IPSJ Transactions on Computers and Education Vol.1 No.1 10–17 (Jan. 2015)

**Short Paper**

# Programming Education at High Schools and Universities: Design, Development, and Assessment

Eiko Takaoka[1,a) ] Takayuki Yonezu[1] Takafumi Namekawa[2]

Received: April 9, 2014, Revised: August 30, 2014,
Accepted: October 18, 2014

**Abstract:** We designed, developed, and evaluated computer programming education at high schools and universities in 2012 and 2013 to offer students an opportunity to experience practical programming. We provided a programming course to a high school, preparing content-rich materials. In order to generate motivation for learning in students, we set "to create practical applications" as a goal of the course. In 2012, only a few students could create practical applications; however, in 2013, we improved the teaching methods and by the end of the course, almost half of the participants were able to produce practical applications. In addition, some students applied their programs for the Live E! science contest and they received prizes in 2012 and two students applied and one of them received a prize in 2013. An additional notable outcome of the course that we provided was the extent to which first-year students became interested in programming.

**Keywords:** programming education, PHP programming, Live E! Project

## 1. Introduction

Many students studying computer science in college sometimes think that learning a programming language is difficult. The idea of visual programming such as Alice [1] and Scratch [2] has been proposed to enable learners to acquire a programming language intuitively by using a visual environment.

However, the following problems exist in using visual programming.

1) There is a huge gap between learning a visual, block-based programming language or drag-and-drop learning environments and learning a high-level language such as an object-oriented programming language.
2) It is hard to acquire the ability to write a real program for problem solving.
3) The goal of learning programming is to acquire only how to write programs which are correct in grammar not acquire the programming skills in a real sense.
4) Many junior high school and high school students have an image that creating a program equates to making a game and, as a result, they do not have the ability to apply their programming skills to real programming. However, learning programming is a problem-solving skill that ought to be acquired.

To address these problems, we designed, developed, and evaluated programming education at high schools and universities in 2012 and 2013 to enable students to experience practical programming. To address 1), the aim of our course was "students create practical applications". To address 2)-4), we measured the skills of the students to ascertain whether they could acquire practical programming skills by the application they produced at the end of this course.

The purpose of our practice is that we provide a programming course to get skill to create practical applications for students. We used the following definition to measure the extent to which the purpose is achieved: Students acquire the skills to create practical applications by designing and implementing the applications by themselves, taking into considering WHO would use them and for WHAT purpose. In addition, students who have enough energy submit their programs to the Live E! science contest (The details are described in Section 3.2). As a result, while in 2012, only a few students could produce practical applications, in 2013, the number of students who could produce practical applications increased.

## 2. Related Work

There have been a number of attempts to influence the views and attitudes of students toward programming through drag-and-drop learning environments such as Alice [1] and Scratch [2], which are visual, block-based programming languages designed to facilitate media manipulation for novice programmers by eliminating syntax errors, making them attractive as ways to introduce programming concepts to students. There are many case studies in which Alice and Scratch have been shown to have had a significant impact on the educational effects of learning computer science for high school students [3], [4], [5].

In recent years, there has been a decline in the number of high school and college students choosing to study computer science. Some of the reasons cited for this are that the students have negative attitudes and views about the field in that it is perceived as

1 Sophia University, Chiyoda, Tokyo 102–8554, Japan
2 Chiba Prefectural Kashiwanoha Senior High School, Kashiwa, Chiba 277–0882, Japan
a) m-g-eiko@sophia.ac.jp

being boring and tedious, requiring workers to spend many hours in front of a computer [6], [7]. To address these difficulties, Computer Science (CS) Unplugged was developed by Tim Bell and his colleagues at the University of Canterbury in New Zealand. CS Unplugged contains activities on various topics related to CS. Many studies have described the effectiveness of CS Unplugged. We also developed a package using CS Unplugged and proved its effectiveness in high school classrooms [8]. This study approaches the issue of getting high school students interested in programming from another direction. Mead et al. [9] said that traditional approaches to programming education have not taken advantage of the long record of psychological and experimental studies on the development of programming skills, and they indicated a need for a new curricular strategy for developing programming skills.

