# Importance of Course Content and Classroom Design in Project-Based Learning in Programming Courses - Case Study of a Liberal Arts College, Ferris University in Japan -

Natsuko Uchida
Ferris University
Yokohama, Japan
uchida@ferris.ac.jp

Chuchi Montenegro
Silliman University
Dumaguete, Philippines
chuchismontenegro@su.edu.ph

Yasushi Kuno
University of Electro-Communications
Tokyo, Japan
y-kuno@uec.ac.jp

Katsuhiko Kakehi
Waseda University
Tokyo, Japan
kakehi@waseda.jp

**Abstract**

The demand for a technology savvy workforce in the 21st century has stirred a number of challenges on how to best equip "bun-kei" (social sciences and humanities) and "ri-kei" (natural science and engineering) graduates with computational thinking and computer programming skills. For higher education to cope with such demand in March 2016, the Science Council of Japan released a reference standard named "Reference standard on educational curriculum organization for quality assurance in higher education - informatics field." Its purpose is to unify all branches of informatics as a field of science for creating value from information. This means that the delivery of IT skills to "bun-kei" university students will not be limited to office application tools but rather will include an understanding of how software is developed so as to answer the demands of rapid societal change. "Bun-kei" graduates are expected to possess skills in communication and problem solving, and

develop an understanding of information utilization, software development and project management processes. Thus the authors designed the curriculum as 50% programming learning and 50% project experienced contents.

In this paper, we discuss the detail of the course design, class room design, and students' outcomes of the course. We also show examples of the effect of lab setup and layout on students. The same course design was delivered to both sections in two different laboratory setups: a) desktop computers arranged in a typical classroom arrangement, and b) laptops arranged in a meeting-style layout.

**Keywords**

Programming Education, Project Based Learning(PBL), Course and Classroom Design

## 1. Introduction

The continuous evolution of information and communication technologies (ICT) requires that all members of the society should have the necessary skills to use these technologies and should possessed enough knowledge on how to access information in order for each individual to efficiently function in the information society. Education in the information society enunciates that when learning is sought, it also focuses on the objective to transcend data, to know, to comprehend, to learn, and to be able to create. Thus fluency in other related technology and especially programming, a core skill in Information Technology, is strongly desirable. This fact leads to many Colleges and Universities to offer programming classes even in outside of IT-related courses. However, in "bun-kei"[1] universities such as Ferris University, programming classes presents a great challenge to its students because students aren't accustomed to science and related topics.

To attract some students in an actual problem-solving activities, one of the authors have been exposing some Ferris University students to participate in a number of community based IT-related projects. It was observed that those students of the course actively worked in such projects to solve problems that might be found in their life, and learned unconsciously elementary programming.

Thus, the premise was that if programming course are designed. Thus, the premise was that if programming course are designed in a problem-solving-focused manner, such course can be viable and an effective alternative to "bun-kei" students in learning programming content. Additionally, actual skills required when Ferris University student graduates will be more of problem solving skills and project management skills, rather than just purely programming skill itself. Thus the authors designed the curriculum as 50% programming learning and 50% project experienced contents.

This paper discusses the detail of the course design, class room design, and students' outcomes of the course.

[1] In Japan, since high school, there are two major categories humanities and Social science ("bun-kei" in Japanese) and engineering and science ("ri-kei").

## 2. Background of The Current IT Education in Japan

### 2.1. IT Education in General

Due to the larger coverage and positive effect of the Internet since the late 1990's, information processing education in higher education institutions is facing another challenge. In 2002, the Ministry of Education, Culture, Sports, Science and Technology (MEXT) through Information Processing Society of Japan (IPSJ) conducted a survey on the actual condition of IT education in higher education institutions. Result of the survey shows many schools have set lessons as elective courses rather than required courses wherein course content was education focused on operation, keyboard operation, Web browser, and office productivity tools.

In the primary and secondary education in the late 1990s, Internet education in these levels was promoted as a "100 school project", a project which was initially participated by 111 schools and educational institutions. This being followed by the revision of the high school curriculum guidelines in 1999 stating Information courses should be mandatory by 2003 and that all high schools students were to take information technology education. In Junior High School, Information Technology was added as a unit of the existing home economics engineering by virtue of revision of the 2002 curriculum guidelines. MEXT positioned it mainly as learning that information technology supports the industry.

