Teaching programming with flipped classroom method: a study from two programming courses

Antti Herala   Erno Vanhala   Antti Knutas   Jouni Ikonen

Lappeenranta University of Technology

Department of Innovation and Software

P.O.Box 20

FI-53851 Lappeenranta, Finland

firstname.lastname@lut.fi

ABSTRACT

Flipped classroom teaching method, where theory is studied at home and exercises are done in the classroom, is gaining foothold in teaching. The method has been used with different approaches and guidelines, yet a single unified process has not been described. In this work we compare existing literature to our experiences in teaching. As our main result we outline a simple process description and guidelines for building a course structure with flipped classroom. Flipping the classroom has been found to be more efficient than traditional lecture-exercises model and the our findings support this. Therefore we recommend teachers to explore the possibility of utilizing the method.

CCS Concepts

• **Social and professional topics~Computer science education**
• *Applied computing~Collaborative learning*

Keywords

Programming; Flipped classroom; Reversed classroom; Peer learning; Peer teaching

1. INTRODUCTION

Since the introduction of the Internet, teaching and learning have changed significantly [5]. The traditional university lecturing with blackboard and overhead projector is competing with online courses and digital learning material. At the same time paper exams and individual essays are replaced with online exams and peer learning. This has increased pressure on teachers to improve their teaching perspectives, methods and skills. The new task of the lecturer is to guide and facilitate the learning, while the students take care of their individual learning based on their own learning styles.

Flipped classroom is a method where students first learn theory independently outside classroom and then practice in class with guidance from a teacher. Recently the method has gained publicity [1] and it has been praised to improve learning and reduce the teachers' workload [4]. The flipped classroom method has been utilized at various levels and has been found suitable for programming education [3]. While the use of the method has

increased, there is still a lack of guidelines about how and why the method should be applied.

In this study we researched *how to create a general approach for applying the flipped classroom principles to a programming course.*

2. RELATED RESEARCH

The computer science and programming education have been studied from various perspectives. Here we focus only on the flipped classroom method with both peer learning and peer teaching.

The concept of flipped classroom was developed in step with technological advancements, with the provision of wide amount of learning materials through easily accessible channels. This gives room to a philosophy where teacher's and student's contact time is aimed to be used as effectively as possible [7]. Aided by technology, students can study the necessary basic material independently, enabling more advanced topics, peer learning and peer teaching in exercise sessions.

Peer learning and peer teaching are collaborative teaching methods where students are not a passive audience in teaching, but instead they participate in it [2]. Peer learning has been found suitable especially in the environment where problem and case examples are commonly used. When students are familiar with the basics before arriving to the class, they can collaborate with peers more, leading to an improved learning process [6, 8].

3. FLIPPED CLASSROOM PROCESS

The overall answer to how to build a course and improve learning outcomes with flipped classroom method is multilateral and in the Figure 1 we present a two step learning process for utilizing the flipped classroom method in programming. Because programming as an activity centers around on writing code, the events in classroom concentrate on problem solving and programming. The learning activities outside classroom are focused on reading programming literature and watching videos and it is controlled with quizzes that present questions on the weekly changing topic.

The quizzes control that the students have sufficient theoretical knowledge to start working on the exercise tasks. On programming course the main task is to solve a problem by programming an application. However, theoretical knowledge is required, since there are multiple skills required from the graduating students outside programming. These skills are necessary for a computer science professional and relevant to programming but cannot be measured or taught through programming in a programming course.

The peer support can be utilized to support learning in the weekly exercises and especially larger programming projects can be done in pairs or in groups. On one hand it gives students the
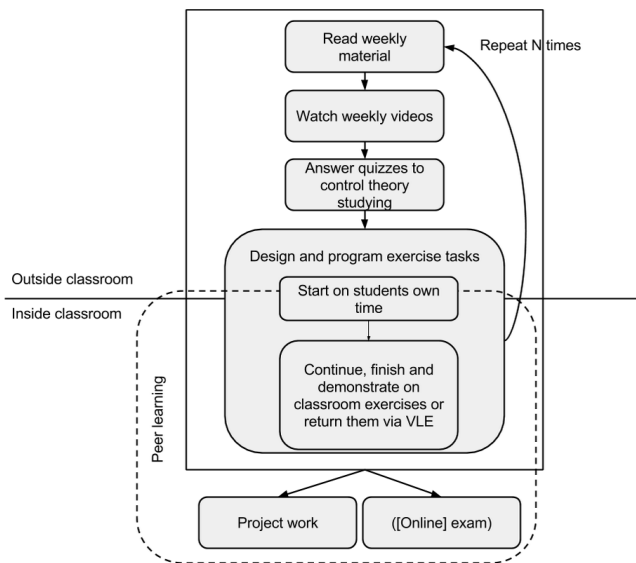
Figure 1. The process flow utilized in flipped classroom learning in programming.