There are many examples of the creation of practical applications [10], [11], [12]. Reference [10] gives the case study of A PBL Approach using Real World Application Development between University and Local Government, which does not include the case study regarding high schools. References [11], [12] give details of academic–industrial IT education projects that were conducted from 2004 to 2006. Over the course of these three years, almost 30 projects took place; however, there are few examples of practical teaching sessions in high schools, with just 10% of high schools offering such courses.

Regarding such projects in elementary schools, junior high schools, and high schools, there are examples of educational projects that use remote sensors such as the "100 Schools Project" and the "E-Square Project" in Japan [13], [14], [15], [16], [17], [18], [19]. However, these projects focused on using the Internet and creating websites. By comparison, in our work, through making a program that uses data from remote sensors, we provide high school students with the opportunity to make practical programs.

Some high school teachers argue that the aim of learning programming should be making practical applications, not learning how to do programming [20], [21]. Thus, in this study, we provided a programming course to a high school, prepared content-rich materials, and generated motivation for learning by students by setting "to create practical applications" as the goal of the course.

Our instructional theory is based on instructional design [22], [23]. There are five phases in the model of instructional design: Analyze, Design, Develop, Implement and Evaluate. We designed the course offered in 2012 based on the students who would take the course. The students all were motivated and we designed the course schedule based on this. In 2013, we modified the program based the results of the 2012 program.

## 3. Toward Practical Learning

### 3.1 Live E! Project

The Live E! project entails independently observing weather conditions in various regions using meteorological equipment (hereinafter referred to as "digital instrument shelters") that gather information on weather through an IP network at intervals of several seconds. The aim of the project is to gain an understanding of weather conditions in various places, changes in the environment, and freak weather conditions by gathering and analyzing data derived from these digital instrument shelters. By sharing and developing this network, people can comprehend real-time weather conditions throughout the world and, as a result, lead safer lives due to being aware of possible changes in the environment. We are studying ways to use digital instrument shelters for environmental-protection measures such as countering the problems of global warming, and providing education, public services, and business applications.

### 3.2 Science Contest

Of the more than 80 digital instrument shelters that are located around the world, one is located at Chiba Prefectural Kashiwanoha Senior High School (hereinafter referred to as "Kashiwanoha High School"). Data on weather measurements are uploaded to the Live E! server via the Internet every several seconds. In 2012, we gave a lecture on how to acquire weather data at Kashiwanoha High School from the Live E! server and display the data on a web page as one example of a practical application. We offered a PHP programming course during summer vacation for students who were interested in the course. Some of the students were able to submit their programs at the inaugural Live E! science contest, which was organized by the Live E! project and sponsored by UBITEQ Co., Ltd. Applicants were required to produce one new device or software, or to analyze data from the Live E! server. High school students, university students, and young researchers were all eligible to apply to take part in the contest. Applicants were free to submit multiple programs, or several applicants could jointly compile a single program. Students who have enough energy in the course can submit their programs to this science contest.

## 4. Course for High School Students at Kashiwanoha High School

### 4.1 Program in 2012

#### 4.1.1 Outline

We designed and developed curricula and materials for programming education for high schools and universities and provided a PHP programming course for the students at Kashiwanoha High School in 2012. There are two reasons we adopted PHP for this course. One is that it allows easy development of a web application. The other is that it is a commonly used language because it can be embedded into HTML.

The preceding year (2011), a course on embedded programming and retrieving data from sensors was offered. The purpose of the course was to provide students with the opportunity to measure environmental information using different sensors. The students were very interested in programming and learned it eagerly. This is believed to be due to the fact they tried to make the practical applications, which control sensors or retrieve data from sensors via the Internet. In addition, the lecturer gave an explanation about what one can do using technology such as sensors and networks by giving specific examples. This course was a good chance for students who associated programming with creating a game to learn practical programming. Therefore, in 2012, we continued by building on this activity by having students learn programming so that they could visualize the measured data [24]. To achieve the aim of our course, in which students create practical applications, we planned that students would carry out programming that is of use to society by visualizing information, and thus develop an interest in environmental changes such as global warming and in the utilization of the network.