In elementary and secondary education, the curriculum is structured following the curriculum guidelines, a collaboration between elementary and secondary schools carrying out periodic revisions to adapt current demands. However, in higher education, there is no synergy between secondary education and standard of curriculum. It is left to respective universities to decide how IT education is to be implemented. Further, because there is no need for these subjects in college entrance examinations, there are textbooks that conform to the curriculum guidelines, but the students do not necessarily learn all of them, which creates a huge gap in skills when entering university.

The change in high school curriculum in 2002 mandating information education subjects to be part of curriculum expects graduates of such group to possess the needed information technology skills when they enter university, however most universities observed through the entrants of 2006 that the necessity change in high school was not reflected, which was the same situation as in 2002. It was also observed that the lesson plan and learning style are not limited to information technology education. There are many lecture-type lessons wherein the content are tied to the text found on the textbooks and are repeating the explanation and examples. It is not clear whether "living ability" (qualities/abilities as a learning base) mentioned in the curriculum guideline is being emphasized. Furthermore, it is not certain if it can be applied to the real world after graduation.

Adapting to the recent trends in course delivery, active learning and project based learning at higher education have been drawing attention in recent years and this has also spread to primary and secondary education. In science universities, there are many examples of initiatives in these area, but not in Ferris University, classified as a liberal arts university. It is difficult to implement these subjects as it will take so much time for all students to take these courses.

## 2.2. Programming Education

Nowadays, programming education is advancing not only in Japan, but also in other countries around the world. In the Japanese university education, the MEXT survey conducted in December 2013 reveals 67.1% of the universities are not carrying out programming education. Likewise, there are only few case studies covering programming education at undergraduate universities.

In elementary and secondary education, the curriculum was developed to be implemented at the junior and senior high schools based on the current curriculum guideline. But the results of the survey on freshmen at Ferris University in April 2017 revealed that their past learning experience in programming education is only 13% which is difficult to conclude that students have studied the programming education. ( Figure 1)

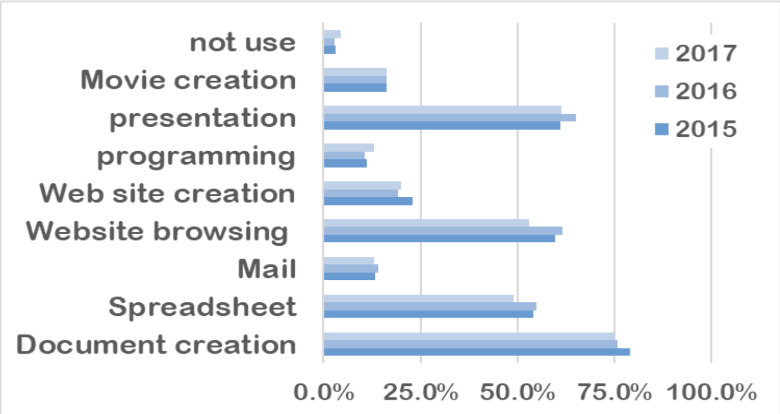



**Figure 1 What students have learned before entering university**

In April 2017, MEXT announced about the implementation of programming education at elementary school level in the next course of study. Guidance and investigation are underway for full implementation in primary education in 2020. Challenged by the fact of introducing programming concepts at this early age, the curriculum is designed to introduce the concepts in a manner easy to understand and fitted to the demands of the time. In elementary school, instead of learning the code, the objective should be beyond the era that fosters the programming thinking, a universal skill requirement of every human being. Elementary schools are to implement programming education in the curriculum under each subject without establishing any particular subject. This means programming education will be taught during the lectures in Math, Science, national language, and so on, in an integrated style. By this approach, it is meaningful to realize programming education regardless of "bun-kei" or "ri-kei" in higher education.

In the recent years, due to the demand of higher education graduates with IT skills needed to apply for work, a number of "bun-kei" colleges and universities have tried to introduce programming to their students. Ferris University, according to its study has 5% of its graduates who find work in IT related companies and this figure is expected to grow year by year. However, in most of these "bun-kei" colleges and universities, delivery of programming classes are of somewhat "experiential" approach, wherein students experienced a few lines (at most 20 lines) and often just copy the sample text as given. However, due to the change in Japan's IT Education

System, such "experience-based" delivery of programming lessons will become unnecessary because those experiences are already planned in elementary school curriculum in Japan's next course guidelines; and thus there is a need to a more useful and relevant "college-level" contents of programming classes.

As shown in the Figure 2, there is a comparative data of education that industry should give importance to educational institutions and education that educational institutions value.

