Teaching management to resolve student problems in *Computer Programming 1*

Vipa Jakchaikul† & Thawatchai Kansrirat‡

Rajamangala University of Technology Phra Nakhon, Bangkok, Thailand†
Mahidol University, Phayathai Campus, Bangkok, Thailand‡

ABSTRACT: This study aimed to reduce the number of students that failed in the examination of *Computer Programming 1*, which is a part of the Bachelor of Science Programme in Computer Science at Rajamangala University of Technology Phra Nakhon, Bangkok, Thailand. This subject concentrates on practical teaching. The authors oversaw teaching that subject and collected the academic results of 192 students in the academic years 2013 and 2014. With a traditional method of teaching, it was found that 15.11% of students in the academic year 2013 failed compared with 30.18% of students in 2014. The authors, then, improved teaching management of Computer Programming 1 for the academic year 2015. This improved teaching management used *best practice* to reduce the number of students with poor results or failing in the examination of the subject in future academic years, as well as another subject concentrating on the practical teaching.

INTRODUCTION

Computing is …*introduced as a new subject in a school curriculum in many countries, and as an important part of informal learning opportunities in others* [1]. However, computer programming education is …*one of the greatest challenges that have remained for years in Computer Science Education* [2], as computer programming …*is a difficult course to teach and learn* [3]. Many students found that programming is difficult and disheartening. Programming is the basic skill required of computer programmers [4].

The experience in teaching computer programming has proved to be a challenge [4]. Therefore, it is recommended that more than one teaching strategy be adopted for teaching programming, and adaptation of peer tutoring, pair/group programming and problem-solving, to give …*learners better opportunities to interact with their peers and teachers* [5].

From the challenge of computer programming, many students find that it is difficult to write a program, so many of them received a poor result (*D+* and *D* grades) or failed the examination (*F* grade) of Computer Programming 1 in previous semesters. As a subject, it is provided as a part of Bachelor of Science Programme at Rajamangala University of Technology Phra Nakhon, Bangkok, Thailand. It is both a fundamental course for computer programming and a prerequisite course prior to taking another course. One of the authors is an instructor in that subject, and is interested in solving-problems for students who receive a poor or fail examination result in Computer Programming 1.

TEACHING PLAN OF COMPUTER PROGRAMMING 1

The teaching plan of Computer Programming 1 was used for computer science undergraduate students at Rajamangala University of Technology. This subject lasted for one semester and spanned 16 weeks, as shown in Table 1.

Table 1: Teaching plan of Computer Programming 1 for computer science's undergraduate programme.

Week	Topic/detail (credit/chapter/topic)	Hours	Teaching activity/media used
1	Overview of course, evaluation, studying materials Introduction to computer	4	Lecture and use media
2	Introduction to C programming Practice of C programming	2 2	Lecture and use media Workshop
3	Structured program development Practice of structured program development	2 2	Lecture and use media Workshop
4	Program control Practice of programming program controls	2 2	Lecture and use media Workshop

Week	Topic/detail (credit/chapter/topic)	Hours	Teaching activity/media used
5	Function Practice of function programming	2 2	Lecture and use media Workshop
6	Array Practice of array programming	2 2	Lecture and use media Workshop
7	Pointers Practice of pointers programming	2 2	Lecture and use media Workshop
8	Test	4	-
9	Formatted input and output Practice of programming with formatted input and output	2 2	Lecture and use media Workshop
10	Structures, unions, bit Practice of programming with structures, unions, bit	2 2	Lecture and use media Workshop
11	File processing Practice of file processing	2 2	Lecture and use media Workshop
12	Data structures Practice of programming with data structures	2 2	Lecture and use media Workshop
13	Introduction class and objects Practice of class and objects programming	2 2	Lecture and use media Workshop
14	Introduction to object-oriented programming Practice of object-oriented programming	2 2	Lecture and use media Workshop
15	Practical test	4	-
16	Final examination	4	-

DATA COLLECTON

Data were collected from 192 undergraduate students for two semesters in 2013 and 2014, as shown in Table 2.

Table 2: Study results of Computer Programming 1 in the academic year 2013-2014.

Academic year 2013			Academic year 2014		
Grades	No. of students (86)	%	Grades	No. of students (106)	%
A	4	4.65	A	6	5.66
B+	-	-	B+	6	5.66
B	7	8.13	B	12	11.32
C+	8	9.30	C+	11	10.37
C	11	12.80	C	17	16.03
D+	22	25.58	D+	12	11.32
D	21	24.42	D	10	9.43
F	13	15.11	F	32	30.18

Results shown in Table 2 revealed that 22 + 21 = 43 students, which were 25.58% + 24.42% = 50% of the total number of students in the academic year 2013, received a poor study result (*D*+ and *D* grades), while 13 students, which were 15.11% of the total number of students, failed (*F* grade).

For the academic year 2014, 12 + 10 = 22 students, which were 11.32% + 9.43% = 20.75% of the total number of students, received a poor study result (*D*+ and *D* grades), while 32 students, which were 30.18% of the total number of students, failed in an examination (*F* grade).