Additionally, we plan to develop a training theme of information training for specialized courses and research themes for project studies. We want to foster a feeling of accomplishment by students and develop their intellectual curiosity though activities in which they present what they learned after taking a course.

We designed the course schedule so that the students could create practical programs; for example, acquiring weather data from a sensor at their own high school and visualize the data at the end of the course. This course consisted of several lessons. However, the lessons were held over a short period of time (just four days during the summer vacation) and, as a result, we could not develop lessons to the extent that we could during a regular school period. Furthermore, we faced the problem of how to conduct the lessons with a mixed assortment of students because there was a wide skill gap in programming skills between them. The second-year students were learning complete programming, and some of the third-year students already had programming skills developing application using PHP or C on a Linux server.

On the other hand, first-year students were not learning programming so much. We expected that some of the students taking the course would be able to apply to take part in the science contest. In pursuing this aim, we took into account the fact that the lessons were limited to four days and so we introduced the materials to the students ahead of the scheduled lessons. The course on information and science for Kashiwanoha High School used an electronic bulletin board, referred to as a net commons, and we uploaded learning materials ahead of the lessons so the students could study the material in advance of the lessons. The students taking the course on information and science had learned processing programming in their first year, but had not started to learn complete programming until the end of the year. Therefore, there was a skill gap in programming between the students, and our first goal was to reduce the skill gap and lift students to a uniform level.

#### 4.1.2 Programming Environment

The data acquisition library was provided by Live E!, meaning that Live E! provided application program interfaces (APIs) to retrieve data or update supporting Java, PHP, and Ruby. PHP is a popular server-side scripting language, with which one can create dynamic web pages that interact with users and offers customized information and is easy to use. In addition, PHP code can be inserted directly into HTML. We adopted PHP in consideration of the fact that students had to develop a web application within just four days. Furthermore, by using XAMPP on Windows, PHP can start up a pseudo web server. For these reasons, we used PHP in our course. XAMPP is a package of free software required for running web applications and is provided by apachefriends.org.

#### 4.1.3 Preparing Materials

We provided materials for students taking the course by uploading them to their community site. **Table 1** shows the table of contents.

**Table 1** Schedule of uploading materials.

| Uploading dates | Theme of materials |
|---|---|
| May 30 | PHP statements |
| June 13 | Constants and variables |
| June 20 | Operators |
| June 27 | Arrays |
| July 4 | Conditional logic |
| July 11 | Loops |
| July 25 | Functions |

We began to upload materials on May 30. We uploaded a PHP manual describing “What is PHP?” and “What is XAMPP?” as well as basic knowledge. In addition, a teacher at Kashiwanoha High School taught the installation method for XAMPP and the execution method for a PHP program. After that, we uploaded learning materials for advance study once a week to the high school community site. Each package included slides, practice materials, and the answers to the previous practice lesson.

#### 4.1.4 Course Design and Result

The lessons were held on four occasions during summer vacation starting July 2012. The application deadline for the science contest was August 31. Thus, we planned that students learned basic programming in the first two lessons and created an application to submit to the science contest during the last two lessons. However, the students who could not develop their skills to the point where they could submit studied the practice materials by themselves.

**Table 2** shows the schedule of the lessons in 2012. In the first lesson, there were 19 participants (13 first years, 4 second years, 2 third years). In the last lesson, there were 11 participants (8 first years, 2 second years, 1 third year). Our schedule proceeded as planned. However, we still had a problem in that there was a skill gap in programming between students. We taught beginners (mostly first-year students) how to create a game involving guessing a number and a questionnaire form, and provided the time to create an application for the science contest for advanced students (mostly third-year students) in the last two lessons. Additionally, we presented a program on drawing figures and painting using the imagecreate function in PHP.

As a result, the second- and third-year students could submit their programs to the Live E! science contest, whereas the first-year students only created games and forms. Two second-year students produced one web application that displays weather data (temperature, humidity, barometric pressure, rainfall, wind speed, and wind direction) and won the programming effort award. Four third-year students produced one web application shown in **Fig. 1** and won the programming award. In this application, the data were distributed through Twitter, and visualized as a graph and on GoogleMap. All of them started work on making applications immediately after the second lesson. They examined the methods to implement what they wanted to do themselves.