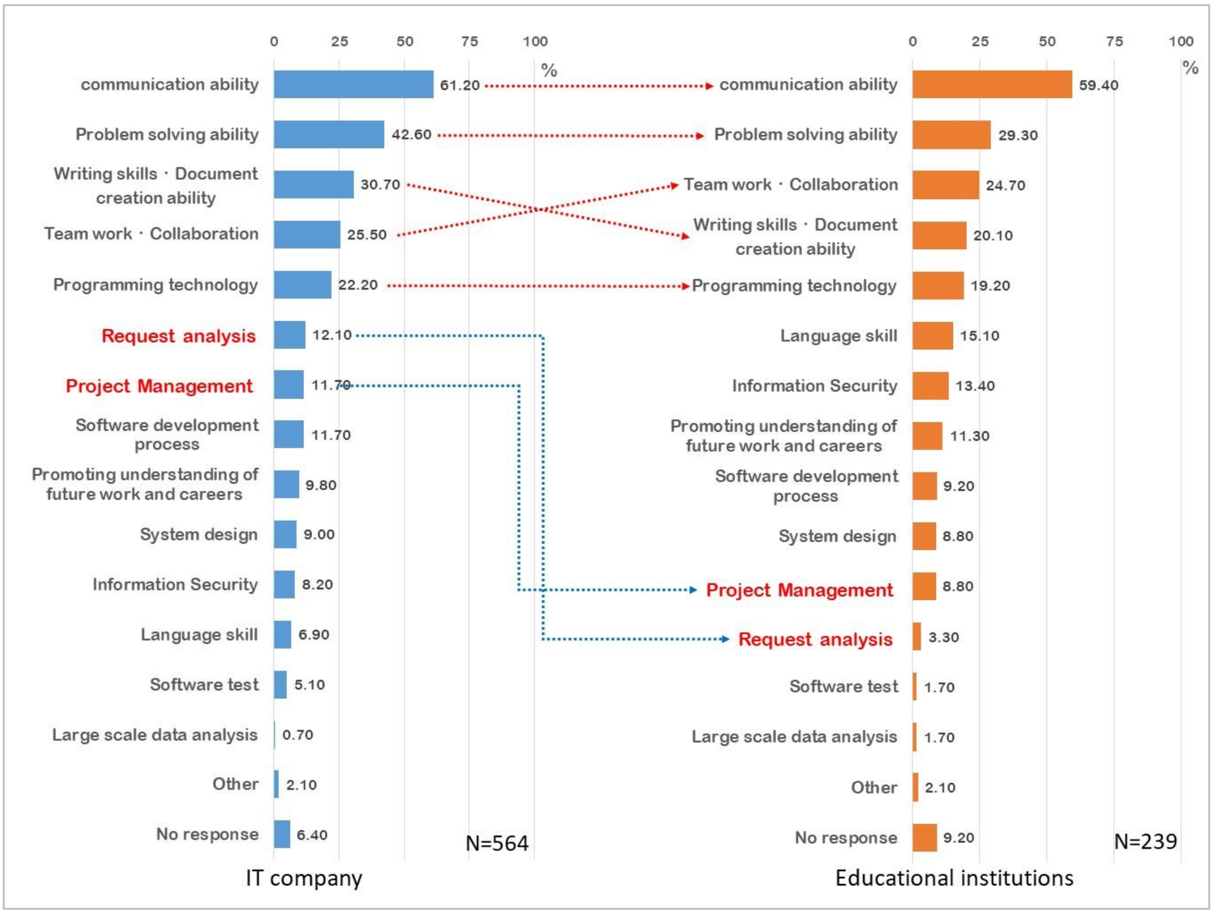



**Figure 2** Comparison of education that IT companies want to emphasize educational institutions and educational institutions emphasize in recent years[2]

From this figure, requirements analysis and project management are not considered important in educational institutions despite the items required in real society. In promoting programming education, paying attention to these items is important to consider the useful curriculum in the real world.

## 3. Design of Project-Based Learning in Programming Courses

### 3.1. Design of The Course

[2] Information-technology Promotion Agency, Japan（IPA）(2013). Practical course construction guide ~ Advancing Independent Development of Industry-University Collaborative Education ~ Part 3 Evaluation Criteria [in Japanese]. P.3-6
https://jinzaiipedia.ipa.go.jp/wp-content/uploads/2013/11/3_hyouka_m20131028.pdf (accessed 2017-9-1)

As mentioned in the previous sections, although programming education is expanding globally, efforts in higher education have many elements taken as part of information education at the "bun-kei" university. However, many of them are efforts in the science course and there is room for further study.

