International Journal of Internet, Broadcasting and Communication Vol.12 No.1 104-110 (2020)
http://dx.doi.org/10.7236/IJIBC.2020.12.1.104

IJIBC 20-1-14

A Study on the Effect of Pair Check Cooperative Learning in Operating System Class

Woochang Shin

Professor, Dept. of Computer Science, Seokyeong University, Korea
wcshin@skuniv.ac.kr

Abstract

In the 4th Industrial Revolution, the competitiveness of the software industry is important, and as a solution to fundamentally secure the competitiveness of the software industry, education classes should be provided to educate high quality software personnel in educational institutions. Despite this social situation, software-related classes in universities are largely composed of competitive or individual learning structures. Cooperative learning is a learning model that can complement the problems of competitive and individual learning. Cooperative learning is more effective in improving academic achievement than individual or competitive learning. In addition, most learners have the advantage of having a more desirable self-image by having a successful experience. In this paper, we apply a pair check model, which is a type of cooperative learning, in operating system classes. In addition, the class procedure and instruction plan are designed to apply the pair check model. We analyze the test results to analyze the performance of the cooperative learning model.

Keywords: *Pair check model, Cooperative learning, OS instruction plan, T-test verification*

1. Introduction

Software is at the core of the Fourth Industrial Revolution, represented by the Internet of Things, Cloud, Big Data, and Artificial Intelligence, but Korea's software industry is not competitive. It has been argued that educational institutions should provide training to produce high quality software personnel as a solution that can fundamentally secure the competitiveness of the software industry [1, 2]. In particular, in order to innovate university education to meet the needs of the software industry, the Ministry of Science and ICT has been investing heavily in software-oriented universities since 2015.

Despite this situation, software-related classes at universities are largely composed of competitive or individual learning structures. Competitive learning is a learning structure in which teachers present certain learning goals and encourage competition among groups or individuals so that only a small number of groups can reach them. This resulted in a few winners and a large number of losers, leading to a waste of human resources and a minority of winners. Competitive learning structures have earned the stigma of being the worst invention in the 20th century [3]. Individual learning models are also criticized in terms of whether

Manuscript Received: December. 21, 2019 / Revised: December. 29, 2019 / Accepted: January. 2, 2020
Corresponding Author: wcshin@skuniv.ac.kr
Tel:+82-2-940-7755, Fax: +82-2-940-7755
Professor, Dept. of Computer Science, Seokyeong University, Korea

the individualized teaching system can be applied in multi-person classrooms, and in the negative aspects due to the restriction of student interaction [4].

Cooperative learning is a learning model that can complement the problems of competitive and individual learning. Cooperative learning is a method of teaching students with different learning skills working together in small groups towards the same learning objectives [5]. In general, cooperative learning is more effective in improving academic achievement than individual or competitive learning. In addition, most learners have the advantage of having a more desirable self-image through success experiences [4].

In this study, we apply a pair check model, which is a type of cooperative learning, in operating system classes that software developers should learn. We design the teaching procedures and instruction for applying the pair check model. We also analyze the effects of cooperative learning models through class assessment.

2. Related Work

In Korea, there is a case study of applying the pair check cooperative learning model to English or mathematics education in elementary school. Research by Lim and Kim examined the effect of the pair check learning model on mathematics achievement and mathematical attitude [6, 7]. Research by Park examined the effect of the student's experience of solving problems in the curriculum with a partner and creating similar problems on the students' mathematical thinking and communication skills [8].

Overseas, there have been case studies of applying cooperative learning in the undergraduate computer science curriculum [9-11]. Research by E. F. Gehringer shows a variety of active and collaborative learning strategies [9]. Research by J. C. Prey described the integration of collaborative learning environments into the computing learning curriculum [10]. There was also a study on the cooperative learning effect using the mobile terminal [11].

In addition, there are studies related to mobile learning that provide an educational environment based on student interaction and dialogue to support cooperative learning [12, 13].

3. Cooperative Learning in Operating System Class

3.1 Operating System Class

An operating system (OS) is software that controls hardware, manages computer resources, eases the use of the computer, facilitates the execution of applications, and acts as an intermediary between the user and the hardware. Operating system courses that study concepts and algorithms related to operating systems are fundamental and important courses in software majors. In general, operating system subjects are composed of theory-based lectures, which makes it difficult for students to understand the contents of the lessons and to apply algorithms.