In this way, some students acquired practical programming skills, but the first-year students were unable to achieve the aim of

**Table 2** 2012 Course Schedule.

| Date | Lesson plan |
|---|---|
| July 27 | First lesson:<br>● PHP statements<br>● Constants and variables<br>● Operators<br>● Arrays |
| August 8 | Second lesson:<br>● Conditional logic<br>● Loops<br>● Functions |
| August 28 | Third lesson:<br>First-year students:<br>● Exercise Lesson<br>Second-, third-year students:<br>● Creating an application for the science contest |
| August 29 | Fourth lesson:<br>First-year students:<br>● Making a game involving guessing a number, a questionnaire form and drawing figures and painting using imagecreate<br>Second-, third-year students:<br>● Creating an application for the science contest |

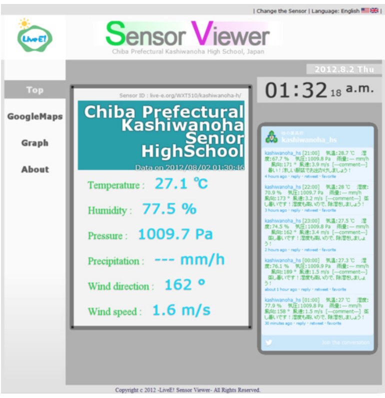


**Fig. 1** The website third-year students produced.

the course to the expected degree. However, according to a questionnaire survey completed in the fourth lesson, positive feedback was gained from the first-year students, for example, "I want to learn other programming languages. I would like to participate if there is a lecture like this next year.", "I often copied the sample program when it was difficult for me, but I learned much by thinking for myself.", and "At first, I had the impression that PHP was a difficult programming language to make a program using it; however, I don't think that now." The reason there was so much positive feedback is that students with high motivation participated in this course. The results indicate that even first-year students were interested in programming.

### 4.2 Program in 2013

In 2013, we improved on the contents we provided the previous year and constructed a Website for Learning PHP.

We created the PHP self-study website in response to the result of the previous year's questionnaire. In addition to the previous


**Fig. 2** Website for learning PHP.

year's topics, we added "How to make a questionnaire form" and "How to acquire weather data at Kashiwanoha High School from the Live E! server." Furthermore, we created the content "How web pages work" and "Difference between static and dynamic pages" because PHP is a programming language for making dynamic pages. We created animation material using FLASH to enhance the learning effect. **Figure 2** shows the website that we created for learning PHP and **Table 3** compares what learning materials we used in 2012 and 2013. The list shown in Fig. 2 corresponds to the 2013 contents in Table 3. The main improvements were that we did not teach how to draw figures and paint using imagecreate in the 2013 course and that we taught students who were not submitting applications to the science contest, mainly first- and second-year students, how to make a questionnaire form. Participants in the first, second, third and fourth lessons are 29, 24, 25 and 12, respectively. Participants in the fourth lesson were almost presenters.

In addition, we improved the teaching methods. **Table 4** shows the schedule for the 2013 course, which basically followed that of the 2012 course; however, in 2013, we set "to create practical applications" as the goal for the final lesson of the course. By the fourth lesson, students were required to make practical programs and if students could do so, they made a presentation. The fourth lesson had a dual purpose: one was that students could prepare for their presentation at the science contest, and the other was to raise students' motivation by setting aims and by learning by sharing other students' work and ideas. In addition, in the third lesson, we provided some advice about what they would create by the fourth lesson. The students who submitted applications to the contest had used the contents on our website as self-study materials. During the fourth lesson, some students presented the work that they had created in the first to third lessons. All the students who created practical applications were qualified to present. We measured the skills of the students as to whether they had acquired practical programming skills by the applications that they produced at the end of this course. **Table 5** shows the list of the applications that students produced through the end of the course

