Supplementary JAVA Programming Course for e-Learning with Small-Group Instruction

Eiko Takaoka, Yuji Osawa

Abstract**—**We have designed and implemented e-Learning materials for a JAVA programming course since 2004 and have found that "normal" students, meaning motivated and capable students, can successfully learn the course material taught in a fully online manner. However, for "weaker" students, meaning those lacking motivation, experience, and/or aptitude, the results have been unsatisfactory, and such students thus fall into the supplementary category. From 2007 to 2008, we offered a face-to-face class with small-group instruction for the weaker students, while we provided the fully online course for the normal students. Consequently, we succeeded in helping the weaker students to overcome their programming phobia and develop the ability to create basic programs.

Keywords**—**e-learning, JAVA Programming Course, Small-Group Instruction, Supplementary.

I. Introduction

COURSES that utilize Internet technology to replace or supplement traditional classroom learning are becoming increasingly common in Japan, as elsewhere, for the convenience and effectiveness they offer. For students, the benefits are "access anytime, anywhere" and "self-paced learning." For teachers, despite the initial workload involved in creating the programs, such courses provide the opportunity to focus on individuals according to his or her particular needs for more effective and efficient teacher-student interaction. However, e-Learning has some disadvantages: it is not easy to keep students motivated to learn, there is a lack of interaction and exchange of ideas between students, and instructors are not able to monitor the students' study patterns.

Tutoring has become an indispensable component of higher education in some countries [1], [2]. Since a supplementary course often parallels the normal curriculum study for graduation in Japanese universities and colleges, web-based learning is frequently employed as an effective tool to lessen the burden on teachers. Extensive research has been conducted to demonstrate the general effectiveness of e-Learning, and therefore this is not the primary topic of the present paper [1], [3], [4].

In our laboratory, we have been designing and implementing Java programming e-Learning materials since 2004. From 2004 to 2006, we conducted the following types of classes for introductory Java programming courses: face-to-face, fully online, and hybrid of both face-to-face and online. We found that for motivated students, the fully online style is more suitable than the traditional style in which students create computer programs every week with help from the lecturer and the teaching assistant [5]. However, the results have been unsatisfactory for students who lack motivation, experience, and/or aptitude (hereafter called "weaker" students), and such students thus fall into the supplementary category. From 2007 to 2008, we offered hybrid-style programming courses supplementing e-Learning with small-group instruction. As a result, the weaker students received better grades, and the number of students who failed decreased compared to the previous year [6]. Briefly, we provided two styles of classes:

- Hybrid-style course for the weaker students.
- A normal, fully online course for students not having any problems academically (hereafter called "normal" students).

Until the midterm examination, both the normal and the weaker students took the normal course and the same examination. After the midterm examination, we provided a hybrid-style course for the weaker students in addition to the normal course. The weaker students were identified as follows:

- Their online score for the midterm exam was less than 15, which means they could not create a basic Java program.
- They had poor access to e-Learning contents and did not take the exam.

After the midterm examination, we conducted six face-to-face classes for the weaker students, and attendance was required. As a result, we were able to raise the students' levels of programming ability through the combination of e-Learning and small-group instruction.

The remainder of this paper is organized as follows. First, we present related research, including the e-Learning system at the Chitose Institute of Science and Technology (CIST) and the contents that we developed. Second, we describe the programming focus for CIST students. Third, we outline the Java programming course. Then the paper presents the evaluation method and the results from the evaluation of our supplementary course. Finally, we conclude the paper with some remarks.

II. Related Research

The topic of this paper is supplemental small-group instruction for online classes. Supplemental instruction is a cooperative learning model designed to improve student performance and retention in courses with high failure and withdrawal rates. The poor performance typically occurs in

Eiko Takaoka is with the Sophia University, 7-1, Kioi-cho, Chiyoda-ku, Tokyo 1028554 Japan (phone: +81-3-3238-4123; fax: +81-3-3238-4123; e-mail: m-g-eiko@sophia.ac.jp).

Yuji Osawa is with the Chitose Institute of Science and Technology, 758-65, Bibi, Chitose, Hokkaido 0668557 Japan (phone: +81-123-27-6001).

large classes. At the same time, keeping normal online learners engaged is a challenge, and many studies point out that some e-Learners become less confident, lack motivation, and finally drop out of the course.

In this section, we first describe some studies on supplemental instruction and some small-group teaching and then mention some studies on keeping online learners engaged.

Supplemental instruction was originally a cooperative learning model and has been used for years to improve student performance and retention in courses with high attrition rates [7]. The supplemental instruction model developed during the mid-1970s was successful in improving course grades and student retention [8].

Several studies have found supplemental instruction to be effective in improving student performance in a cross-section of undergraduate courses, including biology, chemistry, economics, engineering, history, accounting and mathematics and statistics [9]-[12]. However, [11] detected no significant differences in SI and non-SI student grades for political science and calculus courses.

In contrast, the importance of small-group teaching to foster engagement with educationally effective practices has been shown [9], [13]. Allowing students to spend more time in smaller groups encourages a greater sense of belonging and solidarity, which aids student engagement. Highlighting peer-group tutoring as a valuable instrument in the integration process, [9] support this premise. Many of these previous studies include both small groups in a large class and small class sizes.

In these previous studies, few studies conducted the supplemental small-group instruction for online classes and in this paper we present descriptive data on the effectiveness of supplemental small-group instruction for online classes. Reference [14] stated that the students' background and characteristics, the curriculum and its contents, the educational environment, and confirmation that they could achieve results are factors affecting students' lack of motivation. Reference [14] also pointed out that the first semester's score after entrance to an educational institution is very important for them, as is content amount and content quality.

Reference [15] pointed out that some students lose their confidence and leave online courses. Reference [16] suggests that even company employees can easily lose their motivation.

Reference [17] pointed out that the important factors of e-Learning are attrition and retention, and defined a "dropout" as one who never completes the course of study and never returns, and a "stopout" as one who leaves but comes back later to finish. In our paper, we use the term "weaker students" as students who become stopouts. Reference [17] also proposed that successfully reducing the dropout rate allows better allocation of delivery resources as well as providing an improved return on their investment to student. We have conducted an annual questionnaire survey to improve the allocation of delivery resources and believe we have improved the return on our investment.

Reference [18] investigated the UK Open University and identified the need to limit the amount of content-specific information and activity in the early stages. She also advocates the simplification/limitation of navigation options early on and the release of content as learners gain mastery of the basic skills. This would reduce some of the cognitive overload that learners experience in the early stages.