Also, even though it is being done, there is no way to think about themselves or to work on groups, such as skeleton programming. However, it is meaningful to provide a more practical learning environment for students coming to society from now on.

Also, from the data presented earlier, project management, problem solving ability, required analytical skill and others required in real society cannot be learned by studying once; more practical work and repeated experiences is effective. At the "bun-kei" university like Ferris University, this kind of environment is not very meaningful.

Thus the authors designed the curriculum as 50% programming learning and 50% project experienced contents.

The goals are

(1) Programming: we don't have programming curriculum other than this course. Students are interested in new technology and some students want to study (Figure 3) and desires to become system engineer (SE).

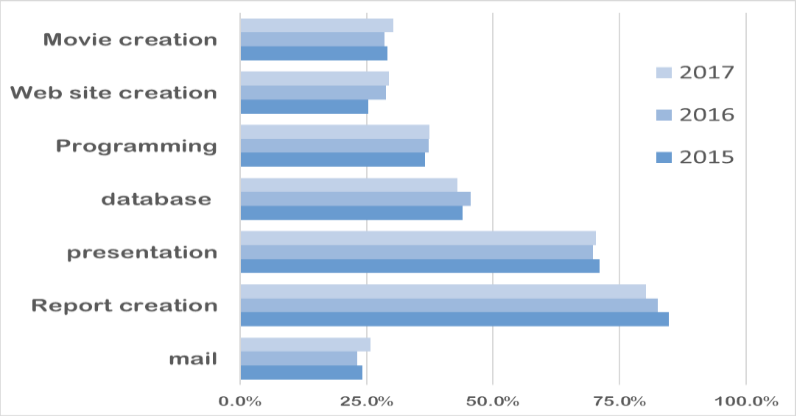



**Figure 3 What students would like to study at university**

(2) PBL: Students should experience group work, engage in communication, create outcome including report, presentation and drawing.

(3) IT skill: Students need to use computer for programing, sharing information, writing report.

Our course was designed with the objective that at the end of the semester, students are to produce a student project. Thus, the semester (15 weeks) was divided into first and second half. In the first half, students learn the concepts and syntax of a programming language and learn programming with emphasis on the knowledge required and useful for the second half. In the second half, students form several groups and engaged in a project. Although there are many "project-based" programming classes in many college and university, that are mainly targeted to 2nd through 4th-grade students (or even in graduate level) and who are already fluent with basic programming. Our approach is different because our targets are newbies, and this is first programming class for them.

The difficulty is that they are just beginners, yet they should program something meaningful as the output of group project.

Applying programming concepts, we have chosen "drawing figure" as the theme; students learn how to draw picture in the first half, and then create actual drawing (with animation) in the project part.

Project based learning approach in programming projects by default expects students to work on the project anytime they want, and not necessarily during scheduled class hours. However, due to the busy schedule, consultation on the progress of the project is only limited according the availability of the teachers. In our proposed class design, we are using the official class schedule for consultation, group discussion, and implementation.

Moreover, in ordinary class, what students receive from the class largely differ among students according to students' affinity to the class theme. Alternatively, our class design includes various activities such as programming, teamwork and report writing. Therefore, each student can contribute at strong point, and can receive their own outcome.

### 3.2. The Programming Part

The target students are “bun-kei” students learning programming for the first time. We have chosen Ruby as the programming language because it is easy to learn and the syntax is simple which is appropriate for beginners. The textbook adapted was made by one of the authors considering programming education.

The authors think that the followings should be avoided in programming education.

- Required to "just memorize" a lot of knowledge written in textbooks.
- Practice making exercises repeatedly with drills and others based on memorization. For example, "what kind of problem is there" "how to solve"
- When testing, you will be asked to solve many problems in a short time with "remembered" way.

Instead, the authors think the following points should be pursued in effective programming education.

- Students should understand "minimum rules for programming" and make those rules repeatedly in various way. Thereby they should practice "how to think like a programmer."
- Students should choose problem they solve according to their skills and interests, which result in better motivation.
- Students should understand there are large variety in writing "correct" programs, and there are no "single correct solution."

Moreover, conventional textbook has the following undesirable features.