After that, the study results from two semesters were used to forecast a trend in the 2015 academic year by using simple moving average (SMA): $F_{t+1} = (Y_t + Y_{t-1} + Y_{t-2} + \ldots + Y_{t-k}+1)/k$. The results are shown in Table 3.

Table 3: Forecast results for the academic year 2015.

Academic year 2015*		
Grades	No. of students (96)	%
D+	12	12.50
D	11	11.45
F	23	23.95

* Estimated for the academic year 2015 in the forecast

When using study results of two academic years to forecast the number of students and study results for the academic year 2015, it was expected that there would be 96 students, and 23 of them (23.95% of the total number of estimated

students) would receive *D*+ or *D* grade. For estimation of the number of students who would fail in the examination, 12 and 11 students (12.50% and 11.45% = 23.95% of the total number of estimated students) were expected. From the forecast, the issue had to be resolved as soon as possible.

ANALYSIS OF STUDENT PROBLEMS

Interviews were used, by purposively sampling 110 students in the academic years 2013 and 2014 who had received *D*+ and *D* grade (total 65 persons) or *F* grade (total 45 persons). It was found from the interviews that students felt that the subject was difficult because of programming course characteristics, as shown in Figure 1.

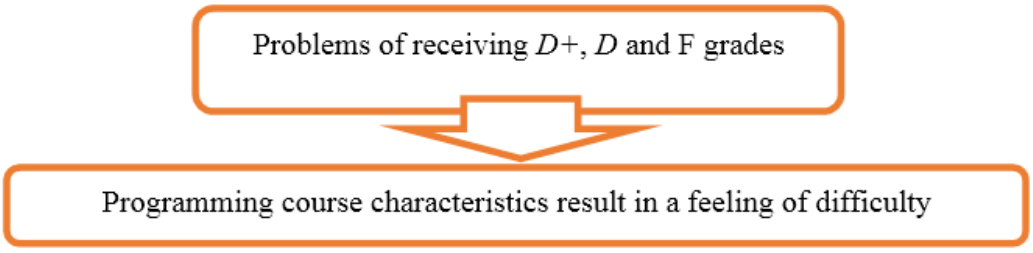

Figure 1: Summary of problems by students undertaking Computer Programming 1.

SOLUTIONS

From the issue that programming course characteristics result in a feeling of difficulties, the authors designed supplemental teaching techniques to support the existing teaching plan as follows: 1) developing laboratory sheets; 2) increasing student participation; 3) providing formative tests; and 4) pairing two classmates to work on a project:

1. *Laboratory sheet development*: in each practical class, a laboratory sheet was distributed, comprising questions sorted from easy to difficult, as well as programming guidelines for students to achieve the target.
2. *Student participation*: after developing a small module of programs, a student presents to classmates for knowledge exchange, then, an instructor summarises the session.
3. *Formative test*: after performing a formative test of each chapter, a student obtains a test result, and those not passing the test are required to solve the issue, either by taking an additional test or being taught by an instructor again.
4. *Two-person project*: an instructor informs students that once the class has finished, they will have to build a software project in which knowledge learnt will be adopted for programing. The students then present their project to classmates so the latter could understand the project.

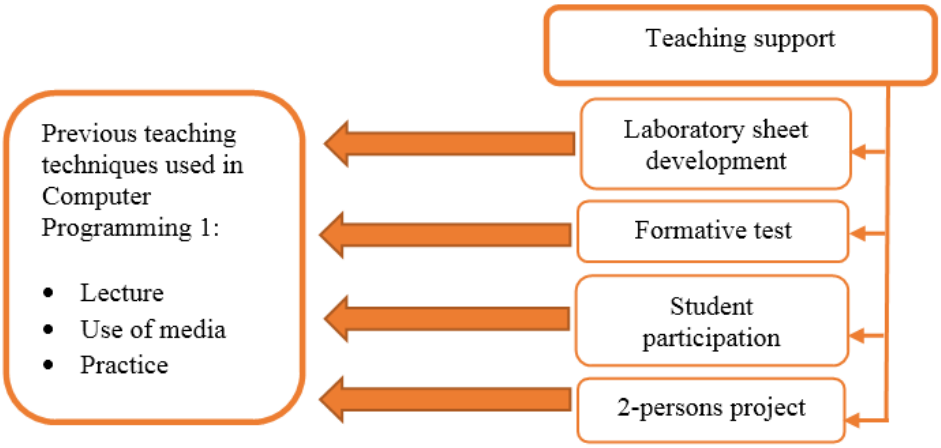

Figure 2: Improvements of teaching management for Computer Programming 1.

CONCLUSIONS

After improving the teaching plan by developing laboratory sheets, encouraging student participation, doing formative tests and setting up two-person project, out of 92 students in the academic year 2015, only one person (0.92% of the total number of students) received a poor study result (*D* grade) and nine persons (4.6% of the total number of students) failed the examination (*F* grade). When compared with the academic year 2013-2014, the number of students that obtained a *D*+, *D* or *F* decreased, as shown in Table 4.

Table 4: Study results of students during the academic years 2013, 2014 and 2015.