Reference [19] also suggested that first-time e-Learners can often experience cognitive overload in the early stages of an online course, which likely contributes to high dropout rates, particularly for those withdrawing within the first few weeks of the course.

Most of today's students have already developed online skills, so they can begin to use the e-Learning system soon after they enter the university. In addition, at CIST, we conduct an orientation session and at least two face-to-face classes to teach the use of e-Learning in order to reduce the cognitive overload as much as we can.

Much e-Learning content for programming education is used as an aid in face-to-face classes. At Shinshu University in Japan, for example, an Internet school provides a learning environment for many working people, and they can learn through lectures by using effective materials (texts, pictures, animations, drills, and videos) and support systems [20]. Reference [21] discussed the design, implementation, and evaluation of online course materials for an undergraduate community information elective used in combination with face-to-face teaching. Sasaki et al. [22] proposed an introductory object-oriented programming course using WebCT. The course was designed as a self-learning course in which each student can learn at his/her own pace, and students are expected to adopt the appropriate attitude and be able to learn by themselves. In addition, Sasaki et al. applied a story-based structure in the teaching material contents to construct virtually situated learning and incorporate content familiar to the students.

Willging's and Johnson's online survey collected data from students who dropped out of an online program [23]. Logistic regression analysis was used to compare various factors between those who remained in the program and those who dropped out. The results show that the students' reasons for dropping out of an online program were varied and unique to each individual, although these reasons were not very different from those typically given by students dropping out of traditional face-to-face programs. Although specific reasons such as technology issues, lack of human interaction, and communication problems are clearly unique to the e-Learning environment, there was no evidence to suggest that these were the primary reasons for online students leaving the program. For the dropout students, the much proclaimed adage of "learning anytime, anywhere" does not seem to apply.

In accordance with these previous studies, we tried to help weaker students in the early stages of our online course by providing them with face-to-face classes, and then examined the effectiveness.

III. A Real-Life Example of E-Learning

CIST is a private university now in its 12th year of existence. In 2003, MEXT highly praised our program and adopted it as "a support program for contemporary educational needs." (http://www.chitose.ac.jp/english/award/award_for_distinctive.html).

CIST has maintained its e-Learning project since 2003. Our program includes the following:

- The development of effective classes through shared knowledge
- Remedial classes with a combination of the e-Learning system and small-group tutorials
- Implementation of new information education for improving career prospects
- Student participation in making and developing the e-Learning system and content.

In addition, we have been developing an e-Learning system covering math, English, physics, chemistry and information/media. However, in this paper we introduce the e-Learning materials for Java programming that we designed and implemented.

A. Java Programming e-Learning Materials

We designed the content for an introductory Java programming course based on the results of a questionnaire from students. Every year we revise the content of the previous year. The questionnaire includes such questions as "What concept is difficult to understand?" and "Which item requires an explanation by animation?"

1. Texts, Animations, and Videos

The content of our course is listed below. Each item consists of several texts, animations, and videos. For example, for the content in 8, "Programming Practice," we provide 12 texts, 4 animations, and 6 videos.

- Object-oriented Paradigm
- Java
- Variables of Primitive Type, Expressions and Operators
- Class and Instance
- Class Definition
- Fields
- Methods
- Programming Practice
- Interim Summary
- Inheritance I
- Inheritance II
- Arrays
- Abstract Class

Each text is written in HTML. We also provide two types of animations made with Macromedia Flash to explain concepts that are difficult to learn by just reading text. One animation type helps students understand the concept. Fig. 1 shows an example animation that explains the relation between an actual parameter and a formal parameter and the process of returning a value from a method when the method is called. The second animation type is a detailed explanation for a long program. Fig. 2 shows an example animation in which the corresponding explanation is displayed when students move their mouse over the lines of a long program.

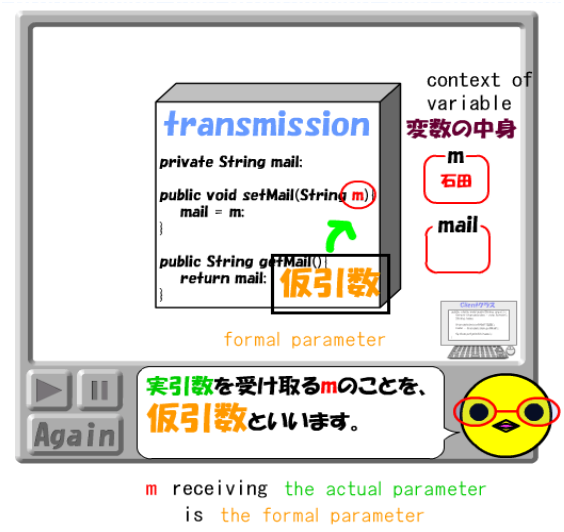

Fig. 1 An example animation explaining an actual parameter and a formal parameter

