Flipped Classroom in Programming Course: A Systematic Literature Review

https://doi.org/10.3991/ijet.v16i02.15229

Hendrik (✉), Almed Hamzah
Universitas Islam Indonesia, Yogyakarta, Indonesia
hendrik@uii.ac.id

Abstract—The emerging trend technologies imply the increase in the new role of the future of jobs. Most of them required Computer Science-related skills a lot, especially programming. However, there is a skill gap for it. Some previous studies showed that many students feel difficulties in learning programming. In the last decades, there are many approaches to improve student's learning achievement through active learning methods. One of those approaches is the Flipped Classroom (FC) method. This study presents a systematic literature review of the FC approach in the programming course. The aim of this study is to explore the strategy of FC implementation in the programming class. A total of 32 papers from ACM Digital Library and IEEE Xplore Digital Library are selected for this review. We found four types of implementation of FC in programming, with the variations of the in-class or out-class activities.

Keywords—Flipped classroom, inverted classroom, programming, literature review, computational thinking, future skills

1 Introduction

Now, we live in the 4th industrial revolution era. This era drives to a lot of emerging trend technologies such as Big Data, Artificial Intelligence, robotics, Blockchain, the Internet of Things, 3D printing, and autonomous vehicles. Those technologies change our current world and how we live today. They also imply the increase in the new role of the future of jobs, which more specific to answer the demand for those technologies. The role such as AI and machine learning specialist, Big Data specialist, user experience and human-machine interaction designer, and software and application developer and analyst, are some of the examples [1].

For several decades, higher education institutions have implemented Computer Science (CS) curriculum. However, with the situation as mentioned earlier, there are many countries that aware of the importance of computer science-related skills for their future generation. Thus, they developed several initiatives to integrate CS into their primary and secondary school in the nation-wide curriculum [2][3].

The young people aware that technologies will affect their future. Among the top ten recognized trend technologies, the majority is related to computing [4]. Another report shows that ASEAN youth believe that among the hard skills, the highly valued skills are the technology use (e.g., social media, e-commerce, and e-payment), as well as technology design (e.g., software programming) [5].

Even though people realize that some future jobs required CS-related skills a lot, especially programming, there is a skill gap for it. As mentioned in [5], most young ASEAN people believe that software design and programming is one of the essential skills for their future, but they are not skillful on it. This current situation confirms the results of some previous studies that students, even for the computer science' students, feel difficulties in learning programming [6][7][8][9][10].

In the last decades, there are many approaches to improve student's learning achievement through active learning methods. It is suitable for programming-related courses where students are required to practice more often [11]. One of those approaches is the Flipped Classroom (FC) method. In this approach, the in-class activities in the traditional approach inverted into out-of-class activities and vice versa, with the help of information technology. The common practice of this approach is the students watch a pre-recorded lecturer video at home, and then in the class meeting, they do a quiz or some assignments related to the subject they learned before [12].

This method becomes popular in the last two decades. Many works studied the implementation of the FC method in a specific discipline such as economics [13], language study [14], and chemistry [15]. Meanwhile, some studies only focus on reviewing the FC implementation papers, such as [16] [17][18] [19]. Most of these literature review papers studied the FC implementation's articles in various disciplines. On the contrary, [18] only reviewed papers that discussed the implementation of FC in medical education. Unlike the other papers that analyzed the benefits and challenges of the FC implementation, [17] examined only the effect of FC on learning outcomes and student satisfaction.

Based on the issues as mentioned earlier, the authors focusing this literature review only on papers discussed the adoption of the FC in a programming course. This study aims to answer some research questions to help us understand the practice of FC in programming classes.

The followings are the further organization of this paper. In section 2, we elaborate on the method that we use to do this systematic literature review. Then, we present and discuss the results of the review in section 3. Finally, in section 4, we conclude and recommend some ideas for future work.

2 Method

We conducted this systematic literature review by adopting guidelines from [20]. In the following subsections, we discuss the detail of each step.

2.1 Research questions

To obtain the objective of this paper, we define three research questions as follows.

- **RQ1**: In which educational level, the FC most adopted in programming courses?
- **RQ2**: How is the FC approach implemented in programming courses?
- **RQ3**: What are the in-class activities and out-class activities that used in FC for programming classes?

2.2 Search process

The first author did all this process on 23rd January 2020. He searched on the leading academic databases in the field of computer science and information technology. The databases are ScienceDirect (sciencedirect.com), IEEE Explore (ieeexplore.ieee.org), and ACM digital library (dl.acm.org). The searching was done in abstract only, except for ScienceDirect since it does not provide an advance search to search only for the abstract. The search string for this process were:

"flipped classroom" AND "programming", and
"flipped learning" AND "programming"

2.3 Study selection

The author filtered the results only for papers published between 2015 to 2019. The initial search resulted in 126 papers from three databases. The first author then examined the relevance of the resulting papers by defining the inclusion criteria that is "the paper must describe the evaluation or analysis of the implementation of the flipped classroom approach in the programming course." We apply the inclusion criteria for the abstract and then the full text. For the papers obtained from ScienceDirect, no paper matches our criteria. Therefore, we got only 32 papers to review from two databases, IEEE and ACM. Table 1. lists the searching results.

2.4 Data extraction

Table 2 shows all of the selected papers to review as the result of data extraction process. For each article, we identify:

- The year of publication
- The author(s)'s affiliation country
- The level of education that the literature studied
- The name of the programming course
- The programming language used in the course
- Mode of implementation
- Number of meetings that implemented the FC approach.