- Rules (writing, functions) of the programming language are explained one by one in the textbook, and they learn in order even in classes.
- It is a question of whether the exercise problem is not "writing a program" but also whether or not you learned the knowledge of the content being explained.
- There are only a few examples of the program, and we carefully explain a little of that example.

Alternatively, we have adapted text with following features.

- Minimum rule of thumb are explained, and students are encouraged to apply those rule through various exercises.
- Many number of exercises are provided with various difficulty; students choose their exercise to solve according to their levels and interests.
- Many topics and example code are provided, and students choose whichever topic to attract their interests.

Our textbook consists of following sections.

- Introduction
- Selection and Repetition
- Control Structure and Arrays
- Procedure and Abstraction
- 2-Dimensional Arrays and Images

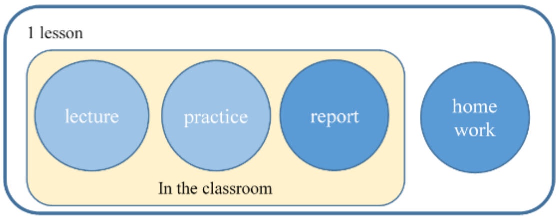



**Figure 4 Image of 1 lesson**

A class includes explanations, exercises, tasks at the time of class and homework. In our text, tasks and homework are devised so that no one tackles all problems, they can select and try according to their abilities. Through these processes, students will be able to draw pictures.

### 3.3. The Project Part

The purpose of this part is to make students acquire "Project Management ability," "ability to use IT," and " requirements analysis ability."

In this part, students make small group of 3-5 persons per group. They develop ruby program to make animation in collaboration.

Their tasks are the following:

- Make a group and decide the leader and each member's role
- Decide theme (main character) and structure of the animation
- Decide each member's part in creation
- Decide schedule and integration plan
- Make program one by one
- Integrate each part and correct defect

- Prepared and perform presentation

The skills such as communication, problem solving, writing, teamwork, programming, among others are necessary skills set for society. From the diagram shown in chapter 2, "Request analysis" and "Project management" can be read as necessary items after entering the society. Therefore, we designed the course focusing on these points:
Imagine developing a software. Software developers need programming skills. For collaboration with software engineers, it is a good idea for orders to have knowledge of programming. For both sides, knowledge of requirements analysis and knowledge of project management is necessary, which is important for "bun-kei" students.

## 4. Case Study

### 4.1. Class Overview

We experimented on two (2) classes in the first semester (April-July) of 2017. One class (class A) is composed of 8 students and the other class (class B) is composed of 5 students. For both classes, students enrolled are in grade level 2 - 4 with majors in English literature and International Cultural affairs. Normally, classes are done in Japanese, but in this case, we did the class in English.

We used LMS (Moodle) to distribute text and other materials and used the same to collect the impressions (a discussion of individuals understanding per lesson), requests of lessons, and homework of the students.

The project is distributed according to small groups. In class A, we made 2 groups (A1 and A2) and 1 group in class B (B1). The course was delivered in two different classroom layout. Class A used traditional lecture-room style layout (Figure 5, left picture), and class B used collaborative style room layout (Figure 5, right picture). Those room layout difference resulted in difference of students' experiences and outcomes, which will be explained later.


**Figure 5** Picture of Classroom A & B

Project task and goal setting was being discussed in the first lesson of group work and a schedule of what is expected during final presentation was presented. See Figure 6 below for sample. As a team, each members of the group expressed ideas on what character to create, discussed the nitty-gritty of the requirements, difficulty of the drawings, and accepts each assigned task/role.

## Schedule

Final presentation of the project in class

July 24, 2017

What to present

- The output or the drawing
- A discussion of the role of members in the group in creating the drawing
- Schedule, a discussion of the time spent in the development of the drawing project
- Review of the code, wherein you will discuss how you implemented the drawing
- Challenges in the development of the project and how it is being answered

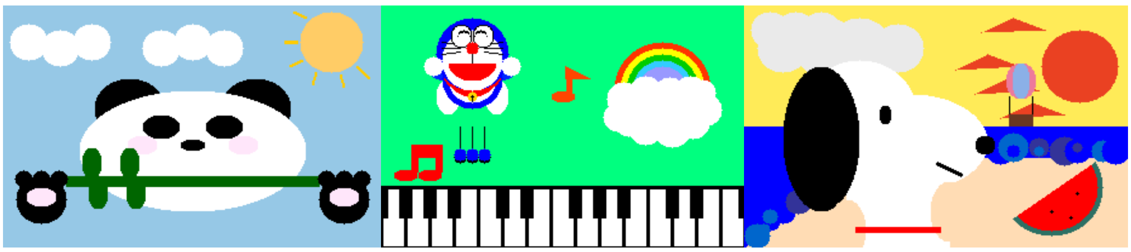
**Figure 6** sample of Schedule

Further, we explained the following important items

- project management
- Information utilization ability
- Request Analysis
- Software development
- Hearing
- Role sharing of team
- Policy formulation