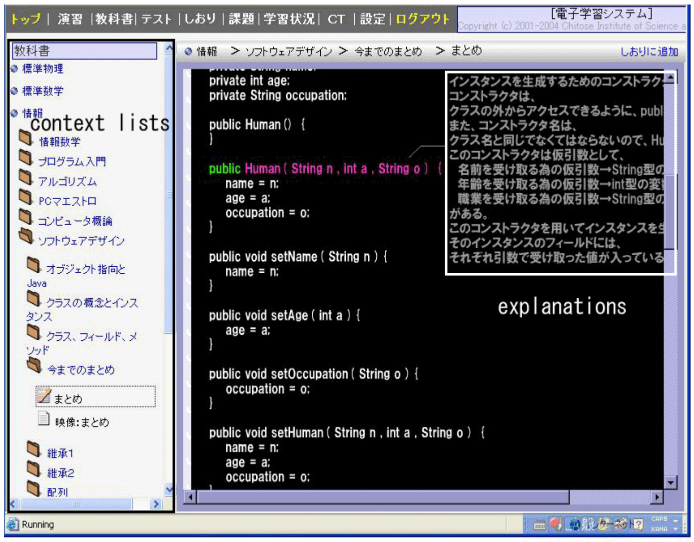

Fig. 2 An example animation explaining a program

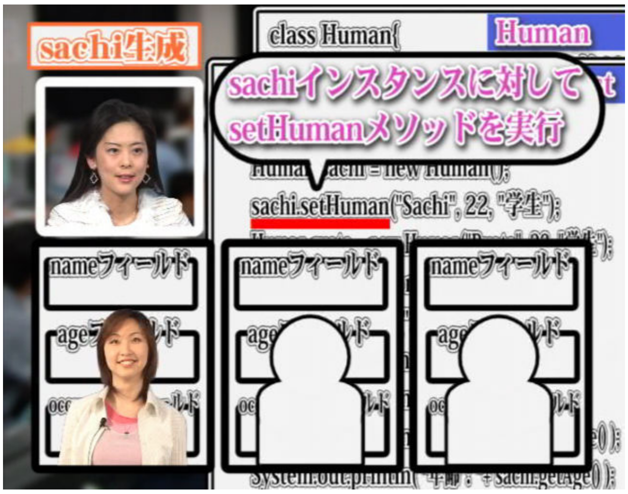

Fig. 3 An example of a video

Video materials have many advantages. Using video helps students understand complex concepts such as Object, Class, Message Passing, and Inheritance. We tried various measures. For example, we chose to feature our students as instances to make learners understand the process of creating an instance. A scenario and video were developed exclusively and the length of each video was approximately 15 minutes. Telops (subtitles) were added to attract the students' attention. The use of telops is a technique familiar to students who watch Japanese television (see [24]).

It is also very helpful for students to watch an operation procedure repeatedly. Fig. 3 shows an example of a video in which, by tracing the execution of a program, students can understand that creating an instance with a constructor means that the field value of each instance changes.

2. Drills

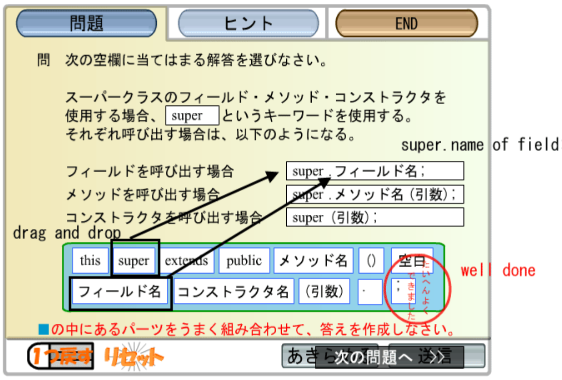

Fig. 4 An example of a drill (puzzle style)

We provide drills for each item to check the students' comprehension. There are two modes in the drill content: the practice mode and the assignment mode. Students can practice at their own pace in the practice mode, but they have to achieve a 100% success rate within a time deadline in the assignment mode. The drills contain fill-in-the-blank questions and questions about the program execution results. Answer styles include multiple choice and puzzles (dragging and dropping the appropriate answer to a text box from the multiple-choice choices). When students fill in the blank and click the "submit" button, the system indicates whether their answer is right ("well done," as shown in Fig. 4 or "wrong"). Students can also quickly confirm the percentage of questions answered correctly.

B. Student Experience with the e-Learning System

Soon after new freshmen take the orientation training for using our e-Learning system, they use the system in some classes. Primarily, the e-Learning system is used to supplement face-to-face classes. For example, a required course for freshmen is math, which covers differentials and integrals, and students are expected to use the e-Learning system in private study (e.g., lecture reviews and preparation for exams). English is another required course for freshmen. To obtain the skills equivalent to the second level of the English proficiency examination (Japanese eiken), 41 units are required; each unit includes one text, 10 comprehension drills, 10 dictation drills, and 10 grammar drills. Students use the e-Learning system in the classroom and are required to achieve a 100% success rate on the drills within 14 weeks.

C. Programming at CIST

We next provide a brief explanation of the Japanese school year, where the first term is in the spring and the second term is in the fall. At CIST, students major in science and technology. In the Department of Applied Photonics System Technology (APST), students learn hardware, software and systems to strengthen their capabilities in our information-oriented society. In the APST department, the main field of study is optical technology based on electronics, system engineering, and information engineering, and the students are required to master object-oriented design and Java programming.

All CIST freshmen learn how to use computers and increase their computer literacy soon after entering. APST sophomores learn the C language in the spring. From 2004 to 2006, juniors learned Java programming (the course name is Software Design) only in the spring. Since 2006, students can take this subject in the autumn of their sophomore year or the spring of their junior year. The reasons for this curriculum change are as follows.

- It is better that students learn Java programming immediately after learning C programming.
- Although teaching the course twice a year overloads the teachers' schedules, the fully online style mitigates the overload.
- Students can acquire credit because of the more flexible schedule.

Almost all sophomores opt to take the course in autumn. If they receive a poor grade, they can then re-take the course in the following year.

IV. Java Programming Course Design

In this section, we present an overview of the Java programming course in CIST for the years 2004 to 08. We also state the aims, goals, and objectives, followed by an explanation of the course schedule and course structure.

A. Overview

The introductory Java programming course is the only subject that offers fully online instruction with college credit [5]. Before taking this subject, students take the C language course in their sophomore spring semester in the hybrid style that combines e-Learning and face-to-face classes.

In designing and developing the e-Learning resources for the Java programming course, we attempted to address issues on a continuing basis (see Table I for an overview of years 2004–08). We prepared the following content to achieve fully online learning: 50 texts, 12 animations, 77 drills, and 13 movies.

TABLE I
The Evolution of the Teaching Style of the Java Programming Course at CIST

Term	Year	Method	Students	1st Period	2nd Period	Attendance
2004 Spring	Junior	Face-to-face	126	Lecture	Students created programs on the computer	Required
2005 Spring	Junior	Hybrid	144	Self-study via e-Learning	Teacher summarized, and students created programs	Required
2006 Spring	Junior	Fully online 1)	125	Attendance required for the initial orientation session, two face-to-face classes, and mid & final examination (out of 14 weeks)		
2006 Autumn	Sophomore	Fully online 1)	123			
2007 Autumn	Sophomore	Hybrid	122	Hybrid-style course comprising the supplementary course and a fully online course for passing students		
2008 Autumn	Sophomore	Hybrid	96			

1) Similar to the 2007–2008 normal course, as described in detail in the main text

The introductory Java programming course is a required subject for junior students. Each class consists of two 90-minute periods per week and lasts for 14 weeks. There are two examinations: a midterm and a final. Each exam consists of one written part and one practical part (making a program on the computer "on the spot"). Grading is based on the examination results. The details are described later in the section "Evaluation Methods". The aims of this class, called Software Design, are as follows.

Aims: To have students learn the following programming concepts.

- "Class" (the abstract characteristics of a thing (object), including the thing's characteristics (its attributes, fields or properties) and the things it can do)
- "Object" (a particular instance of a class)
- Within the program, using a "method" should only affect one particular object
- Inheritance (a mechanism for the re-use of several specifications)

Although we educated the normal students effectively using the fully online style, we found that it is still important to provide well-designed coursework for all students and face-to-face sessions for the weaker students [5].