Table 1. Search results from three databases

Database	Initial results	Relevant
ScienceDirect	33	0
IEEE Explore	64	24
ACM Digital Library	29	8
Total	**126**	**32**

Table 2. List of selected papers

Ref.	Year	Level of education	Course name	Programming Language	Mode of implementation	Amount of meetings
[21]	2018	UG	Introductory Programming	Python	H	Partial
[22]	2015	UG	OO Programming (OOP)	C, Java	H	Partial
[23]	2017	HS	ICT	Javascript (App Lab)	SA	Partial
[24]	2016	UG	Java Programming	Java	SA	Partial
[25]	2016	UG	Introductory Programming Programming	C++	SA	Partial
[26]	2015	UG	Data Structures	N/A	SA	Full
[27]	2015	UG	N/A	N/A	N/A	N/A
[28]	2015	UG	OOP and Design	Java	H	Full
[29]	2016	UG	Digital Game-based Learning	WebGL (JS API)/Unity	H	Partial
[30]	2015	UG	Introductory Programming	N/A	H	Partial
[31]	2018	UG	Introductory Programming	C#	H	Partial
[32]	2017	UG	Introductory Programming	Java	SA	Full
[33]	2015	UG	Java Programming	Java	H	Partial
[34]	2019	UG	Programming, OOP, Software Engineering	N/A	N/A	N/A
[35]	2018	UG	Data Structures	Java	H	Full
[36]	2016	UG	Introductory Programming; Data Structures	C and Java	SA	Full
[37]	2019	UG	C++ programming	C++	SA	Full
[38]	2017	UG	IOT Development (Advanced Programming)	Java	H	Full
[39]	2018	UG	Introductory Programming; Data Structures	C and Java	SA	Full
[40]	2016	UG	Foundations of Engineering 1	Mathlab	SA	Full
[41]	2019	UG	OOP	C++	H	Full
[42]	2019	UG	Web Programming	PHP, JS	H	Partial
[43]	2015	UG	Java Technology	Java	H	Partial
[44]	2018	UG	Introductory Computer Programming	N/A	N/A	N/A
[45]	2015	UG	Introductory Programming; Data Structures	N/A	N/A	Full
[46]	2016	UG	OOP, web application	Java	SA	Full
[47]	2016	UG	functional programming	Java	H	Partial
[48]	2019	UG	Web Programming	Java	SA	Full
[49]	2017	UG	Educational Programming Language (Scratch Programming)	Scratch	SA	Full

[50]	2017	M	Programming Language	N/A	SA	Partial
[51]	2018	HS	Introductory Programming	Python and Ruby	SA	Full
[52]	2019	UG	Introductory Programming	Java	H	Partial

In this subsection, we elaborate on the demographic analysis of the selected papers. Most of the selected papers are conference proceedings articles (31 of 32; 97%), and only one paper (1 of 32; 3%) is a journal article. Fig. 1 depicts the distribution of selected papers based on their publication year. As we can see from Fig. 1, there was a decline in the number of publications from 2015 until 2017. However, it was a relatively steady number from after 2017. On average, six papers per year studied the adoption of FC in the programming course.

Based on the origin countries of the authors' affiliation, the selected papers coming from seventeen countries. Table 3. lists the distribution of these countries. The table also indicates the Worldwide Educating for the Future Index (WEFFI) 2019. The index evaluates the effectiveness of a country's education system in providing young people with future-oriented capabilities such as critical thinking, collaboration, creativity, leadership, problem-solving, and entrepreneurship. The WEFFI 2019 comprises twenty indicators of three thematic categories: policy environment, teaching environment, and socio-economic environment [53].

We use this index because it is more relevant to our case since programming is a part of the skills needed in the future. Therefore, we do not refer to the Education Index, which is part of the Human Development Index, as the index measures the education quality of a country in general.

Among eight countries in the high score group of WEFFI 2019, only authors from four countries: Finland, Sweden, Singapore, and Canada, contributed to selected papers. Those countries, except Sweden, contributed more than one paper studied the FC in the programming course. Meanwhile, most countries of the authors of the selected papers (10 of 17; 59%) were included in the medium score group. However, the rest countries (3 of 17; 17%): Macedonia, Qatar, and Taiwan, are not listed in the WEFFI 2019. Specifically for Taiwan, it was listed as the 17th rank in the WEFFI 2018 and made it as one of the medium score group [54]. Surprisingly, Indonesia, that relatively less technologically advanced than China and India, ranked higher than both countries in the WEFFI 2019. Nevertheless, only one paper provided by an author from Indonesia.