In this paper, we analyzed the effect of cooperative learning on students' learning outcomes by applying a pair check model, which is a type of cooperative learning in the OS subject.

3.2 Pair Check Cooperative Model

The pair check model is a structure-based collaborative learning model proposed by Kagan in 1994 and is often used to practice skills and solve structured worksheets [3]. This model is based on the simple common sense that complex and difficult learning materials are much easier to understand using the thinking power of two people than one.

The partner enhances the cognitive achievement effect by alternately performing the roles of summarizer,

finisher, speaker and listener. Teachers should not be directly involved in activities, but rather motivate them with praise and encouragement to play a facilitating role.

3.3 Lesson Goals

Lesson goals direct the direction of the following educational activities and give students expectations [14].

The goal of the operating system class is set in the form of the goal of the cognitive domain divided into 'knowledge' / 'understanding' / 'application' due to the nature of the subject contents. Table 1 shows an example of lesson goals for each category.

Table 1. Example of lesson goals in OS class

Category	Lesson Goals
Knowledge	- Know the basic concept of virtual memory. - Know the 4 conditions that cause the deadlock.
understanding	- Understand how to translate a logical memory address to a physical address. - Understand the cause of internal / external fragmentation.
Application	- You can draw a Gantt chart and calculate the average waiting time by applying the CPU scheduling algorithm in a given environment.

When you start an operating system class, you first introduce the lesson goals, briefly explain the schedule, and then proceed with the class.

3.4 Class procedure

The class are divided into "introduction", "development", "pair check activities" and "evaluation and arrangement". The activities of each stage are as follows.

(1) Introduction: Introduce the lesson goals of the class and briefly explain the learning schedule. It also explains the background of the learning topic and related concepts.

(2) Development: Conduct a detailed theoretical lecture on the contents of the study. Give students examples of how to solve problems by applying theory.

(3) Pair Check Activities: Cooperative learning of students according to the pair check model. The roles of professor and student during pair check activities are as follows:

<Professor's Role>

- Matching pairs: Pairs are paired at the beginning of the semester, instructing pair members to sit next to each other.
- Matching problems with paired groups: Present problems that can be applied to the content described in the theoretical lecture (algorithm, etc.).
- Assigning roles in pairs (organizer/ assistant): First, the organizer actively suggests the opinions, and the assistant discusses them in the form of accepting or refusing them. Alternate roles according to class time.
- Observation of the discussion process: A professor do not participate directly in discussions unless paired groups request mediation.

<Student's Role>

- Understand the problem presented: Ask the professor if further explanation of the problem is needed

- Conduct discussions in pairs: the organizer actively suggests opinions first, and the assistant conducts the discussion in the form of refusing or accepting them.
- Check the problem solving method and answer: After the discussion, solve the problem separately, and compare the problem solving process and results with each other.
- Determine the pair's answers after the final discussion.
- Write the pair's final solution and answer in their quiz notes.
- Present the pair's answers

(4) Evaluation and Arrangement: Briefly summarize and organize the lessons. Evaluate pair check activities by having each paired group report their answer to the problem. Finally, notice the next lesson.

The professor encourage active discussions with external pairs, as well as within pairs, during pair check activities. Sometimes students are not familiar with the way in which they discuss each other in their major classes, so active discussions may not be possible at first. It is necessary to encourage active discussions with each other.

3.5 Instruction Plan

Table 2 shows an example of an instruction plan that uses a pair check model in operating system classes.