**Table 3** Learning materials in 2012 and 2013.

| 2012 | 2013 |
|---|---|
| ● Background knowledge<br>· No material | ● Background knowledge<br>· How web pages work (animation)<br>· Difference between a static and dynamic page (animation)<br>· Introduction to PHP |
| ● Basic practice PHP<br>· Introduction to PHP<br>· PHP Statements<br>· Constants and variables<br>· Operators<br>· Arrays<br>· Conditional logic<br>· Programming loops<br>· Functions | ● Basic practice PHP<br>· PHP Statements<br>· Constants and variables<br>· Constants and variables (animation)<br>· Operators<br>· Arrays<br>· Conditional logic<br>· Conditional logic (animation)<br>· Loops<br>· Loops (animation)<br>· Functions |
| ● Application of PHP<br>· No material | ● Application of PHP<br>–How to make a questionnaire form<br>· An explanation of source codes making a questionnaire form (animation)<br>· An example a questionnaire form<br>–How to acquire weather data<br>· Basic knowledge of IEEE1888[25] (animation)<br>· An explanation of source codes for acquiring weather data (animation) |

**Table 4** 2013 Course Schedule.

| Date | Lesson plan |
|---|---|
| July 18 | First lesson:<br>● Reviewing practice materials<br>● How to acquire weather data from the LiveE! server |
| July 25 | Second lesson:<br>First-year students:<br>● Basic practice PHP<br>Second-, third-year students:<br>● Creating an application for the science contest |
| August 1 | Third lesson:<br>First-year students:<br>● Making a questionnaire form<br>Second-, third-year students:<br>● Creating an application for the science contest |
| August 30 | Fourth lesson:<br>● Presentation of applications by students |

in 2013. As mentioned above, while in 2012 few students produced practical applications, in 2013, the number of students who produced practical applications increased. The authors feel that setting a clear goal to be achieved by the fourth lesson raised students' motivation. However, there were some students (almost first-year students) who did not.

Comments made in response to the questionnaire survey conducted at the end of the course are shown below. The results indicate that even the first-year students were interested in programming.

**Table 5** List of applications students produced up to the end of the course in 2013.

| Year (group Members) | Submission to the contest | Application |
|---|---|---|
| 1 | | Sushi search system |
| 2 | | Visualization of weather data |
| 2 | Submitted and won the programming award | Visualization of weather data representation using numbers and colors |
| 2 (3 students) | | Login system |
| 2 | | Visualization of students' grading data |
| 3 | | Visualization of weather data |
| 3 | Submitted | Visualization applications of weather data for Android. (Figure 3) |
| 3 (2 students) | | Visualization of weather data |

1) There are a lot of things to remember, but I really enjoyed the course.
2) I feel happy when the program is executed correctly.
3) I feel that we are able to do many things by making programs. I was very interested in this course.

In addition, a second-year student and a third-year student submitted their programs and one of them (second-year student) received the programming award as shown Table 5. Although unfortunately the third-year students could not win the prize, the program he created was suffice as practical application.

**Fig. 3** 2013 Web site for learning PHP.

## 5. Program for College Students (Sophia University)

In 2013, we provided lectures using the contents we developed for students taking the course Database, which was an elective for sophomore and junior students lasting 14 weeks at Sophia University. We provided lectures on PHP programming for two of the 14 weeks. Most of the students who took the course belonged to Department of Information and Communication Sciences, and had learned Java and C programming prior to the course. We provided the students with access to material on the abovementioned self-study website. Students were required to use both MySQL and PHP; thus, it was mandatory for them to acquire PHP skills. However, there was not enough time to teach PHP programming starting from the basics, so they used our self-study website to learn fundamental PHP programming skills. The students took quizzes and questionnaires at the end of the lectures.

The questionnaire included such questions as "Could you solve PHP problems using our self-study website?" **Table 6** shows the results. A four-point rating scale was used. Table 6 shows that about 80% of students self-evaluated themselves as using the self-study website as an auxiliary teaching material. The opinions of students who gave the evaluations "disagree" were as follows:
(1) I didn't understand which units were important or acquire the ability to write a program; however, I could understand the concepts.
(2) There were few example programs. I can't apply my knowledge to other situations.