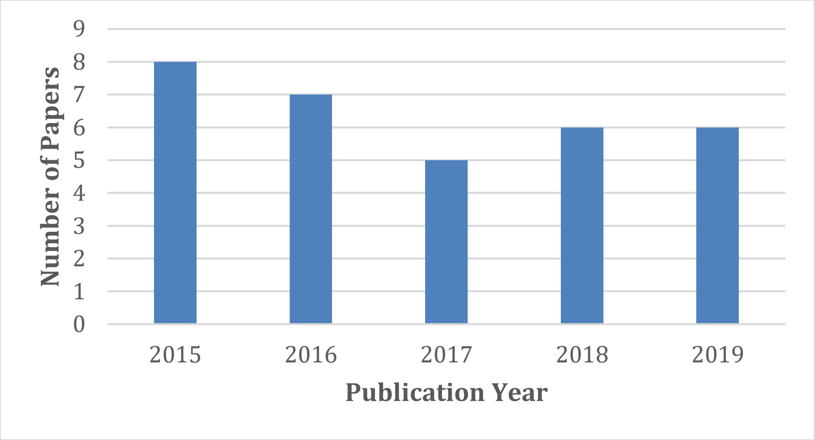

Fig. 1. Number of selected papers by publication year

Table 3. Distribution of papers based on the authors' affiliation country

Country	Number of Papers	Worldwide Educating for the Future Index 2019
Brazil	1	51.9
Canada	2	79.1
China	3	53.8
Finland	3	84.8
Germany	1	74.9
Hong Kong	2	73
India	2	53
Indonesia	1	57.9
Japan	1	74.2
Kazakhstan	2	57
Macedonia	1	N/A
Qatar	3	N/A
Singapore	2	79.7
South Korea	1	72.6
Sweden	1	84.3
Taiwan	4	N/A
United States	2	61.4

Regarding the course names and the programming language used in the selected papers, Fig. 2 and Fig. 3 depict the distribution of them. Almost half of the authors experiment with FC for the introductory programming class. Other courses in the top 5, respectively Data Structures, Object-Oriented Programming, Web Programming, and Java Technology. Meanwhile, Java dominates the programming language applied in the FC programming course. In the top 5, it followed by C++, Javascript, C, and Python.

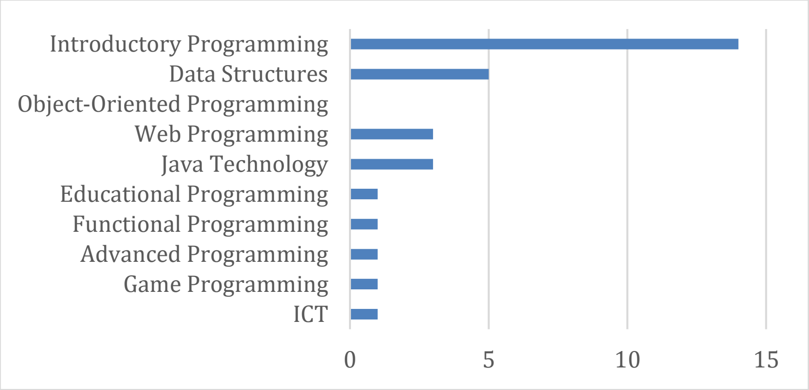

Fig. 2. The list of course names in the reviewed papers

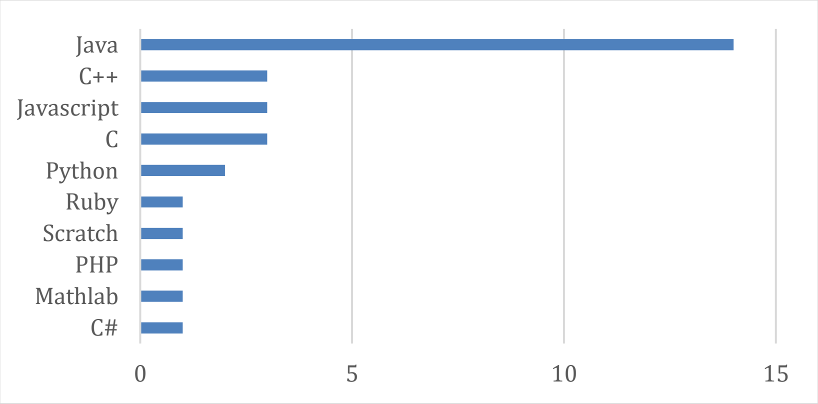

Fig. 3. The programming languages in the reviewed papers

3 Results and Discussion

In this section, we present and discuss the answers to our research questions as follows.

3.1 RQ1. In which educational level, the FC most adopted in programming courses?

Even though Bergmann and Sams firstly introduced the FC at the K-12 level [12], our finding shows that most of the studies conducting the FC at higher education level (91% at the undergraduate level and 3% at the graduate level [50]). Conversely, only two papers ([23] and [51]) show the implementation at the K-12 level, more specifically for high school students (see Fig. 4).

This finding is supported by [55]. It mentioned that there are not many articles that study the implementation of the FC at the K-12 level. Another reason that supports this finding is that some studies show that the ability of students of higher education in terms of self-regulation, restraint ability, independence, and responsibility in learning, are more reliable than K-12 students [56][57][58]. It is well-known that FC requires those capabilities in the learning process [56] [59][60].

3.2 RQ2. How is the FC approach implemented in programming courses?

As far as our knowledge, there is no literature review discussing how the FC is implemented. In this study, we review this condition based on two characteristics: the mode of approach and the number of the course meetings that performed using the FC.

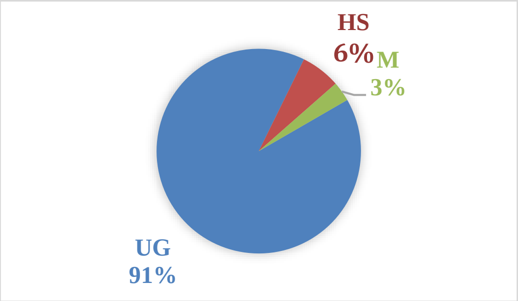

Fig. 4. The FC implementation in various education levels

In the mode of approach, we analyzed and labeled the papers either as a single approach (SA) or a combination with another method (Hybrid/H). It should be noted that a SA implementation means that the course implementing the FC without combination with another approach but could be a complement of the traditional method [23][24]. On the other hand, the hybrid applying the FC with another strategy, such as project-based learning [38][42], problem-based learning [29][41], or collaborative learning [22][52]. As we can see from Fig. 5, we can analyze the mode of approach from most of the reviewed papers (88%). Among them, the studies that are implementing the SA share the same proportion with the others that adopting the hybrid.