Table 2. Example of instruction plan using pair check model in OS classes

Date	Monday, October 15	Total time	1 hour 15 minutes
Lesson unit	Chapter 6. CPU Scheduling		
Subject	CPU Scheduling Algorithms		
Lesson goals	• The student can explain the characteristics of 3 CPU scheduling algorithms – First Come First Served (FCFS), Shortest Job First (SJF), and Round Robin (RR) - • The student can draw Gantt charts and calculate the average waiting time using 3 CPU scheduling algorithms in a given environment		

Class procedure	Teaching and learning activities		Time
	Teacher	Student	
Introduction	• Describe lesson goals • Introduction to the CPU scheduler feature • Evaluation criteria of CPU scheduling algorithm	• Take notes in class • Answer teacher's questions	15
Development	• Describe 3 CPU scheduling algorithms - FCFS, SJF, RR • Shows how to draw Gantt charts and calculates average waiting time	• Ask questions you don't understand • Answer teacher's questions	30

Pair Check Activities	• Check group placement • Describe the problem to be solved • Assigning pair roles (organizer/ assistant) • Observation of the discussion process • Respond to paired groups' request for mediation	• Understand the given problem • Conduct discussions in pairs. The organizer actively suggests opinions first, and the assistant conducts the discussion in the form of refusing or accepting them • Check the problem solving method and answer • Determine the pair's answers after the final discussion. • Write the pair's final solution and answer in their quiz notes. • Present the pair's answers	20
Evaluation and Arrangement	• Summarize the lesson • Evaluate pair check activities • Preview of the next lesson		10

4. Effect of the Cooperative Learning Model

The cooperative study was carried out in 17 teams of 2 students, each of whom was a computer science major. In each pair, one played the odd-organizer role and the other played the even-organizer. If the lesson unit is an odd chapter, the student with the role of an odd-organizer becomes the organizer and actively discusses. In the case of an even chapter, the even-organizer student becomes the organizer.

In this study, the results of the exam score were analyzed to determine whether the learning outcomes differed according to the roles within the team. The null and alternative hypotheses for the statistical hypothesis test are as follows.

- H_0: $\mu \leq \mu_0$

  Ha: $\mu > \mu 0$

  - μ : Odd-Organizer Group's Exam Score

  - μ_0 : Even-Organizer Group's Exam Score

The results of the independent sample t-test analysis on the scores of odd-organizers and even-organizers for each exam question are shown in Table 3.

Table 3. One tailed t-test results

Question number	Chapter	Group	Mean	Std.Dev.	t	P(T<=t)
1	Odd Chapter	Odd-organizer group	8	4.25	1.55	0.0657
		Even-organizer group	6.88	4.61		
2	Odd Chapter	Odd-organizer group	8.47	24.14	-0.57	0.2873
		Even-organizer group	9.35	16.99		
3	Even Chapter	Odd-organizer group	10.29	21.72	-0.36	0.3590
		Even-organizer group	10.88	22.61		

4	Even Chapter	Odd-organizer group	7.71	10.71	-1.94	0.0305
		Even-organizer group	20.72	19.85		
5	Even Chapter	Odd-organizer group	11.35	16.87	-0.49	0.3150
		Even-organizer group	12.06	18.93		
6	Even Chapter	Odd-organizer group	17.88	49.99	-1.72	0.0471
		Even-organizer group	22.06	49.68		
7	Odd Chapter	Odd-organizer group	9.71	49.85	-0.15	0.4407
		Even-organizer group	10.06	43.68		
8	Odd Chapter	Odd-organizer group	17.82	34.40	0.15	0.4391
		Even-organizer group	17.52	27.26		
9	Even Chapter	Odd-organizer group	26.59	19.76	-1.25	0.1104
		Even-organizer group	28.29	11.97		

In most of the exam questions, the average score of the group that matched the even-odd types of the chapters in which the question was asked was high. However, questions 2 and 7, although not significant, showed lower scores for the organizer's role.

Of the 9 questions, 1,2,3,5,7,8 were related to concepts or theories, and 4,6,9 were related to the algorithm and practice. For questions related to concepts or theories, there was no significant difference in exam scores depending on the role of the team. On the other hand, in case of 4, 6, and 9 algorithms and practice-oriented questions, the difference in exam scores between groups was noticeable.

For questions 4 and 6, the t values were -1.94 and -1.72, respectively, and the P ($T <= t$) values were both less than the significance level alpha = 0.05. This means that there is a difference in learning outcomes depending on the student's role in the pair check model at the significance level of 0.05. In case of question 9, no significant difference was found at the significance level of 0.05, but more than 6% of score difference occurred.