Academic year 2013			Academic year 2014			Academic year 2015		
Grades	No. of students (86)	%	Grades	No. of students (106)	%	Grades	No. of students (92)	%
A	4	4.65	A	6	5.66	A	4	4.34
B+	-	-	B+	6	5.66	B+	12	13.04
B	7	8.13	B	12	11.32	B	19	20.65
C+	8	9.30	C+	11	10.37	C+	25	27.17
C	11	12.80	C	17	16.03	C	26	28.26
D+	22	25.58	D+	12	11.32	D+	-	-
D	21	24.42	D	10	9.43	D	1	1.08
F	13	15.11	F	32	30.18	F	5	5.43

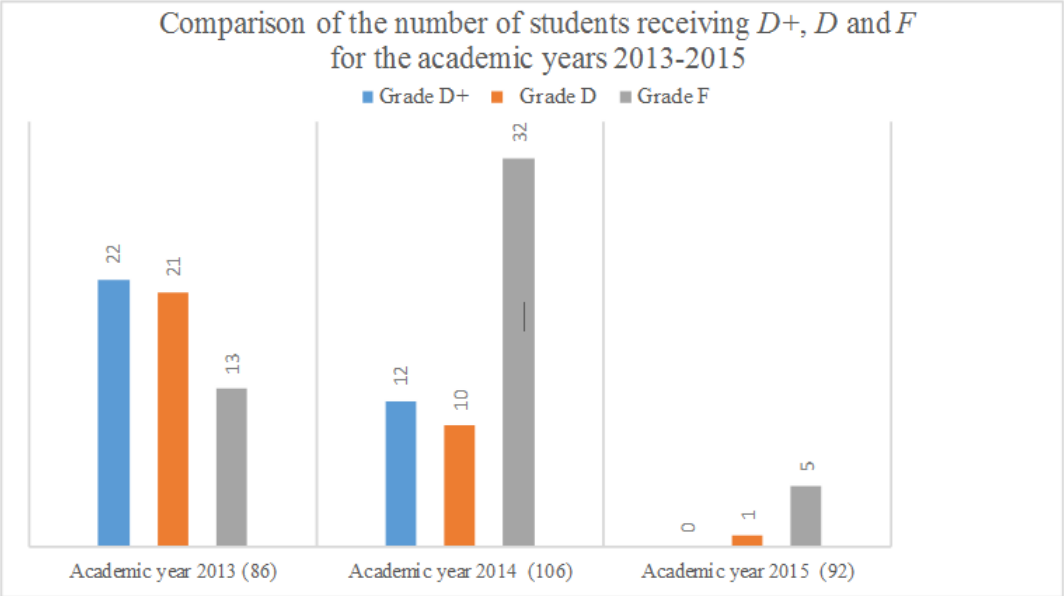

Figure 3: Comparison of study results between academic years 2013, 2014 and 2015.

It can be seen from Figure 3 that the 2015 academic year had an acceptably low number of students with poor grades (*D+* and *D*) or those that failed in an examination (*F* grade) when one of the authors was the instructor of the course. This study set a target that the number of students failing in an examination (*F* grade) in the next semester would be further reduced.

SUGGESTIONS

The following suggestions can be made:

1. Prior to a later theoretical or practical class, the instructor should assign a task so that students can prepare themselves before entering the class.
2. The instructor should give assignments for each chapter after theoretical study. Then, he/she should review the answers to those assignments and explain a solution to an incorrect answer, so that students will gain more understanding.
3. The instructor should give a formative test in the class prior to the mid-term and final examination.
4. The instructor should answer and inform about a score every time after hosting either a theoretical or practical formative test, so that students will be aware of their learning progress.

ACKNOWLEDGEMENTS

This work was supported by Rajamangala University of Technology, Phra Nakhon, Thailand.

REFERENCES

1. Sentence, S. and Csizmadia, A., Computing in the curriculum: challenges and strategies from a teacher's perspective (2016), 10 May 2016, *Educ. and Infor. Technologies*, http://link.springer.com/article/10.1007/s10639-016-9482-0

2. Bassey, I., A methodology for teaching computer programming: first year students' perspective (2014), 10 May 2016, *Inter. J. of Modern Educ. and Computer Science*, 9, 15-21, http://www.mecs-press.org/ijmecs/ijmecs-v6-n9/IJMECS-V6-N9-3.pdf
3. Oroma, J., Wanga, H. and Ngumbuke, F., Challenges of teaching and learning computer programming in a developing country: lessons from Tanzania. *INTED 2012 Conf.*, Valencia, Spain, 3820-3826 (2012).
4. Ismail, M.N., Ngah, N.A. and Umar, I.N., Instructional strategy in the teaching of computer programming: a need assessment analyses (2010), 20 May 2016, *The Turkish Online J. of Educational Technol.*, 9, **2**, 125-131 (2010), http://www.tojet.net/articles/v9i2/9214.pdf
5. Sarpong, K.A-M., Arthur, J.K. and Amoako, P.Y.O., Causes of failure of students in computer programming courses: the teacher - learner perspective (2013), 30 May 2016, *Inter. J. of Computer Applications*, 17, **12**, 27-32 (2013), http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.403.2714&rep=rep1&type=pdf