It turned out that the students were not able to solve both problems. In the case of a test carried out at the end of Course II we abandoned the task of checking first two components (a) and b)) components, because all the students have mastered these elements in a very good degree. These skills were necessary to complete the task to check the ability at the level c) and d) it was not necessary to separately study the degree of mastery of basic skills. Test results at the end of Course II - in fact all people solve the task implemented them in full - showed that the design of the test was optimal.

The test was designed on the principle of progressive test. In most cases this assumption has been confirmed by students and cases where there is a solution or attempt to solve the task more difficult without the solution of the easier problem are few. Therefore we can asses certain level of skills according to the level to which tasks have been solved by the student.

The tasks were formulated in a way that does not allow students to recognize that they are used to check the achievements, therefore it can be assumed that the results obtained are objective. For each test the programming contests did not repeated and thus we have tested the abilities of students rather than their memory.

Figure 2 shows that for the test at the beginning of Course I (blue) and at the beginning of Course II (red) the results are distributed around skill level c): in the first one just below, and in the second one above. For test performed at the end of Course II the results are centered at the skill level d). These results show that the test was well chosen, and checks the programming skills of students. Algorithmic skills were tested using one programming problem, however it required knowledge of a couple of algorithms. As one can see in the Figure 3, the students who have completed the Course II, solved the problem perfectly.

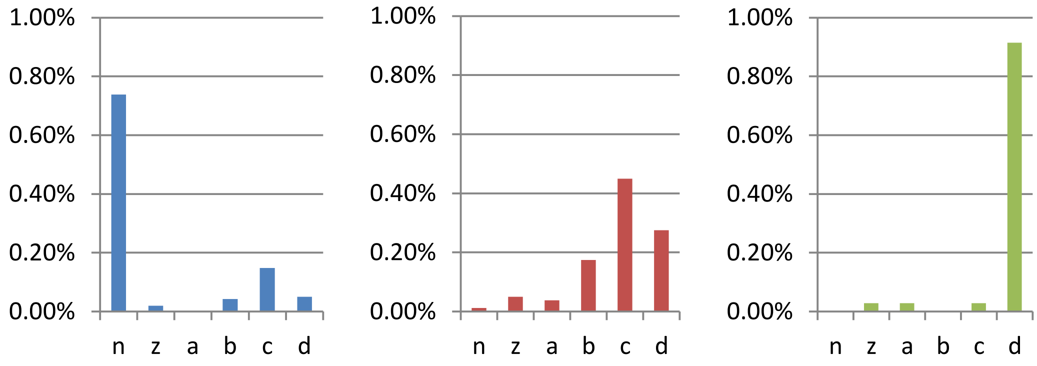

Figure 2: Number of students with specific skills normalized to the total number of students taking active participation in the course at the given moment. The results for the first contest taken at the beginning of the Course I is shown on the left, the results at the beginning of Course II in middle, at the end of Course II on the right.

We can conclude that the test solving skills of students improved after the completion of Course I. The students had better programming skills than the best ones starting Course I. It is clearly shown that people who have completed a series of programming skills I have been in excess of the best skills of students aspiring to take part in Course III. Persons who have completed courses gained a solid knowledge and programming skills.

In accordance with accepted principles of teaching diplomas from the cycle and were given to all persons Students who have obtained at least 39 points for the solution of tasks, which means sending more than 15 solutions to programming tasks received diplomas. In absolute terms 155 people have significantly raised their programming skills.

Persons who have completed a Course II increased both programming and algorithmic skills. Based on the tests is difficult to determine the degree of mastery of algorithmic skills, but high school students who have completed the Course I and II successfully participated in the Olympiad in Informatics. It means that 79 students significantly raised their skills in the algorithms.

Number of pupils with the programming skills, increased from 20% at the start of classes, up to 70% after the Course I, and to more than 90% after the Course II. Taking into account the basic programming skills, and after the cycle I had over 95% of the students. In practice this means that all students who participated in the classes had raised their skills in programming.

Similarly, the d-level skills (knowledge of algorithms) increased from 5% to over 25% after Course I to more than 90% after Course II. Again, this means that all those who participated in Cycle II classes have increased their skills in algorithms.

Students achievements – regular class with on line homework

The main purpose of using on line system for validating students programs was to motivate students to program and to minimize teacher’s work necessary to check assignments. The on-line ranking showing current status of solutions additionally motivated students to solve homework on time, or even much before deadline.

Difficulty of the problems has been asses according to the problems discussed in the classroom.