We also asked the students, "Does this self-study website help when used on its own?" The answers are given in **Table 7**. A four-point rating scale was used.

Table 7 shows that about 98% of students favorably evaluated our website as self-study materials. The opinions of students who gave the evaluations "disagree" were as follows:
(1) I would like to use self-study materials that include fill-in-the blank questions to enhance the learning experience.
(2) Self-study materials need explanations of technical terms.

**Table 6** Evaluation as auxiliary teaching material.

| | Evaluation (Number of valid answers: 72) |
|---|---|
| Strongly agree | 7 |
| Agree | 52 |
| Disagree | 13 |
| Strongly disagree | 0 |

**Table 7** Evaluation as self-study material.

| | Evaluation (Number of valid answers: 69) |
|---|---|
| Strongly agree | 18 |
| Agree | 50 |
| Disagree | 1 |
| Strongly disagree | 0 |

Students also took a quiz. Their average score was 53 out of 100, which is a somewhat low score. The quiz is written examination including 8 questions; selecting the wrong statement, PHP overview, variable naming rule, finding the compile errors, fill in the blank (short program), declaration of array, and making two short programs using loop and conditional logic. There is difference between high-performing students and low-performing students in the making programs. The result of the questionnaire of low-performing students was as follows:
(1) Contents are very useful for self-study, however, the practices or drills are needed much more for making programs.
(2) Contents are very interesting for self-study, and I would try to learn these contents in the future as the lifelong learning, however, they were a little difficult to understand in a short time during the lecture.
(3) I need the rich description regarding words and each program.

The result shows our contents should be sufficient as self-study materials for the high-performing students and for the lifelong learning programming, however, should be improved in order to help low-performing students making programs in a short time, providing the more practices and the more explanation.

## 6. Programs for Other Universities

We provided our contents to students who took Teaching Methods in Information within the informatics teacher-training program. In this study, we focused on students majoring in informatics; however, in future work, we will focus on students not majoring in informatics or computer science; specifically, we are making plans to apply the contents to students taking a seminar in economics as a professor in the Economics Department wishes to use our contents.

## 7. Conclusion

In this study, we provided a programming course to a high school, and prepared content-rich materials to generate motivation for learning in students by setting "to create practical applications" as a goal of the course. The results show that in the 2012 project, few students could make practical applications; however, in the 2013 course we improved the teaching methods and by the end of the course, almost half of the participants produced practical applications. The authors feel the reason for this depends to a large extent on the design of the fourth lesson. By the fourth lesson, students were required to make practical programs and students who could do so made a presentation.

In addition, we produced students who applied for and received prizes from the Live E! science contest, and we got first-year students interested in programming. The students who applied for the contest grew to the point where they touched upon the breadth of what they could do to take advantage of the network and sensors in the process of using them for the contest. It can be regarded that the students who produced applications and won prizes gained skills in practical programming.

However, these courses were conducted over only a short period of time, just four days, and students had to learn programming using our content as preliminary assignments. On this point, we will need to force students to submit practice problems to counter this. We cite the following as topics of future work.

1) We measured the skills of the students as to whether they could acquire practical programming skills by the applications they produced at the end of this course. When discussed from that perspective, some students acquired practical programming skills; however, there were some students who did not. The cause of this might that there were differences in students' skill levels, grades (years), and interests. In 2014, we continued the course and first-students could create practical applications. This indicates that it is not difficult for lower-grade students to gain such skills according to the program. The analysis of 2014 activity is future work, but this is believed to be due to the fact that the lessons were focused to creating practice materials. In addition, we provided improved contents and reexamined the lecture schedule because students seem to forget the knowledge that they have obtained due to long breaks in the lecture schedule.

2) We need to design a course schedule suitable for the curriculum of individual high schools. In addition, we need to examine students of the same level by conducting a class for a single grade with a small number of students.

3) We are planning to develop education packages on information education using our contents. We will specify whether a package is for general education or professional education, and whether PHP is an appropriate language for the target students.