We gave special care to students when we conducted the fully online style. We examined whether there are notable relationships between content access and students' grades in the hybrid style. The text access rate, the number of text accesses, the drill access rate, and the number of drill accesses were used as factors for the evaluation. The study found that there were strong correlations between all factors of content access and students' grades. We showed students the data on this correlation to strengthen their motivation. In addition, we monitored students' access behavior of e-Learning contents and identified "at risk" students and gave them encouragement.

B. Course Schedule

Appendix I shows the course schedule and contents for both the weaker and normal students, who followed the same schedule until the midterm exam. This subsection gives detailed explanations regarding the contents, followed by course details.

C. Course Structure

As stated before, we provided two styles of classes:

- hybrid style for the supplementary course
- fully online for the normal course

1. Normal Course

Until the midterm examination, all students took the normal course and learned the e-Learning content by following the schedule shown in Appendix. Students were required to attend classes on the initial orientation session, two face-to-face classes, and midterm and final examinations during the 14-week semester. The details are as follows.

- Teaching assistants were available either through consultation by e-mail or in person during a weekly two-hour consultation period.
- Students were provided the weekly learning schedule, shown in Appendix, beforehand, along with the following expectations:
  - Studying the text, animation, and video materials
  - Achieving a 100% success rate on drills
  - Submitting programming assignments

Students with a capacity for learning much more were able to adjust their schedule according to the nature of the work and their own pace.

2. Midterm Examinations

All students (i.e., normal and weaker students) took the same examination, which covered the following.

- Written exam (40 points out of 100)

Fill-in-the blank questions: important words (e.g., object-oriented programming, inheritance, class, and method) and important programming keywords (e.g., extends, public, and private)

Programming: making a program using a given example program.

- Online exam (60 points out of 100)

Students write programs in accordance with a given specification within two hours on-site. There are three steps.

Step 1. Defining a new class given in the instructions including

- Declare several fields
- Define a constructor that takes no arguments
- Define some methods
- Create several instances
- Print the field value of each instance.

Step 2. Adding to the program of Step 1

- Define one constructor that takes arguments
- Define new methods.

Step 3. Designing and defining a new class given in the instructions that is completely different from that in Step 1 and Step 2.

3. Supplementary Course

After the midterm examination, we provided the hybrid-style course (e-Learning and face-to-face classes) for the weaker students, whereas the normal course continued for students who did not need tutoring. Both styles followed the schedule shown in Appendix before the midterm examination. After the midterm examination, the weaker students adopted the supplementary course schedule. For the supplementary course, we conducted six face-to-face classes with required attendance. The weaker students were identified as follows.

- Students with an online score less than 15, which means that they could not complete Step 1.
- Students who had poor access to e-Learning contents and did not take the exam.

We provided a special course for these students, in which they learned in almost the same way as the normal course, except they could design and implement his/her own programs. In addition, students who had more ability could try the same assignments as the normal course. The course design details for the supplementary course are as follows.

1st week: reviewing for the midterm online exam

2nd week: designing and implementing at least two classes, with individual implementation of his/her own class

3rd week: designing and implementing a program using inheritance individually

4th week: face-to-face class with normal course students; making a program in accordance with the given specification

5th to 6th week: making a program in accordance with the given specification in the computer classroom. (students in the normal course were required to submit the same assignments)

Supplementary course students were required to attend each class and six teaching assistants were available at all times.

4. Final Examinations

Both the normal and weaker students took the same examination, which covered the following material.

- Written exam: same style as that in the midterm examination
- Online exam: same style as that in the midterm examination. The details are as follows.

Step 1. Defining two new classes, including

- Declare several fields
- Define a constructor that takes no arguments
- Define some methods
- Create several instances
- Print the field value of each instance

Step 2. Defining a superclass of the two classes they created in Step 1.

Step 3. Defining a new subclass extending the existing class they created in Step 2.

V. Evaluation Methods

To evaluate the effectiveness of the supplementary course, we examined autumn 2007 and autumn 2008 because both of these classes were composed of sophomores only and because we offered the supplementary face-to-face course after the midterm examinations while providing the e-Learning course for normal students.

We took different approaches: three objective assessments and one subjective assessment. First, as the objective assessment, we used the two-sample independent t-test to compare the average midterm and final scores of the students in the normal course and the students in the supplementary course.

Second, as another objective assessment, we analyzed the midterm exam score, the final exam score and the final grade of the supplementary course students. As mentioned above, the midterm exam included a written exam (40 points out of 100) and an online exam (60 points out of 100). The grading policy was as follows.

A: 70, B: <70 and 45, C: <45 and 15, D: <15, where the grade = midterm exam score 0.3 + the final exam score 0.7 to show the higher importance placed on the final exam.

Third, as another objective assessment, we counted the number of students who received “D” for the final grade, where grade “D” was a “failing” grade from 2006 to 2008.

Finally, as a subjective assessment, we conducted the questionnaire survey on the e-Learning environment. The survey was administered on paper in autumn 2007 and autumn 2008, when the hybrid-style system was first used.

VI. Objective Assessments

A. Autumn 2007

Twenty-six of 127 students took the supplementary course. Almost all of the students in the supplementary course attended the face-to-face classes and began making programs by receiving lectures in the small-group instruction. Five students had poor attendance and three of them did not come to the university. Of these 26 students, two had poor attendance in this subject only, but three were dropouts from the university.

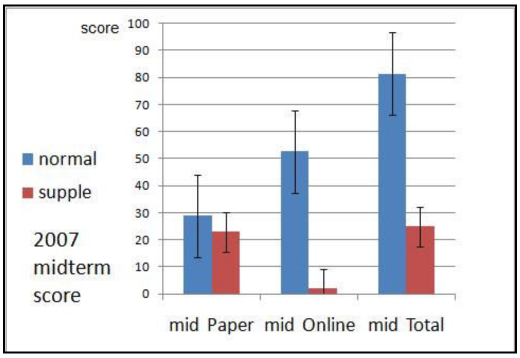

Fig. 5 Midterm average examination scores of the normal course and the supplementary ("supple") course in 2007