In conclusion, in the case of concept or theoretical learning contents, there were no significant differences in exam scores according to roles in the paired check activity, but in the case of algorithm or practice related contents, there were differences in exam scores according to roles.

5. Conclusion

Educational institutions should provide training to produce high quality software personnel as a solution that can fundamentally secure the competitiveness of the software industry. Despite this situation, software-related classes at universities are largely composed of competitive or individual learning structures. Cooperative learning is a learning model that can complement the problems of competitive and individual learning. Cooperative learning is more effective in improving academic achievement than individual or competitive learning. In addition, most learners have the advantage of having a more desirable self-image through success experiences.

In this paper, the class procedure and instruction plan were designed to apply the pair check model which is a kind of cooperative learning in the operating system class. In addition, the results of the exam scoring were analyzed to determine whether the learning outcomes differ depending on the roles of the teams. As a result of the t-test, there was no difference in the learning outcomes according to the roles for the theory-related learning contents, but there was a significant difference in the learning outcomes according to the roles for the algorithm and practice-related learning contents.

References

[1] J.S. Kim, S.K. Han, S.H. Kim, S.W. Jung, J.M. Yang, U.D. Chang, and J.N. Kim, *A Research on the Development of Teaching and Learning Models for SW Education,* CR 2015-35, Korea Education & Research Information Service, 2015.

[2] S.G. Han and M.Y. Ryu, *Software Education for Computing Thinking*, SaengNeung Publishing, 2016.

[3] M.S. Jeong, *Understanding and Practice of Collaborative Learning,* KyoYookBook Publishing, pp. 384-386, 2002.

[4] Y.G. Byeon and G.H. Kim, *Theory and Practice of Cooperative Learning,* HakJiSa Publishing, 1999.

[5] R.E. Slavin, E.A. Hurley, and A. Chamberlain, "Cooperative learning and achievement: Theory and research," in *Handbook of psychology: Educational psychology,* Vol. 7 (pp. 177-198), John Wiley & Sons Inc, 2003. https://doi.org/10.1002/0471264385.

[6] H.K. Lim, *A Study on Effective Composing Partner in the Pairs Check Model,* Master's Thesis. Graduate School of Education, KookMin University, 2009.

[7] J.E. Kim, *The Effects on Academic Achievement & Mathematical Learning Attitude When Pairs Check Cooperative Learning,* Master's Thesis, Graduate School of Education, KangWon National University, 2012.

[8] B.Y. Park, *The Effect of Problem Posing with the Pairs Check Cooperative Learning on Mathematics Learning,* Master's Thesis. Graduate School of Education, DongKuk University, 2014.

[9] E.F. Gehringer, "Active and Collaborative Learning Strategies for Teaching Computing," in *Proceedings of the 2007 American Society for Engineering Education Annual Conference & Exposition*, American Society for Engineering Education, 2007. https://doi.org/10.3102/0034654317710096.

[10] J.C. Prey, "Cooperative Learning in an Undergraduate Computer Science Curriculum," in *Proceedings Frontiers in Education 1995 25th Annual Conference*. Engineering Education for the 21st Century, 1995. DOI:10.1109/FIE.1995.483139.

[11] R. Valdivia and M. Nussbaum, "Face-to-Face Collaborative Learning in Computer Science Classes," in *International Journal of Engineering Education*, January 2007.

[12] M. Masrom and A. Hakemi, "Technical and Infrastructural Aspects of Mobile Learning Adoption in Iran Higher Education," *International Journal of Internet, Broadcasting and Communication* Vol.11 No.1 1-7, 2019. http://dx.doi.org/10.7236/IJIBC.2019.11.1.1.

[13] Y. Gelogo and H.J. Kim, "Educational Paradigm Shift from E-Learning to Mobile Learning Toward Ubiquitous Learning," *International Journal of Internet, Broadcasting and Communication* Vol.4 No.1 8-12, 2012. http://dx.doi.org/10.7236/IJIBC.2012.4.1.8.

[14] S.W. Yoo, H.T. Lim,C.H. Kwon, S.J. Lee, S.D. Lee, and H.J. Jeon, *Education Method and Educational Technology,* YangSeoWon Publishing, 2010.