By examining developing packages focused on different targets, we will be able to provide practical programming for a range of students.

We have published our contents on a website (http://imbsvr102.cc.sophia.ac.jp/PHP/index.php). In addition, according to the result of our study, we will provide packages for high school teachers to use as materials.

**Acknowledgments** This paper was supported by The Science Partnership Project promoted by the Japan Science and Technology Agency.

## References


[1] Alice.org, available from ⟨http://www.alice.org/index.php⟩ (accessed 2014-01-30).

[2] Scratch–Imagine, Program, Share, available from ⟨http://scratch.mit.edu/⟩ (accessed 2014-01-30).

[3] Cooper, S., Dann, W. and Pausch, R.: Teaching Objects-first in Introductory Computer Science, *34th SIGCSE Technical Symposium*, pp.191–195 (2003).

[4] Adams, J.C.: Alice, Middle Schoolers, and The Imaginary Worlds Camps, *38th SIGCSE Technical Symposium*, pp.307–311 (2007).

[5] Maloney, J., Peppler, K., Kafai, Y.B., Resnick, M. and Rusk, N.: Programming by Choice: Urban Youth Learning Programming with Scratch, *39th SIGCSE Technical Symposium*, pp.367–371 (2008).

[6] Carter, L.: Why Students with an Apparent Aptitude for Computer Science Don't Choose to Major in Computer Science, *SIGCSE Bull.*, Vol.38, No.1, pp.27–31 (2006).

[7] Taub, R., Armoni, M. and Ben-Ari, M.: CS Unplugged and Middle-School Students' Views, Attitudes, and Intentions Regarding CS, *ACM Trans. Comput. Educ.*, Vol.12, No.2, Article 8, pp.1–29 (2012).

[8] Takaoka, E., Fukushima, Y., Hirose, K. and Hasegawa, T.: Learning Based On Computer Science Unplugged in Computer Science Education: Design, Development, and Assessment, World Academy of Science, Engineering and Technology, International Science Index 91, *International Journal of Social, Management, Economics and Business Engineerin*, Vol.8, No.7, pp.2057–2062 (2014).

[9] Mead, J., Gray, S., Hamer, J., James, R., Sorva, J., Clair, C.St. and Thomas, L.: A cognitive approach to identifying measurable milestones for programming skill acquisition, *SIGCSE Bull.*, Vol.38, No.4, pp.182–194 (2006).

[10] Inoue, A. and Kaneda, S.: A PBL Approach using Real World Application Development between University and Local Government, *IPSJ Journal*, Vol.49, No.2, pp.930–943 (2008) (in Japanese).

[11] Report of the distributive IT education project, available from ⟨http://www.meti.go.jp/policy/it_policy/it_jinzai/Sangakureport-honpen.pdf⟩ (in Japanese) (accessed 2014-08-10).

[12] Report of the distributive IT education project: Case study, available from ⟨http://www.meti.go.jp/policy/it_policy/it_jinzai/Shiryou1.pdf⟩ (in Japanese) (accessed 2014-08-10).

[13] Mind of a child growing up in the day-to-day use of groupware, available from ⟨http://web3.cec.or.jp/jissenjirei/public/jyugyou_jissen/CEC01315_0.html⟩ (in Japanese) (accessed 2014-08-10).

[14] Learning activities to tell reading the literary works through the Web page creation, available from ⟨http://web3.cec.or.jp/jissenjirei/public/jyugyou_jissen/CEC00862_0.html⟩ (in Japanese) (accessed 2014-08-10).

[15] Software development in order to enhance the awareness of human rights, available from ⟨http://web3.cec.or.jp/jissenjirei/public/jyugyou_jissen/CEC01044_0.html⟩ (in Japanese) (accessed 2014-08-10).

[16] Using the internet in the school library, available from ⟨http://web3.cec.or.jp/jissenjirei/public/jyugyou_jissen/CEC00785_0.html⟩ (in Japanese) (accessed 2014-08-10).

[17] Improvement, Development and Utilization of Web site supporting for classroom design, available from ⟨http://web3.cec.or.jp/jissenjirei/public/jyugyou_jissen/CEC01003_0.html⟩ (in Japanese) (accessed 2014-08-10).