### 4.2. Case Study Result

Applying what they learned in class, the following pictures (Figure 7) are the output of each group. These pictures are still images of the animated version as presented in class.



Group A-1 Group A-2 Group B-1

**Figure 7** Output of drawing

In the next section, we show how group B-1 go about their process to make picture. (Figure 8) These information are coming from the data as presented during group presentation in class.

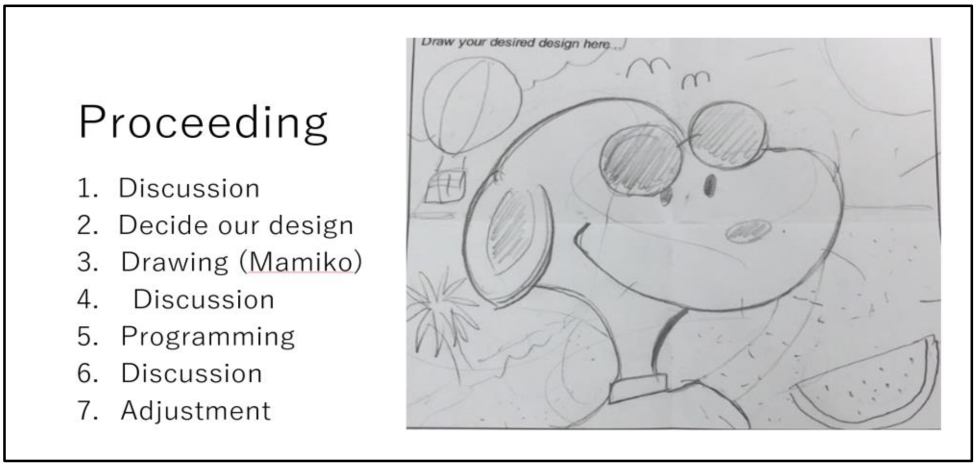



**Figure 8** presentation slid of Group B

First they thought of a character to create. Once decided, they drafted the image and decides on the background to integrate. Using a matrix, they plotted the lines to make up the expected character. The figures (Figure 9) below are notes of actual work.

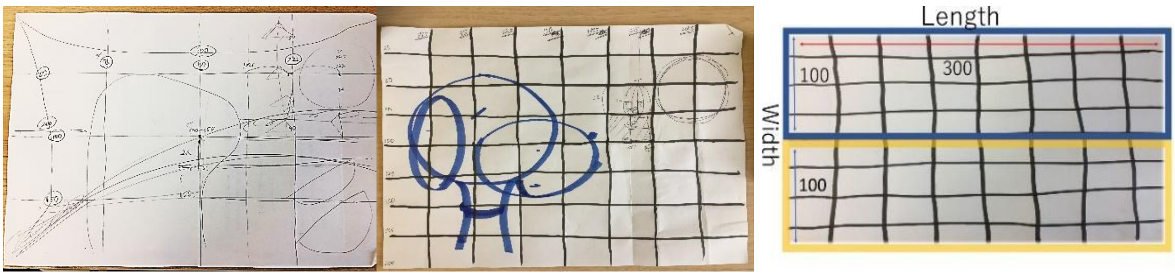



**Figure 9** Sample of Draft

Through these processes, students programed 5 to 10 pictures in each group and finished them into animation.

The situation of each group is shown below.
A-1:
There are 3 students in this group. At first, they draw base picture (face of panda) one by one. Then they chose good one with competition. Because face of panda is simple. This method has merit in choosing pretty picture. However, this method is not good for more complex picture. They understood those merit and demerit. After deciding base picture, they assigned different role to each of the members.
Role of parts:

- cloud・sun・bamboo / Make animation of the sun
- Make animation of Panda’s eyes
- Panda’s hands / Change the color of Panda’s cheeks

Role assignment resulted in more complex and integrated picture. They had difficulty in combining code written separately. They have learned effective use of comment in the code (to indicate which part was newly written) to solve the problem.