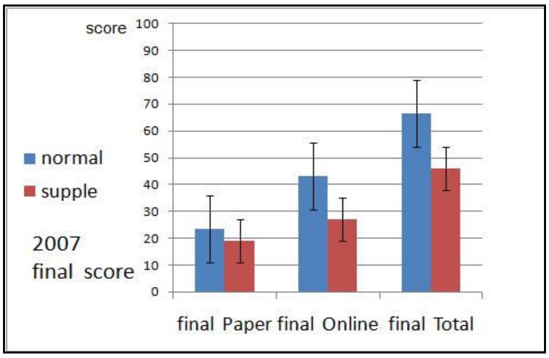

Fig. 6 Final average examination scores of the normal course and the supplementary ("supple") course in 2007

Figs. 5 and 6 show the comparison between the mean paper, online, and total scores of the normal course and the supplementary course for the midterm examination and final examination. The mean, standard deviation, t-value, and p-value of the students in the normal course and in the supplementary course are shown in Table II. According to Figs. 5 and 6, Table II and the two-sample t-test results, the scores of the students in the normal course tend to be much higher than those of the students in the supplementary course. However, the differences between the courses in the final exam were less than those in the midterm exam. Particularly, although the midterm online mean score of the supplementary course is much lower than that of the normal course, the difference is reduced in the final exam. This result indicates that the supplementary course students studied harder and gained much more programming skill.

TABLE II
PERFORMANCE SCORES OF STUDENTS WHO TOOK THE NORMAL COURSE (N = 96) OR THE SUPPLEMENTARY COURSE (N = 21) IN 2007

	Normal course mean score	Normal course SD	Supplementary course mean score	Supplementary course SD	t-value (0 DOF)	p-value
mid Paper	28.81	7.85	22.88	8.75	2.70 (110)	.01
mid Online	52.63	10.70	2.00	4.40	18.39 (111)	.01
mid Total	81.44	15.51	24.88	11.60	13.73 (111)	.01
final Paper	23.39	7.25	18.95	9.06	2.37 (115)	.01
final Online	43.22	21.05	27.10	23.42	3.06 (115)	.01
final Total	66.60	25.60	46.05	28.36	3.21 (115)	.01

Table III shows the midterm exam, final exam, and final grades of the students in the supplementary course. Of these 26 students, two had poor attendance in this course, and three were dropouts from the university. Grade D means "fail." The final grade consists of the midterm examination (paper and online) and the final examination (paper and online), as mentioned above. Supplementary course students who took the final exam obtained a higher score than they did on the midterm exam (particularly online), as shown in Table III. Almost all students obtained the minimum programming skill because the online score of 20 means that they completed Step 1. Three students (Students 5, 11, and 12) did not create any program. Although Students 5 and 11 received grades of C according to the grading policy, they were not particularly successful at making programs. In contrast, surprisingly, the two students with poor attendance in the first half of the semester (Students 20 and 21) attended the supplementary class and received high scores. The other three students with no attendance in the first half of this semester also obtained the minimum programming skill. Student 19 was graded as "D" under normal grading; however, he received the minimum programming skill because he received a 20 on the online examination and therefore received a "C".

TABLE III
PERFORMANCE OF STUDENTS IN THE SUPPLEMENTARY COURSE IN AUTUMN 2007

	Midterm Paper (40%)	Midterm Online (60%)	Final Paper (40%)	Final Online (60%)	Grade
Student 1	18	0	24	42	B
Student 2	6	0	2	20	C
Student 3	23	0	14	40	C
Student 4	34	0	21	43	B
Student 5	15	0	21	0	C
Student 6	26	0	23	15	C
Student 7	16	0	9	40	C
Student 8	28	13	32	16	C
Student 9	37	7	31	60	A
Student 10	8	0	11	15	C
Student 11	24	0	18	0	C
Student 12	29	0	7	0	D
Student 13	18	0	21	43	B
Student 14	22	0	11	15	C
Student 15	35	15	28	44	B
Student 16	27	1	28	44	B
Student 17	absent	absent	29	20	C
Student 18	absent	absent	24	20	C
Student 19	absent	absent	22	20	C
Student 20	absent	absent	24	60	B
Student 21	absent	absent	19	60	B

B. Autumn 2008

In the autumn 2008 course, the number of students who were required to take the supplementary course was 40 out of 96 students. Four students did not return to the university; that is, 36 students took the supplementary course.

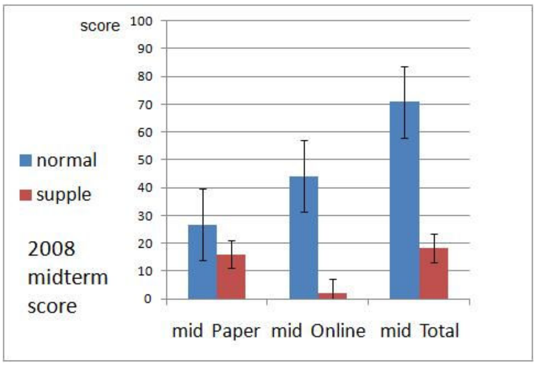

Fig. 7 Midterm mean examination scores of the normal course and the supplementary ("supple") course in 2008

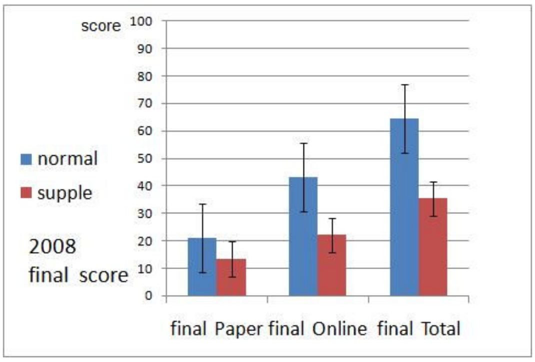

Fig. 8 Final mean examination scores of the normal course and the supplementary ("supple") course in 2008

Figs. 7 and 8 show the comparison of the paper, online, and total mean scores between the normal course and the supplementary course for the midterm examination and final examination. The mean, standard deviation, t-value, and p-value between the students in the normal course and in the supplementary course are shown in

According to Figs. 7, 8, and Table IV and the two-sample t-test results, the scores of the students in the normal course tend to be much higher than those of the students in the supplementary course. However, the differences in the two types of courses decreased in the final exam in comparison with the differences in the midterm exam. This result indicates that the supplementary course students worked harder and gained the required programming skill.