Meanwhile, for the number of course meetings that are performed using the FC, we classify the papers into two categories: partial and full. Partial means the FC is only applied for several sessions[31] [47], while the full means it is implemented for all or most of the meetings [35][37]. Commonly for the partial implementation, the FC was implemented between four and seven sessions. Fig. 6 shows the proportion of the papers based on this category. We can recognize that the portion of the full implementation slightly more dominant than the partial one, but both relatively almost the same.

Since the ratio in both characteristics is relatively the same, we found a pattern that shows a relationship between both attributes. Therefore, we classify the mode of the FC implementation into four categories, as depicted in Fig. 7.

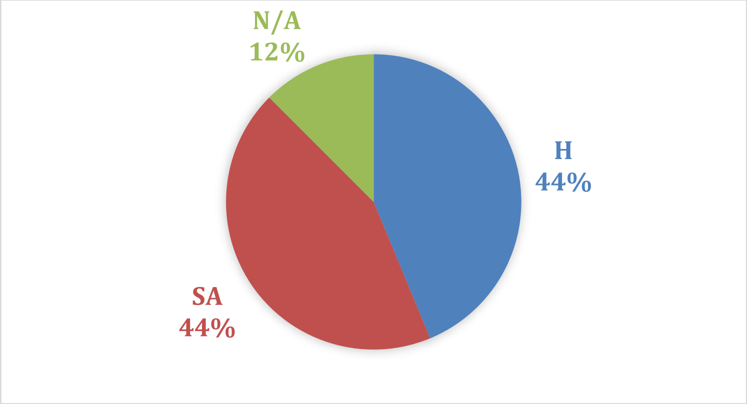

Fig. 5. Distribution of papers based on the mode of approach

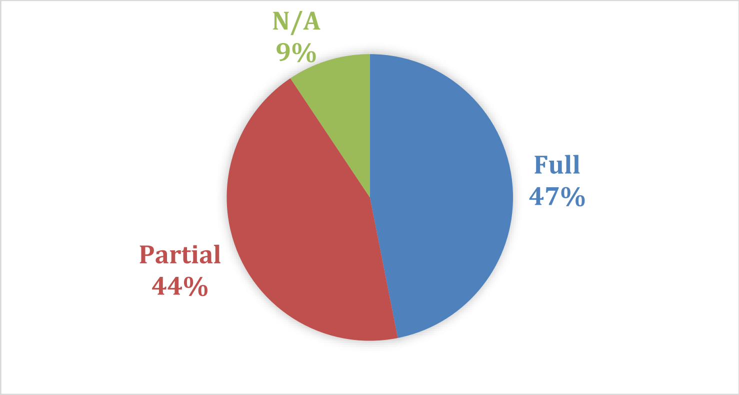

Fig. 6. FC implementation based on the number of meetings

Followings are the four groups of the implementation's form:

- **Pilot implementation**: In this mode, the FC is implemented as a single approach mode and for only a partial number of meetings. Usually, it is a pilot project implementation to study the feasibility of the approach. Mostly used as the complimentary of the traditional method.
- **Solitary implementation**: This mode implements the FC as one and only approach for the whole meetings. Several works with this implementation mode presented their evaluation of the repetitive FC adoption in several semesters or academic years. For example, [36], [39], and [51].
- **Mixed implementation**: This form of implementation shows us the combination of the FC with another approach in several meetings.
- **Fused implementation**: Similar to the mixed mode, the fused also blending the TC with another approach. However, it is for full meetings. In this mode, the other method was implemented as a variation of in-class or out-of the-class activities.

3.3 RQ3. What are the in-class activities and out-class activities that used in FC for programming classes?

The essential thing in FC is the activities, both in-class and out-of-class activities. Hence, we provide an analysis of the FC activities in the programming course used by the researchers of our selected studies. We hope this can give insight and guidance for the teacher who teaches the programming class.

As listed in Table 4. , the top five in-class activities in the programming course are hands-on-experience (40,63%), small briefings (25%), quizzes (25%), assignments with teacher assistance (21,88%), and questions and answers (18,75%). There are several reasons for these findings. First, in the programming course, practicing to develop a program or application is more crucial than only focusing on understanding the language syntax [25] [28]. Therefore, the use of practical experience and programming assignments could improve the skill of the students. Moreover, through programming assignments, it could help the students to develop their problem-solving skills [48] and stimulate their ability for self-learning [51].

Second, through quizzes, the students are motivated to prepare before joining the class. Hence, they would understand the concept of programming or the syntax of the programming language they would use in the practical session or the programming assignments. It also helps the teachers to obtain a general overview of the students' understanding of the subject. If there is a misunderstanding in the concept, the teachers can revise it through the small briefings or the questions and answers activity [24] [30].

Regarding activities outside the classroom, information or knowledge transfer in FC is mostly conveyed via video. Even [61] stated that to comply as an FC implementation, we must use lectures video. However, as we can see in Table 5. , the out-of-class activity in the FC in programming is varied and relatively share the same percentage. The most activity in this group is watching videos. However, no more than 60% of researchers employed it. In line with the finding of [16], another activity that quiet a lot be used is the quiz (34%). Interestingly, several tasks relatively share the same percentage between 15% and 19%. Those activities are readings, online modules, self-exercise, assignments, and interactive tutorials.