A-2:
There are 4 students in this group. This group discussed members of role assignment first.

- Doraemon
- Keyboard

- Rainbow and cloud
- Note and Symbol

There was skill difference among members. So skilled member was assigned to the complex part. Then, they worked mostly independent.

This group's merit is small communication overhead. On the other hand, the integration was done mostly by one person, whose burden was large. Other member sent their part to the integrator with e-mail. Additionally, they had difficulty in preparing presentation, because their co-operation was weak.

B-1:
There are 5 students in this group. This group also discussed members of role assignment first. They divided 3 parts of picture.

- The background: Sky
- The background: Beach
- Main character

Unique point is that 5 members divide 3 parts, and main character made by 3 students.

This group had more communication than other groups. Therefore, throw the project progressed, drawing drafts like the figure, document management and preparation of presentation was well done. On the other hand, there was failure in the early stage of the project. They saved only the output of drawing and forgot to save the code. They re-created the program from the start. However, they got to learn and had a positive impact on the latter work.

**5. Discussion**
For the projects, all three groups succeeded in completing their project on time. Those projects were also manifestations that the students understood the goals as set by the course. As a team, role assignments allowed them to test their newly acquired programming skills, and at the same time experience firsthand project management. Challenges in management and cooperation among group members were also observed during the course. However, in due time the issuers were cooperatively resolved, which helped them understand issue management --- necessary skills in society. Likewise, all issues, mistakes, and challenges were accepted positively by the students, and were treated as valuable experiences.

The followings are the observations with respect to the goals set for non-science students:
(1) Programming: Interests on learning programming was observed in the class. Students did their homework and tried the examples as given on the texts. Additionally, some students also tried solving problems in the text. An answer to the questionnaire at the end of the semester are given as follows:

Student A:
I was able to create and move the instructions to run the program using Ruby. I learned and understood the code such as how to set conditions and how to iterate. However, I do not yet have the ability to fully apply my knowledge.

Student B:
I think that I can write basic Ruby program, although I had too much reliance on samples.

(2) Project Based Learning: Initially the project was a big challenge for most of the students. However, they were able to adapt to this type of learning, and were doing well up to the end of the semester. We have observed that room layout was also an important factor on this type of learning. Collaborative style classroom layout was more suitable for students engage in such learning approach, as it was easier for them to discuss and exchange ideas. Unlike the traditional classroom layout, collaborative style layout will also encourage students to be more relaxed, and thus in a way opened better communication channel.

(3) ICT skill: Students' knowledge of social media applications, cloud technology and office productivity tools contributed to the success of their project. Being used as a measure to connect to other member, those tools helped them proceed with their project even outside of their official class hours. Below are some students' excerpts on this area:

Student A:
Information sharing by Google Drive and LINE was carried out, which allowed me to grasp progress of other members smoothly even when the schedule was tight.

Student B:
All data sharing and storage were done in the cloud. When we worked outside class hours, I was able to inform the progress situation on the cloud, and I could do the insertion of additional code smoothly. Although It was my first time to be engaged in group work using ICT equipment, I was able to see information stored on class portals and clouds at any time using smartphones and PCs; it was useful for preparation and review.

The following are other comments of students;

Student A:
I did not understand English technical terms which I have not encountered in other lessons. I suffered at the beginning, but I gradually began to understand and I was able to carry out the tasks to the end. Therefore, I am filled with feeling of accomplishment now.

Student B:
When I was doing the assignment, I could copy the teacher's code and run the program, I could apply my report tasks using those teacher's code. However, I could not code with my own way of thinking.

In this class, there were a lot of problem-solving learning, and when I did not understand, I could ask the teacher for advice.
However, I feel that it was necessary to make efforts to think on my own a little more.

By the evaluation of the students' comments at the end of the semester, our aim was largely achieved. From almost no knowledge of programming, our students had learned how to write programs and how those programs behave. They understood the cycles in program development, and it helped them better appreciate applications they are using.

In general, the students:
a) learned programming language syntax and program structure which allowed them to code on their own,
b) appreciate schedule management and understood the importance of collaboration, and

c) fully recognized the benefits of information sharing, utilization of cloud and social media applications.

Perhaps these knowledge at present are inadequate for them to challenge larger programs or systems, but they will be enough to allow them to move forward in systems development.