TABLE IV
PERFORMANCE SCORES OF STUDENTS WHO TOOK THE NORMAL COURSE (N = 56) OR THE SUPPLEMENTARY COURSE (N = 36) IN AUTUMN 2008

	Normal course mean score	Normal course SD	Supplementary course mean score	Supplementary course SD	t-value (0 DOF)	p-value
mid Paper	26.64	7.26	16.03	7.53	6.73 (90)	.01
mid Online	44.13	14.80	2.19	4.44	16.56 (90)	.01
mid Total	70.77	19.87	18.22	8.80	14.97 (90)	.01
final Paper	21.14	7.20	13.28	6.47	5.26 (90)	.01
final Online	43.25	22.56	22.06	21.24	4.45 (90)	.01
final Total	64.39	27.30	35.33	23.06	5.23 (90)	.01

Table V shows the midterm exam, final exam and final grades of the students (except for the four students with poor attendance). Compared to autumn 2007, the final paper exam scores were lower; however, the final online scores were higher than the midterm online scores. Almost 70% of the students received 15 as the online score, which means they attained the minimum programming skill. Three students (Students 10, 30, and 35) received the maximum online score. Unfortunately, 12 students received an online score of 0, and 8 of these students failed the class.

TABLE V
PERFORMANCE OF STUDENTS IN THE SUPPLEMENTARY COURSE IN AUTUMN 2008

	Midterm Paper (40%)	Midterm Online (60%)	Final Paper (40%)	Final Online (60%)	Grade
Student 1	16	0	9	19	C
Student 2	16	0	9	54	B
Student 3	16	0	17	0	C
Student 4	21	12	16	40	B
Student 5	3	0	4	42	C
Student 6	13	13	17	0	C
Student 7	8	0	19	33	C
Student 8	23	0	24	44	B
Student 9	16	0	13	0	D
Student 10	32	0	21	60	A
Student 11	19	0	26	14	C
Student 12	32	12	14	43	B
Student 13	21	0	19	15	C
Student 14	25	0	9	0	D
Student 15	17	0	10	0	D
Student 16	20	0	19	34	B
Student 17	6	9	11	0	D
Student 18	21	0	23	19	C
Student 19	17	0	16	38	B
Student 20	12	0	8	15	C
Student 21	26	0	9	15	C
Student 22	7	0	12	44	B
Student 23	22	0	25	0	C
Student 24	8	10	4	20	C
Student 25	15	4	17	16	C
Student 26	8	0	21	0	C
Student 27	22	0	14	37	B
Student 28	11	0	9	17	C
Student 29	7	0	4	0	D
Student 30	absent	0	15	60	B
Student 31	8	0	4	0	D
Student 32	12	0	13	0	D
Student 33	17	13	5	19	C
Student 34	13	0	4	0	D
Student 35	12	8	11	60	B
Student 36	absent	absent	7	19	

C. The Number of Students who Received Bad Marks

We counted the number of students who received a "D" or a 0 score on the final exam online score from autumn 2006 to autumn 2008. (The target students were all sophomores. Before then, the students were juniors and we cannot compare them easily.) Table VI shows the number of students who received a "D" or 0 score on the final online examination, and the total number of students is also given.

TABLE VI
THE NUMBER OF STUDENTS WHO RECEIVED A "D" OR 0 FROM AUTUMN 2006 TO AUTUMN 2008

The number of students who received --	2006	2007	2008
"D"	25 (20%)	1 (1%)	8 (9%)
0 score on the final online exam	23 (19%)	Supplementary course: 3 (3%) Normal course: 6 (5%)	Supplementary course: 12 (13%) Normal course: 6 (6%)

In 2007 and 2008, the number of students who failed decreased compared to the number of students in 2006. However, in 2008, the number of students who received a 0 score on the final online score increased compared to the number in 2007. The total number of students of the supplementary course and the normal course combined was approximately the same as that in 2006.

Unfortunately, the number of students who received a 0 score on the final online examination was almost the same as that in 2006. Even some normal course students received a 0 score. The leading cause could be attributed to the number of students who took the supplementary course. Thirty-six students were too many for the small-group instruction in the supplementary course.

The number of students who received a "D" in 2008 was fewer than that in 2006, although the number of students who received a 0 score on the final online exam was almost the same in both 2006 and 2008. The reason could be that the students who received a 0 score on the final online exam did not receive a good mark on the paper exam, and so the final mark was not a passing score (15) in 2006. In comparison, some students met the passing score by only the final paper examination and the midterm examination.

VII. SUBJECTIVE ASSESSMENT: QUESTIONNAIRE AND INTERVIEW

As a subjective assessment, we conducted the questionnaire survey about the e-Learning environment. First, as shown in TABLE VII, we distributed the questionnaire on the learning method of e-Learning and tallied the responses for the total number of respondents in both the normal and supplementary courses and the number of respondents in the supplementary course only. The total number of respondents in both classes and the number of respondents in the supplementary course in 2007, and the total number of respondents in both classes and the number of respondents in the supplementary course in 2008 were 111, 17, 61, and 23, respectively.

TABLE VII
QUESTIONNAIRE FOR ALL STUDENTS ABOUT THE E-LEARNING METHOD

Question: Please choose the appropriate-Learning method for you in the case using e-Learning					
		2007		2008	
	Alternatives	Total (111)	Supple. (17)	Total (61)	Supple . (23)
1	Prefer traditional face-to-face class rather than e-Learning	32 (29%)	8 (47%)	15 (25%)	13 (57%)
2	Under the condition that a teaching assistant is available weekly in person for two-hour consultation periods.	70 (63%)	7 (41%)	36 (59%)	9 (39%)
3	Only e-Learning is enough. The weekly two-hour period consultations by the teaching assistant are not required.	7 (6%)	0 (0%)	7 (11%)	1 (4%)
4	No response	2 (2%)	2 (12%)	3 (5%)	0 (0%)

In both 2006 and 2007, almost all of the weaker students could not complete their assignments before taking the supplementary course and so we conducted a more detailed survey for only the supplementary course students, as shown in Table VIII. Q2 allowed multiple answers. However, in 2007, after the final class, we asked the questions in person.

TABLE VIII
MORE DETAILED SURVEY FOR STUDENTS IN THE SUPPLEMENTARY COURSE ONLY

Q1: Why did you not complete assignments before taking the supplementary course?			
	Alternatives	2007 Supple. (20)	2008 Supple. (15)
1-1	Could not get motivated without enforcement	10 (59%)	8 (53%)
1-2	Could not understand how to make a program and did not feel like asking questions without a face-to-face class	7 (41%)	7 (47%)
1-3	I don’t like e-Learning	3	N/A
	Q2: How did you learn before taking the supplementary course?		
2-1	Learned according to course schedule	N/A	8
2-2	Asked friends	N/A	11
2-3	Did not learn	N/A	8
	(for students who chose 2-3 in Q2: Why did you not learn?		
	Did not know how to learn	N/A	5
	Could not get motivated	N/A	3

The results (TABLE VII) indicate that half of the weaker students need a face-to-face class and most of them need a place where they are able to ask questions to someone on a regular basis, although the “someone” did not always have to be the lecturer: a teaching assistant was suitable. Table VIII indicates that the weaker students do not ask questions without the face-to-face class, even if they cannot understand the content. They can partially study by being forced to study, but they do not know how they should study. So, this result indicates that small-group instruction is helpful for some students. The reason for the three students who answered they did not like e-Learning was a personal issue, not a technical issue; that is, they simply preferred to listen to a lecture in a face-to-face class.