Instead of using videos, [37] only used readings assignment for their FC implementation. They argued that this activity helps the student to develop their life-long-learning skill. In the software development area, the developers should be familiar with reading e-books, manuals, or documentation relating to the technology they used, which is rapidly changing.

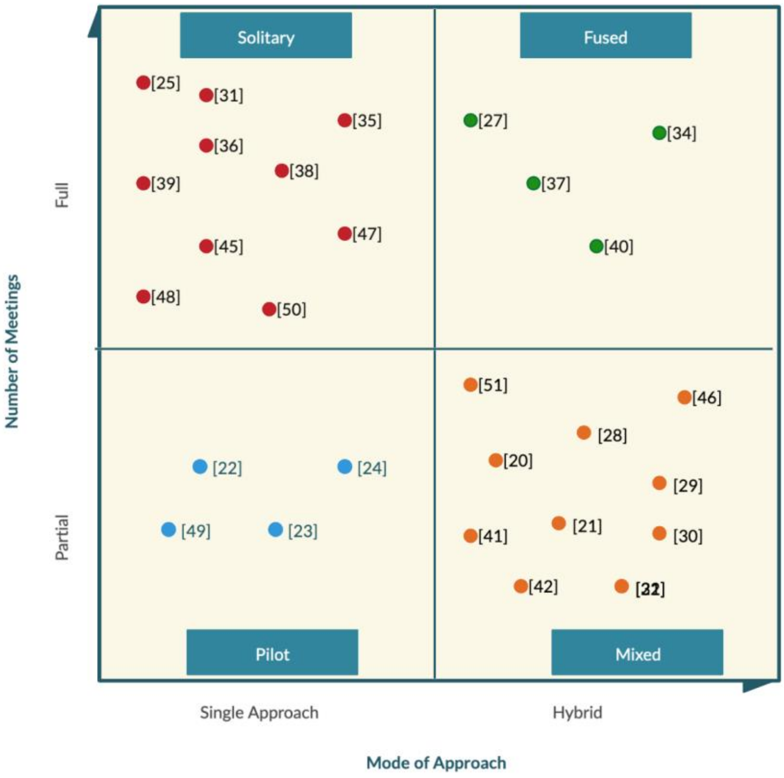

Fig. 7. Quadrant classification of FC implementation mode

The use of online modules, which integrates videos, self-quizzes, and programming assignments, helps improve student learning [32]. Almost similar to the previous one, interactive tutorials comprise of animation, interactive tools, questions set, and embedded homework. This kind of activity encourages students to learn by doing, which is known to be more effective for programming education that requires the students to do more practice. There are also online platforms that support interactive tutorials for programming education, such as zyBooks and openDSA [48].

It should be noted, most researchers use more than 1 activity for either in-class or out-of-class activities in their implementation. On average, they employed 2-3 variation activities for their programming class. This finding aligns with the suggestion to use a few activities during the FC implementation.

Table 4. In-class activities in FC for programming course

Activity	Frequency	Percentage
Hands-on experiments	13	40,63
Small briefings	8	25,00
Quizzes	8	25,00
Assignments with teacher assistance	7	21,88
Questions and answers	6	18,75
Discussion	5	15,63
Problem solving	5	15,63
Students' presentations	5	15,63
Feedback	4	12,50
pair programming	4	12,50
Group discussions	4	12,50
Small group activities	3	9,38
examination	2	6,25
Reflection	2	6,25
Concept mapping	1	3,13
Audience responses (Clicker)	1	3,13
Group projects	1	3,13
Assignments	1	3,13
Collaborative group work	1	3,13

Table 5. Out-of-class activities in FC for programming course

Activity	Frequency	Percentage
Videos	18	56,25
Quizzes	11	34,38
Online modules	6	18,75
Self-exercise	6	18,75
Readings	6	18,75
Assignments	5	15,63
Interactive tutorials	5	15,63
Writing	1	3,13

4 Conclusion and Future Work

In this study, we conducted a systematic literature review on the implementation of the flipped classroom method in the programming course. Recently, there are many initiatives from the government of the world to integrate the computer science curriculum, especially programming, into their education system, even from the very entry-level. However, our research found that there have been little endeavors to practice the FC for programming in the K-12 education level. We think we need to extend the research on this method for this education level, as there are many potentials that can be explored to achieve the advantages of the approach.

Our research uncovers four modes of FC implementation. Hence, it is interesting to conduct future research that investigates how those modes can assist in the strategy of

FC implementation. Moreover, future studies can examine the distinction among those modes in the matter of effectiveness and efficiency of FC utilization.

As we know, the programming course requires students to have more practice time to master the skill. We recognized that most in-class activities in this study related to practical activities instead of active learning activities. We believe this situation will be the same for the other engineering subjects. However, to provide more evidence, future researchers can examine it for courses with similar characteristics to programming.

Many works suggest the use of video lectures to help students prepare themselves before the class. Still, in the context of programming class, we should consider the activity involves a more active process. For example, we can utilize online modules or interactive tutorials, as exhibited by this study. Moreover, we can also consider the use of another interactive technologies such as eye-tracking [62] and brain-computer interfaces [11], [63]. We recommend studying the development of this process model and its learning effect for the FC class.

5 References

[1] World Economic Forum, “The Future of Jobs Report,” 2018.

[2] S. Bocconi et al., Developing Computational Thinking in Compulsory Education - Implications for policy and practice, no. June. 2016.