[18] Four dimensions GIS, available from ⟨http://web3.cec.or.jp/jissenjirei/public/jyugyou_jissen/CEC00977_0.html⟩ (in Japanese) (accessed 2014-08-10).

[19] Yakushima Island meteorological observation, available from ⟨http://web3.cec.or.jp/jissenjirei/public/jyugyou_jissen/CEC00845_0.html⟩ (in Japanese) (accessed 2014-10-26).

[20] The Lecture at Programming Seminar, The Report of Total Quality Education, Vol.27, July, 2011, The Japanese Society For Quality Control, available from ⟨http://www-suzuki.inf.uec.ac.jp/tqe/docs/report27.pdf⟩ (in Japanese) (accessed 2014-10-26).

[21] The Lecture at Information Processing Society of Japan, The Report of Total Quality Education, Vol.45, Nov, 2011, The Japanese Society For Quality Control, available from ⟨http://www-suzuki.inf.uec.ac.jp/tqe/docs/report45.pdf⟩ (in Japanese) (accessed 2014-10-26).

[22] Instructional Design, available from ⟨http://www.instructionaldesign.org/⟩ (accessed 2014-08-08).

[23] Gagne, R.M., Wager, W.W., Golas, K.C. and Keller, J.M.: *Principles of instructional design*, Wadsworth Pub Co; (2004).

[24] Namekawa, T., Ochiai, H., Yamanouchi, M., Takaoka, E., Nakayama, M., Esaki, H. and Sunahara, H.: Practical Programming Education for measuring and visualization of Environmental Information on a High School of Informatics, *IPSJ SIG Technical Report*, 2012-CE-116(16) (2012) (in Japanese).

[25] Esaki, H. and Ochiai, H.: GUTP and IEEE1888 for Smart Facility Systems using Internet Architecture Framework, *1st IEEE Workshop on Holistic Building Intelligence through Sensing Systems* (*HOBSENSE*), cooperating with IEEE DCOSS, Barcelona, Spain (June 2011).

**Editor's Recommendation**

In this paper a challenging research on the introductory education of computer programming for high school and university students is presented. Two factors in the research, i.e., practical but not so difficult programming targets and self-learning support are shown to be the key points of this educational field, with some supporting results. These factors have widely been recognized effective for the education, though it has been difficult to prove they are really effective. The paper definitely shows that steady and constant effort for constructing educational environment is actually fruitful, in the actual educational field. Many educational researchers, including teachers, may well be encouraged by the result of this work.

(Satoru Kawai, Division of Informatics, Graduate School of Arts and Sciences, The Open University of Japan)

**Eiko Takaoka** received her D.Eng. degree from Keio University. She joined the Foundation for establishment of Chitose Institute of Science and Technology and was Assistant Professor, Full-time lecturer, and Associate Professor in Chitose Institute of Science and Technology and is currently Associate Professor of Sophia University. She is also a part-time lecturer of International Christian University and Meiji Gakuin University. She received Yamashita Memorial Award (2006) and Contribution Award (2014) from IPSJ. She is a member of IEICE Japan, IPSJ, DBS Japan, JSISE, ACM, AACE, Meteorological Society of Japan, The Association of Japanese Geographers and Society of Heating, Air-Conditioning and Sanitary Engineers of Japan. Her research interests include Computer and Society (education, environment and medical services), Web application, Programming and Databases.

**Takayuki Yonezu** was in the Graduate School of Science and Technology, Sophia University and received his M.Sc. (information science) degree from Sophia University in 2014. He joined a leading company.

**Takafumi Namekawa** received his M.Ed. degree from Chiba University in 1989. He was with Chiba Prefectural Shonan Senior High School (1989–1998), Kashiwa Municipal Kashiwa Senior High School (1998–2006), Chiba Prefectural Kashiwa Nishi Senior High School (2006) and is currently with Chiba Prefectural Kashiwanoha Senior High School from 2007. He is also in the Japanese National Headmasters' Association for "Informatics," which is a technical subject in senior high school.