After each course, we asked another question: “Did you understand programming?” All students answered “Yes, I prefer a face-to-face class rather than only e-Learning,” “Yes, I could ask questions and understand how to make a program,” and “Yes, the teaching assistants helped me understand programming.”

VIII. DISCUSSION

As shown above, we took two different approaches to assess fully online vs. hybrid-style learning: three objective assessments and one subjective assessment. First, we compared the mean scores of normal course students and supplementary course students for the midterm and final examinations using the two-sample independent t-test. Consequently, the normal students' scores tended to be much higher than those of the supplementary course students. However, the differences between both courses in the final exam decreased compared to the differences in the midterm exam. Particularly, although the midterm online mean score of the supplementary course was very much lower than the midterm online mean score of the normal students, the difference decreased in the final exam. This result indicates that the supplementary course students studied harder and obtained the required programming skill.

As another objective assessment, we analyzed the midterm exam score, the final exam score and the final grade of the supplementary course students and counted the number of students from 2006 to 2008 who received a “D,” which means “fail.” In 2007 and 2008, the number of students who failed decreased compared to that in 2006. However, in 2008, the number of students who received a 0 score on the final online score increased compared to that in 2007. The leading cause could be the number of students who took the supplementary course: 36 students could be too many for small-group instruction in a supplementary course.

Also, we cannot ignore that, even in the normal course, approximately 5% of students received a 0 score on the final online examination. The categorization of weaker students needs to be more precise. This is an issue for future research.

According to the results of the questionnaire for 2007 and 2008, the students could understand the programming concepts and create programs by themselves. This indicates that small-group instruction can boost weaker students and shows a definite positive effect that mitigates poor performance. However, if the help by the teaching assistants is not sufficient, then it is not possible to provide the needed instruction. Although "studying" should not be something one is forced to do, it is necessary to encourage the weaker students.

IX. Conclusions

We conducted various types of classes for an introductory Java programming course from 2004 to 2007: face-to-face, hybrid-style, and fully online. We found that for motivated students, the fully online style is more suitable than the traditional style, in which students create programs in the computer room with help by the lecturer and teaching assistants every week, because e-Learning allows each student to learn at his or her own pace. These students stated that the quantity and quality of course materials were adequate and they could learn to make programs by the fully online style. In addition, such motivated students can further advance their skills.

We offered a face-to-face class with small-group instruction for weaker students after a few weeks of fully online learning, while we provided only the fully online learning for normal students. Consequently, we successfully helped the weaker students overcome their programming phobia and develop the ability to create basic programs. In this paper, the students were CIST students majoring in science and technology, and so it can be assumed that technology issues can be excluded as the cause of leaving an online course. As [23] pointed out, for the dropout students, the much proclaimed adage of "learning anytime, anywhere" does not seem to apply our questionnaire results show that e-Learning is simply not appropriate for some students. For them, we have to conduct traditional face-to-face classes with small-group instruction. In other words, e-Learning should be applied only for students who can learn at their own pace, anytime, anywhere, and for teachers who can include tutoring with small-group instruction.

Although a teacher can concentrate on weaker students in a small group, it is difficult to give the appropriate lecture to various levels of students in a large class. However, more appropriate teaching is possible by adopting most students to fully online. Our study could be very effective in evaluating a teaching framework. Of course, well-designed content, guidance, and a support system for answering students' questions are essential in any course. In addition, it is important that lecturers monitor the students' progress and provide appropriate advice.

Appendix I

Course Schedule showing Content Type and Title of Contents

Weeks	Category	Content type	Title of Contents	Assignment
	Initial Orientation Session			
weeks 1 to 2	Object-Oriented Paradigm	Text	Object-Oriented Paradigm	1
		Text	Object-Oriented Programming	
		Text	Object-Oriented Programming: Purpose	
		Text	Object-Oriented Programming: Difference from procedural language	
		Video	Object-Oriented Paradigm	
		Glossary	Object-Oriented Paradigm	
	Java	Text	Java	
		Text	Java VM (Virtual machine)	
		Video	Java	
		Glossary	Java	
	Variables of Primitive Type, Expressions and Operators	Text	Variables	
		Text	Variables of Primitive Types	
		Text	Variables of Primitive Types: Declaration	
		Text	Variables of Primitive Types: Assignment	
		Text	Variables of Primitive Types: Assignment	
		Animation	Variables of Primitive Types	
		Text	Expressions	
		Text	Operators	
		Text	Variables of Primitive Type : Casting Conversion	
		Text	Casting Conversion	
		Glossary	Variables of Primitive Type, Expressions and Operators	
weeks 3 to 4	Classes	Text	Classes	2
		Animation	Classes and Instances	

Weeks	Category	Content type	Title of Contents	Assignment
		Text	Modifiers	
		Video	Classes	
		Glossary	Classes and Instances	
	Class Definition	Text	Class Definition	
		Animation	Packages	
		Animation	Class Definition	
		Glossary	Class Definition	
	Fields	Text	Field Declaration	
		Animation	Public and Private	
		Animation	Field Declaration	
		Glossary	Field Declaration	
	Methods	Text	Method Definition	
		Text	Arguments	
		Text	Overloading	
		Text	Main Method	
		Text	System.out.println	
		Animation	Method Definition	
		Glossary	Method Definition	
	1st Face-to-Face Class			3
weeks 5 to 8	Programming Practice	Text	Instance and Constructor	4
		Text	Access to Fields and Methods	
		Animation	Formal Parameters, Actual Parameters and Return Values	
		Text	Operator Associativity and Precedence	
		Text	Variables of Reference Types～Reference Types and Primitive Types～	
		Text	Variables of Reference Types～The Class String～	
		Text	Variables of Reference Types～The Class Object～	
		Animation	Variables of Reference Types	
		Video	Variables	
		Text	The Practice of Programming: Editor, Saving Files, Compiling and Execution	
		Text	The Practice of Programming: Indentation	
		Text	The Practice of Programming: Comments	
		Text	The Practice of Programming: Name	
		Video	The Practice of Programming	
		Text	The Practice of Programming: Example 1	
		Text	The Practice of Programming: Example 2	
		Video	Programming Example	
		Video	Constructors	
		Glossary	The Practice of Programming	
	Summary	Text	Summary	5 to 6
		Video	Summary	
		Video	The Practice of Programming 1	
		Video	The Practice of Programming 2	
		Support Material	The Practice of Programming: Support Material	
		Glossary	Summary	
	Midterm Examination			
weeks 9 to 11	Inheritance	Text	Inheritance	7
		Animation	Superclasses and Subclasses	
		Animation	Inheritance and Classes	
		Text	How to use Inheritance	
		Text	Examples of Inheritance	
		Text	Overriding	
		Text	Access to Fields and Methods of Superclass	
		Animation	Method Overriding	
		Video	Inheritance 1	
		Glossary	Inheritance	
weeks 12 to 14	Inheritance and Constructor	Text	Inheritance and Constructors	8 to 10
		Video	Inheritance 2	
		Glossary	Inheritance and Constructors	
	Array	Text	Arrays	
		Text	Declaration and Creation of Arrays 1	
		Text	Declaration and Creation of Arrays 2	
		Text	Length of Arrays	
		Text	Array Access	