[3] P. Seow, C.-K. Looi, M.-L. How, B. Wadhwa, and L.-K. Wu, “Educational Policy and Implementation of Computational Thinking and Programming: Case Study of Singapore,” in Computational Thinking Education, 2019. https://doi.org/10.1007/978-981-13-6528-7_19

[4] “Youth Voice for the Future of Work What do young people think about the future of work?”

[5] World Economic Forum, “ASEAN Youth Technology, Skill and the Future of Work,” Cent. New Econ. Soc., no. August, pp. 1–17, 2019.

[6] M. D. Siti Rosminah and M. A. Ahmad Zamzuri, “Difficulties in learning Programming: Views of students,” in 1st International Conference on Current Issues in Education (ICCIE2012), 2012.

[7] E. Lahtinen, K. Ala-Mutka, and H. M. Järvinen, “A study of the difficulties of novice programmers,” in Proceedings of the 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education, 2005, https://doi.org/10.1145/1067445.1067453.

[8] T. H. Park and S. Wiedenbeck, “Learning web development: Challenges at an earlier stage of computing education,” in ICER’11 - Proceedings of the ACM SIGCSE 2011 International Computing Education Research Workshop, 2011, https://doi.org/10.1145/2016911.2016937.

[9] M. Rahmat, S. Shahrani, R. Latih, N. F. M. Yatim, N. F. A. Zainal, and R. A. Rahman, “Major Problems in Basic Programming that Influence Student Performance,” Procedia - Soc. Behav. Sci., 2012, https://doi.org/10.1016/j.sbspro.2012.09.277

[10] S. Xinogalos and T. H. Kaskalis, “The challenges of teaching web programming: Literature review and proposed guidelines,” in WEBIST 2012 - Proceedings of the 8th International Conference on Web Information Systems and Technologies, 2012, https://doi.org/10.5220/0003960902070212.

[11] J. Katona and A. Kovari, “A Brain–Computer Interface Project Applied in Computer Engineering,” IEEE Trans. Educ., vol. 59, no. 4, pp. 319–326, 2016. https://doi.org/10.1109/te.2016.2558163

[12] S. A. Bergmann J., Flip Your Classroom: Reach Every Student in Every Class Every Day. 2012. https://doi.org/10.1177/073989131401100120

[13] J. Caviglia-Harris, "Flipping the undergraduate economics classroom: Using online videos to enhance teaching and learning," South. Econ. J., 2016, https://doi.org/10.1002/soej.12128.

[14] K. Chilingaryan and E. Zvereva, "Methodology of Flipped Classroom as a Learning Technology in Foreign Language Teaching," Procedia - Soc. Behav. Sci., 2017, https://doi.org/10.1016/j.sbspro.2017.02.236.

[15] P. Baepler, J. D. Walker, and M. Driessen, "It's not about seat time: Blending, flipping, and efficiency in active learning classrooms," Comput. Educ., 2014, https://doi.org/10.1016/j.compedu.2014.06.006.

[16] G. Akçayır and M. Akçayır, "The flipped classroom: A review of its advantages and challenges," Comput. Educ., 2018, https://doi.org/10.1016/j.compedu.2018.07.021.

[17] T. Låg and R. G. Sæle, "Does the Flipped Classroom Improve Student Learning and Satisfaction? A Systematic Review and Meta-Analysis," AERA Open, 2019, https://doi.org/10.1177/2332858419870489.

[18] A. S. Kraut et al., "The flipped classroom: A critical appraisal," West. J. Emerg. Med., 2019, doi: 10.5811/westjem.2019.2.40979.

[19] C. K. Lo, "Systematic Reviews on Flipped Learning in Various Education Contexts," in Systematic Reviews in Educational Research, 2020. https://doi.org/10.1007/978-3-658-27602-7_8

[20] B. Kitchenham, O. Pearl Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, "Systematic literature reviews in software engineering - A systematic literature review," Information and Software Technology. 2009, https://doi.org/10.1016/j.infsof.2008.09.009.

[21] H. C. Hung, "Flipped learning with live-coding approach for programming concepts learning," Proc. - 2018 1st Int. Cogn. Cities Conf. IC3 2018, pp. 223–224, 2018, https://doi.org/10.1109/ic3.2018.00-17.

[22] Y. Hayashi, K. I. Fukamachi, and H. Komatsugawa, "Collaborative learning in computer programming courses that adopted the flipped classroom," Proc. - 2015 Int. Conf. Learn. Teach. Comput. Eng. LaTiCE 2015, pp. 209–212, 2015, https://doi.org/10.1109/latice.2015.43.

[23] O. S. Yan and G. Cheng, "Exploring the impact of flipped classroom on students' acceptance of programming in secondary education," Proc. 2017 IEEE Int. Conf. Teaching, Assess. Learn. Eng. TALE 2017, vol. 2018-Janua, no. December, pp. 246–249, 2017, https://doi.org/10.1109/tale.2017.8252341.

[24] T. S. Indi, "An Experience Report of Flipped Classroom Strategy Implementation for Java Programming Course," Proc. - IEEE 8th Int. Conf. Technol. Educ. T4E 2016, no. 2, pp. 240–241, 2017, https://doi.org/10.1109/t4e.2016.059.

[25] S. Alhazbi, "Using flipped classroom approach to teach computer programming," Proc. 2016 IEEE Int. Conf. Teaching, Assess. Learn. Eng. TALE 2016, no. December, pp. 441–444, 2017, https://doi.org/10.1109/tale.2016.7851837.