Figure 3 presents how many students solved problems. Each problem set consisted from 4 problems. The first problem set presented was easy and was treated as introduction to the on-line programming system. As it is presented in the Figure 3 (left figure) majority of students solved all problems. The two other problem sets were more difficult and required better knowledge of algorithms. The data presented in the middle and right graph were obtained for assignments organized at the end of first year and in the middle of the second year of studies. Since difficulty of problems in both assignments was similar, it is visible that programming capabilities of the students increases and higher number of pupils is able to solve all or almost all problems.

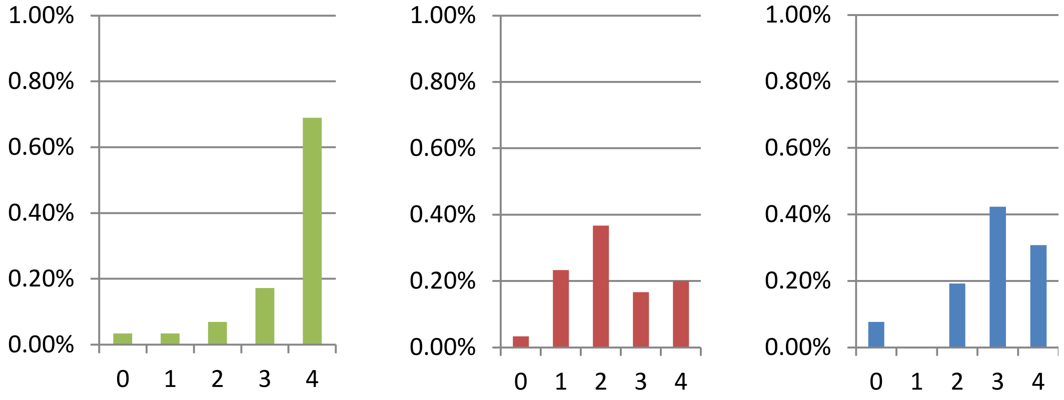

Figure 3: Percentage of students solving correctly given number of homework problems. The results for the easy problem set after the first year is shown on the left, the results of the regular set at the end of first year in middle, results of the regular problem set at the middle of second year on the right.

CONCLUSIONS

We have present activities of the Regional Computer Science Course for the secondary schools students. The course has been performed with the extensive use on the distance learning tools used in innovative way to teach programming and increase knowledge of the algorithms. The data presented show significant progress in obtaining in the programming skills and knowledge of algorithms and their implementations.

We have shown the effectiveness of this method of education and skills testing of pupils. Set of standardized tests to verify programming skills of the students has been performed and proved to be useful.

Presented results open new areas in programming teaching and will allow to improve the quality of such education and will allow to bring this education to wide students population.

REFERENCES

CodeBlocks http://www.codeblocks.org (Accessed 20 March 2013)
ConTEXT http://www.contexteditor.org (Accessed 20 March 2013)
Kluszczyński R., Mikulski Ł, Nowicki M., and Bała P. Contests Hosting Service as a tool to teach programming *Annales UMCS Informatica* vol. 9 (2009) pp. 213-224
NetBeans http://www.netbeans.org (Accessed 20 March 2013)
OLAT http://www.olat.org (Accessed 20 March 2013)

Biographies

Marek Nowicki is a PhD student at the Nicolaus Copernicus University (Toruń, Poland) since 2010, majoring in Computer Science, especially in parallel and distributed calculations using Java language. He is main developer of ZawodyWeb system. Additionally, he takes part in translating Scratch, developed by the Lifelong Kindergarten Group at the MIT Media Lab, into a Polish language.

Michał Matuszak is a PhD student at the Nicolaus Copernicus University (Toruń, Poland) since 2009, majoring in Computer Science, especially in artificial intelligence and computer graphics. He is also a teacher at secondary school with special programme in maths, physics and informatics.

Anna Beata Kwiatkowska is an academic and school teacher of informatics and mathematics, instructor at in-service courses for teachers, supervisor of school students gifted in informatics who participate in various competitions and in the Olympiad in Informatics, organizer of the Bebras Competition in Poland.

Maciej M. Sysło, mathematician and computer scientist – academic and school teacher, instructor at in-service courses for teachers, organizer of the Bebras Competition, author of informatics and ICT curricula, educational software, school textbooks and guidebooks for teachers, member of several national committees on education, Polish representative to IFIP TC3, recipient of national and international awards and grants: Steinhaus and Car (Poland), Mombusho (Japan), Humboldt (Germany), Fulbright (USA).

Piotr Bała, physicist and computer scientist – academic teacher, instructor at courses for teachers and school students, organizer of the competitions and teaching activities in programming for children. Active researcher in the area of new technologies such as high performance computing and web technologies. Has extended experience in parallel and distributed computing. Recipient of numerous national and international research grants in the area of ICT.