Weeks	Category	Content type	Title of Contents	Assignment
		Text	Array Initializer	
		Text	Examples of Arrays	
		Video	Arrays	
		Video	Static	
		Glossary	Arrays	
	Second Face-to-Face Class			11 to 13
	Final Examination			

ACKNOWLEDGMENT

This research was carried out at the Chitose Institute of Science and Technology. We would like to express our gratitude to previous CIST laboratory students and faculty members for their help and encouragement.

REFERENCES

[1] Ono, H. (2005). Basic academic ability construction of Japanese universities students ---- observation from placement tests, A Development of Placement Test and E-learning System for Japanese University Students, National Institute of Multimedia Education, Japan, pp. 1–6.

[2] Campanella, S., Dimauro, G., Ferrante, A., Impedovo, D., Impedovo, S., Lucchese, M. G., Modugno, R., Pirlo, G., Sarcinella, L., Stasolla, E., and Trullo, C. A. (2008). E-Learning platforms in the Italian universities: the technological solutions at the University of Bari, WSEAS Transactions on Advances in Engineering Education, 1(5), 12–19.

[3] Cho, K., Okuda, Y., Takeno, S., and Otsu, A. (2006). A preliminary study on dropout factors in e-Learning of English – learners' motivation, preference of learning activities, and language-learning strategies, LET Kyushu-Okinawa Bulletin, 6, 33–47.

[4] Al-Jarf, R. (2005). The effects of online grammar instruction on low proficiency EFL college students' achievement, Asian EFL Journal, 7(4), 10th Article.

[5] Takaoka, E., and Ishii, W. (2007). A dedicated online Java programming course: design, development and assessment. In Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education (E-Learn), 484–494.

[6] Takaoka, E. (2008). Java programming for supplementary students. In Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications (EDMEDIA), 4435-4440.

[7] Widmar, G. (1994). Supplemental instruction: from small beginnings to national program. New Directions for Teaching and Learning, 60, 3-10.

[8] Burmeister, S., Kenney, P., and Nice, D. (1996). Analysis of effectiveness of SI sessions for college algebra, calculus, and statistics. In Research in Collegiate Mathematics Education II, Providence, RI: American Mathematical Society, 145-154.

[9] Etter, E. R., Burmeister, S. L., and Elder, R. J. (2001) Improving student performance and retention via supplemental instruction, Journal of Accounting and Education 18(4), 355-368.

[10] Kochenour, E., Jolley, D., Kaup, J., Patrick, D., Roach, K., and Wenzler, L. (1997). Supplemental instruction: an effective component of student affairs programming. Journal of College Student Development, 38, 577-586.

[11] Warren, B., and Tonsetic, R. (1997-1998). Supporting large classes with supplemental instruction (SI), Journal of Staff, Program, and Organizational Development, 15, 47-54.

[12] Webster, T., and Dee, K. (1998). Supplemental instruction integrated into an introductory engineering course. Journal of Engineering Education, 87, 377-383.

[13] Longden, B. (2006). An institutional response to changing student expectations and their impact on retention rates, Journal of Higher Education Policy and Management, 28(2), 173-187.

[14] Joo, Y. J., Park, S. H., and Cho, H. K. (2008). Construction and validation of a drop-out scale in cyber university. In Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications (EDMEDIA), 914-919.

[15] Morgan, C. K., and Tam, M. (1999). Unraveling the complexities of distance education and student attrition, Distance Education, 20(1), 96-105.

[16] Frankola, K. (2001). Why online learners drop out, Business Services Industry, http://findarticles.com/p/articles/mi_m0FXS/is_10_80/ai_79352432/

[17] Martinez, M. (2003). High attrition rates in e-Learning: challenges, predictors, and solutions, http://www.elearningguild.com/pdf/2/071403MGT-L.pdf

[18] Salmon, G. (2004). E-moderating: The Key To Teaching and Learning Online, Second Edition, RoutledgeFalmer, London.

[19] Tyler-Smith, K. (2006). Early attrition among first time eLearners: a review of factors that contribute to drop-out, withdrawal and non-completion rates of adult learners undertaking eLearning programmes, MERLOT Journal of Online Learning and Teaching, 2(2), June.

[20] Kunimune, H., Niimura, M., Wasaki, K., and Fuwa Y. (2005). The current activities and the evaluations of graduate school of science and technology on the Internet, Shinshu University, Transactions of Japanese Society for Information and Systems in Education, 22(4), 264-271.

[21] Kendall, M. (2001). Teaching online to campus-based students: the experience of using WebCT for the community information module at Manchester Metropolitan University, Education for Information, 19, 325-346.

[22] Sasaki, S., Watanabe, H., Takai, K., Arai, M., and Takei, S. (2005). A practice example of object-oriented programming education using WebCT. In Proceedings of the 2005 Conference on Towards Sustainable and Scalable Educational Innovations Informed by the Learning Sciences: Sharing Good Practices of Research, Experimentation and Innovation, 871-874.

[23] Willging, P. A., and Johnson, S. D. (2004). Factors that influence students' decision to drop out of online courses, Journal for Asynchronous Learning Networks, 8(4), 105-118.

[24] Takaoka, E. (2006). A learner-friendly and effective multimedia content studying application for IT on broadband networks. In Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications (EDMEDIA), 2636-2643.