[26] G. C. Lee and P. L. Lee, "Data structures in flipped classroom: Students' effort and preference," Proc. - 2015 Int. Conf. Learn. Teach. Comput. Eng. LaTiCE 2015, pp. 152–155, 2015, doi: 10.1109/LaTiCE.2015.28.

[27] C. Pe Rosiene and J. A. Rosiene, "Flipping a programming course: The good, the bad, and the ugly," Proc. - Front. Educ. Conf. FIE, vol. 2014, 2015, https://doi.org/10.1109/fie.2015.7344151.

[28] H. Jonsson, “Using flipped classroom, peer discussion, and just-in-time teaching to increase learning in a programming course,” Proc. - Front. Educ. Conf. FIE, vol. 2015, 2015, https://doi.org/10.1109/fie.2015.7344221.

[29] W. C. Hsu and H. C. K. Lin, “Impact of Applying WebGL Technology to Develop a Web Digital Game-Based Learning System for Computer Programming Course in Flipped Classroom,” Proc. - 5th Int. Conf. Educ. Innov. through Technol. EITT 2016, no. 2004, pp. 64–69, 2017, https://doi.org/10.1109/eitt.2016.20.

[30] A. G. De Oliveira Fassbinder, T. G. Botelho, R. J. Martins, and E. F. Barbosa, “Applying flipped classroom and problem-based learning in a CS1 course,” Proc. - Front. Educ. Conf. FIE, vol. 2015, 2015, https://doi.org/10.1109/fie.2015.7344223.

[31] Y. H. Chang, A. C. Song, and R. J. Fang, “The Study of Programming Language Learning by Applying Flipped Classroom,” 1st IEEE Int. Conf. Knowl. Innov. Invent. ICKII 2018, no. 1, pp. 286–289, 2018, https://doi.org/10.1109/ickii.2018.8569171.

[32] J. Elmaleh and V. Shankararaman, “Improving student learning in an introductory programming course using flipped classroom and competency framework,” IEEE Glob. Eng. Educ. Conf. EDUCON, no. April, pp. 49–55, 2017, https://doi.org/10.1109/educon.2017.7942823.

[33] Z. Li and Q. X. Geng, “The application of hybrid flipped classroom in the course of Java programming,” Proc. - 2015 7th Int. Conf. Inf. Technol. Med. Educ. ITME 2015, pp. 637–641, 2016, https://doi.org/10.1109/itme.2015.95.

[34] M. Fetaji, B. Fetaji, and M. Ebibi, “Analyses of possibilities of Flipped Classroom in Teaching Computer Science Courses,” 2019 42nd Int. Conv. Inf. Commun. Technol. Electron. Microelectron. MIPRO 2019 - Proc., pp. 747–752, 2019, https://doi.org/10.23919/mipro.2019.8757126.

[35] M. K. P. Peethambaran, V. G. Renumol, and S. Murthy, “Flipped Classroom Strategy to Help Underachievers in Java Programming,” Proc. - 2018 6th Int. Conf. Learn. Teach. Comput. Eng. LaTiCE 2018, pp. 44–49, 2018, https://doi.org/10.1109/latice.2018.000-7.

[36] B. Tyler and M. Abdrakhmanova, “Flipping the CS1 and CS2 classrooms in Central Asia,” Proc. - Front. Educ. Conf. FIE, vol. 2016-Novem, pp. 1–5, 2016, https://doi.org/10.1109/fie.2016.7757739.

[37] U. Schäfer, “Teaching modern C++ with flipped classroom and enjoyable iot hardware,” IEEE Glob. Eng. Educ. Conf. EDUCON, vol. April-2019, no. April, pp. 910–919, 2019, https://doi.org/10.1109/educon.2019.8725068.

[38] C. U. Lei et al., “Teaching Internet of Things: Enhancing learning efficiency via full-semester flipped classroom,” Proc. 2017 IEEE Int. Conf. Teaching, Assess. Learn. Eng. TALE 2017, vol. 2018-Janua, no. December, pp. 56–60, 2017, https://doi.org/10.1109/tale.2017.8252304.

[39] B. Tyler and A. Yessenbayeva, “A comparison of flipped programming classroom models - Results by gender and major,” in Proceedings - Frontiers in Education Conference, FIE, 2019, https://doi.org/10.1109/fie.2018.8658809.

[40] B. Ahmed, A. Aljaani, and M. I. Yousuf, “Flipping introductory engineering design courses: Evaluating their effectiveness,” IEEE Glob. Eng. Educ. Conf. EDUCON, vol. 10-13-Apri, no. April, pp. 234–239, 2016, https://doi.org/10.1109/educon.2016.7474559.

[41] G. Wang, H. Zhao, Y. Guo, and M. Li, “Integration of flipped classroom and problem based learning model and its implementation in university programming course,” 14th Int. Conf. Comput. Sci. Educ. ICCSE 2019, no. Iccse, pp. 606–610, 2019, https://doi.org/10.1109/iccse.2019.8845525.

[42] Hendrik, “Flipping Web Programming Class: Student’s Perception and Performance,” in Proceedings of the 2019 IEEE 11th International Conference on Engineering Education, ICEED 2019, 2019, https://doi.org/10.1109/iceed47294.2019.8994822.

[43] Z. Li, S. Y. Zhang, and Q. X. Geng, "Application research of Java technology course teaching reform based on the flipped classroom," 10th Int. Conf. Comput. Sci. Educ. ICCSE 2015, no. Iccse, pp. 814–818, 2015, https://doi.org/10.1109/iccse.2015.7250357.