Additionally, we saw that classroom design had important effect in group-type project development. However, since our study is merely a single case, we need more experience as to how collaborative classroom layout is really effective or not. However, we could observe advantages of the collaborative room layout based on group performances assigned to that room.

**6. Final Remarks**

The 21st century market demands technology savvy workforce addressing the needs of the growing society. Graduates are no longer clearly divided into IT or non-IT, but rather graduates are expected to possess skills in ICT, a common requirement for graduates to enter the workforce. With the fact in mind, it becomes necessary for "bun-kei" students to learn information technology reflecting the needs of society. The challenge however is the fact that these type of students are not mentally programmed to learn programming, thus attention to the content, design, and delivery of the course should be noted. Therefore, we designed and implemented a 50% programming and 50% project-basic programming course to "bun-kei" students. Students were able to actually write programs and learned a lot from the project, appreciate project management, and share data and information.

**References**

HAGIYA, Masami(2015). Invited Papers Defining Informatics across Bun-kei and Ri-kei. Journal of Information Processing Vol. 23 (2015) No. 4 pp. 525-530, July 15, 2015 https://www.jstage.jst.go.jp/article/ipsjjip/23/4/23_525/_article (accessed 2017-9-1)

Information-technology Promotion Agency, Japan（IPA）(2013). Practical course construction guide ~ Advancing Independent Development of Industry-University Collaborative Education ~ Part 3 Evaluation Criteria [in Japanese]. https://jinzaiipedia.ipa.go.jp/wp-content/uploads/2013/11/3_hyouka_m20131028.pdf (accessed 2017-9-1)

KAKESHITA, Tetsuro and TAKAHASHI. Naoko (2017). National Survey of Japanese Universities on IT Education：National Survey of Japanese Universities on IT Education : Overview of the Entire Project and Preliminary Analysis [in Japanese]. Journal of Information Processing Vol. 58 (2017) No. 5 pp. 420-425, April 15, 2017

KAWAMURA, Kazuki (2016). Future Informatics Education in university [in Japanese]. Nikkei BP Marketing, 2016.3, http://macrobrain.sakura.ne.jp/mbopen/201604IPS.zip (accessed 2017-9-1)

KUNO, Yasushi (2016). What purpose do you learn programming?And how? [in Japanese]. Programming Symposium 2016 in Summer, Education and Learning, 2016 http://lecture.ecc.u-tokyo.ac.jp/~ckuno/is16/sprosym-2016.pdf (accessed 2017-9-1)

Ministry of Education, Culture, Sports, Science and Technology. About Way of Programming Education in Elementary School Stage (summary of the discussion) [in Japanese]. http://www.mext.go.jp/b_menu/shingi/chousa/shotou/122/attach/1372525.htm (accessed 2017-9-1)

Ministry of Education, Culture, Sports, Science and Technology (2016). Central Council for Education curriculum planning Special Committee 2016.8.1 [in Japanese]. http://www.mext.go.jp/b_menu/shingi/chukyo/chukyo3/053/siryo/1375316.htm (accessed 2017-9-1)

Ministry of Education, Culture, Sports, Science and Technology (2016). Curriculum Committee (98th) Handouts [in Japanese]. http://www.mext.go.jp/b_menu/shingi/chukyo/chukyo3/004/siryo/1376580.htm (accessed 2017-9-1)

Ministry of Education, Culture, Sports, Science and Technology. Education English version. http://www.mext.go.jp/english/introduction/1303952.htm

Nuutila E., Törmä S., Kinnunen P. & Malmi L. (2008). Learning Programming with the PBL Method — Experiences on PBL Cases and Tutoring. In: Bennedsen J., Caspersen M.E., Kölling M. (eds) Reflections on the Teaching of Programming. Lecture Notes in Computer Science, vol 4821. Springer, Berlin, Heidelberg, pp 47-67

Nuutila E., Törmä S. & Malmi L. (2005). PBL and Computer Programming — The Seven Steps Method with Adaptations. Computer Science Education, v15 n2 p123-142

TAKAHASHI, Naoko (2017). National Survey of Japanese Universities on IT Education : Preliminary Analysis of General IT Education [in Japanese]. Journal of Information Processing Vol. 58 (2017) No. 6 pp. 526-530, May 15, 2017

Science Council of Japan (2016). Reference standard on educational curriculum organization for quality assurance in higher education informatics field [in Japanese]. http://www.scj.go.jp/ja/info/kohyo/pdf/kohyo-23-h160323-2.pdf (accessed 2017-9-1)