possibility to design the project together and get ideas from a partner. On the other hand teachers need to take care that free-rider issue does not rise.

The Figure 1 is a synthesis of the literature and our experiences on how the flipped classroom method can be applied to teaching programming. The presented process is repeated as necessary depending on the topics and the number of teaching weeks. The course can also have project work and/or exams that are not included in this repeatable process.

Because time is a limited resource in the classroom, we recommend that students are encouraged to start their work early and finish their work in a in-class event where they can ask help and get recommendations and guidance to their problems.

We also have noted how both literature and our experiences support peer learning where students work in pairs and collaborate on learning new things. Peer learning is sufficient from various aspects: not only it helps students to go through material more thoroughly, but it also transformers teacher's role from lecturer to active supporter and mentor.

The following guidelines have been identified from the existing literature and our teaching experiences.

- Create or curate videos in addition to text-based material
  - Video curating suggested, if the instructor intends to hold short lectures
- Evaluate the material with other experts in the field
- Use weekly quizzes to evaluate the level of understanding and satisfaction of students
- Strictly integrate the theory and material to the course
- Encourage students to engage peers in-class and through reviewing each other's work
- Require students to start the weekly tasks before the exercises as preparatory work

4. CONCLUSION

In this article we discussed the role of flipped classroom teaching method in university level programming courses. Based on the literature and experiences gained from two of our courses we presented a process flow diagram to illustrate how the flipped classroom teaching can be applied when students are learning programming.

Related research has pointed out several benefits gained by utilizing flipped classroom. Students can learn the theory at their own speed, teachers can concentrate on actual problems instead of repeating the basics in the classroom and no costly lecturing is needed. In the end what matters is that students learning is improved. Our experience supports the existing findings. We have noted that computer science students accepted the flipped classroom method and gave positive feedback on courses. Therefore we can recommend others to try applying the method. It requires initial work and adapting teaching methods, but has been shown to have several benefits.

Lecturing is still the de facto method of teaching at the university level and more research on converting lecture-exercise model to flipped classroom method is needed. As future work our process description should be tested with more courses to verify that it is a sufficient guideline for flipped classroom.

5. REFERENCES

[1] Bishop, J.L. and Verleger, M.A. 2013. The flipped class-room: A survey of the research. *ASEE National Conference Proceedings, Atlanta, GA* (2013).

[2] Carbery, A. and Hegarty, N. 2011. Introducing problem-based learning into one-shot information literacy instruction at Waterford Institute of Technology Libraries. *SCONUL Focus* (Ireland, 2011), 30–33.

[3] Horton, D. and Craig, M. 2015. Drop, Fail, Pass, Continue: Persistence in CS1 and Beyond in Traditional and Inverted Delivery. *Proceedings of the 46th ACM Technical Symposium on Computer Science Education* (New York, NY, USA, 2015), 235–240.

[4] Lage, M.J., Platt, G.J. and Treglia, M. 2000. Inverting the Classroom: A Gateway to Creating an Inclusive Learning Environment. *The Journal of Economic Education*. 31, 1 (Jan. 2000), 30–43.

[5] Lee, U. 1997. Using Internet Tools as an Enhancement of C2 Teaching and Learning. *Foreign Language Annals*. 30, 3 (Oct. 1997), 410–427.

[6] Sperry, R.A. and Tedford, P. 2008. Implementing peer-led team learning in introductory computer science courses. *J. Comput. Sci. Coll.* 23, 6 (2008), 30–35.

[7] Talber, R. 2012. Inverted Classroom. *Colleques*. 9, 1 (2012), 1–2.

[8] Wills, C.E., Finkel, D., Gennert, M.A. and Ward, M.O. 1994. Peer learning in an introductory computer science course. *SIGCSE Bull.* 26, 1 (1994), 309–313.