[44] S. Alhazbi and O. Halabi, "Flipping introductory programming class: Potentials, challenges, and research gaps," ACM Int. Conf. Proceeding Ser., pp. 27–32, 2018, https://doi.org/10.1145/3290511.3290552

[45] D. Horton and M. Craig, "Drop, fail, pass, continue: Persistence in CS1 and beyond in traditional and inverted delivery," SIGCSE 2015 - Proc. 46th ACM Tech. Symp. Comput. Sci. Educ., pp. 235–240, 2015, https://doi.org/10.1145/2676723.2677273.

[46] A. Knutas, A. Herala, E. Vanhala, and J. Ikonen, "The flipped classroom method: Lessons learned from flipping two programming courses," ACM Int. Conf. Proceeding Ser., vol. 1164, no. October 2017, pp. 423–430, 2016, https://doi.org/10.1145/2983468.2983524.

[47] V. Isomöttönen and V. Tirronen, "Flipping and blending-an action research project on improving a functional programming course," ACM Trans. Comput. Educ., vol. 17, no. 1, 2016, https://doi.org/10.1145/2934697.

[48] L. Cao and M. Grabchak, "Interactive PreparatoryWork in a Flipped Programming Course," CompEd 2019 - Proc. ACM Conf. Glob. Comput. Educ., pp. 229–235, 2019, doi: 10.1145/3300115.3309520.

[49] J. A. Kim and H. J. Kim, "Flipped learning of scratch programming with code.org," ACM Int. Conf. Proceeding Ser., pp. 68–72, 2017, doi: 10.1145/3175536.3175542.

[50] A. J. Kaijanaho, "Teaching master's degree students to read research literature Experience in a programming languages course 2002–2017," ACM Int. Conf. Proceeding Ser., pp. 143–147, 2017, https://doi.org/10.1145/3141880.3141893.

[51] H. N. Mok and V. R. Rao, "Introducing basic programming to pre-university students: A successful initiative in Singapore," in 2018 17th International Conference on Information Technology Based Higher Education and Training, ITHET 2018, 2018, https://doi.org/10.1109/ithet.2018.8424783.

[52] A. Mohamed, "Designing a CS1 programming course for a mixed-ability class," Proc. 24th West. Can. Conf. Comput. Educ. WCCCE 2019, pp. 10–15, 2019, doi: 10.1145/3314994.3325084.

[53] The Economist Intelligence Unit, "Worldwide Educating for The Future Index 2019 From policy to practice," Econ., 2019.

[54] The Economist Intelligence Unit, "Worldwide Educating for The Future Index 2018," Econ., no. 2, p. 29, 2019.

[55] C. K. Lo and K. F. Hew, "A critical review of flipped classroom challenges in K-12 education: possible solutions and recommendations for future research," Res. Pract. Technol. Enhanc. Learn., vol. 12, no. 4, pp. 2–22, 2017, https://doi.org/10.1186/s41039-016-0044-2.

[56] H. N. Mok, "Teaching tip: The flipped classroom," J. Inf. Syst. Educ., vol. 25, no. 1, p. 7, 2014.

[57] R. Elliott, "Do students like the flipped classroom? An investigation of student reaction to a flipped undergraduate IT course," in Proceedings - Frontiers in Education Conference, FIE, 2015, https://doi.org/10.1109/fie.2014.7044070.

[58] P. Pugsee, "Effects of using flipped classroom learning in object-oriented analysis and design course," in Ubi-Media 2017 - Proceedings of the 10th International Conference on Ubi-Media Computing and Workshops with the 4th International Workshop on Advanced E-Learning and the 1st International Workshop on Multimedia and IoT: Networks, Systems and Applications, 2017, https://doi.org/10.1109/umedia.2017.8074130.

[59] J. O'Flaherty and C. Phillips, "The use of flipped classrooms in higher education: A scoping review," Internet High. Educ., vol. 25, pp. 85–95, 2015, https://doi.org/10.1016/j.iheduc.2015.02.002.
[60] M. Tanner and E. Scott, "A flipped classroom approach to teaching systems analysis, design and implementation," J. Inf. Technol. Educ. Res., vol. 14, pp. 219–241, 2015, doi: 10.28945/2266.
[61] J. L. Bishop and M. A. Verleger, "The flipped classroom: A survey of the research," in ASEE Annual Conference and Exposition, Conference Proceedings, 2013. https://doi.org/10.18260/1-2--22585
[62] A. Kovari, J. Katona, and C. Costescu, "Evaluation of Eye-Movement Metrics in a Software Debbuging Task using GP3 Eye Tracker," Acta Polytech. Hungarica, vol. 17, no. 2, 2020. https://doi.org/10.12700/aph.17.2.2020.2.4
[63] J. Katona and A. Kovari, "Examining the learning efficiency by a brain-computer interface system," Acta Polytech. Hungarica, vol. 15, no. 3, pp. 251–280, 2018

6 Authors

Hendrik is a senior lecturer at the Department of Informatics, Universitas Islam Indonesia. His research interests are involved in Linked Data, Semantic Web, e-learning, Educational Technology, Business Intelligence, and Information Systems.

Almed Hamzah is a researcher at the Department of Informatics, Universitas Islam Indonesia. He is a member of Center of Study Enterprise Information System at Universitas Islam Indonesia. His researches are in the fields of Mobile Learning, Adaptive Learning, and Human Computer Interaction. Email: almed.hamzah@uii.ac.id

Article submitted 2020-04-26. Resubmitted 2020-08-08. Final acceptance 2020-08-10. Final version published as submitted by the authors.