Teaching master's degree students to read research literature

Experience in a programming languages course 2002–2017

Antti-Juhani Kaijanaho
University of Jyvaskyla
Faculty of Information Technology
University of Jyvaskyla, Finland
antti-juhani.kaijanaho@jyu.fi

ABSTRACT

The skill to read research literature critically belongs in every university graduate's toolbox. I have attempted to teach this skill in a master's degree level course in programming languages over 15 years using, at various times, simulated conferences, voluntary reading exercises, evidence-based practice training, and a flipped classroom with mandatory reading assignments. I discuss my experience and analyze preliminary qualitative data on the use of evidence-based practice and a flipped classroom for this purpose. I present no firm conclusions, but expect that future work (by me or others) will be able to use my experience as a baseline for better teaching of research literature reading.

CCS CONCEPTS

• **Social and professional topics** → *Computer science education*; *Software engineering education*; *Adult education*;

KEYWORDS

critical thinking, critical reading, evidence-based practice, evidence-based programming language design, flipped classroom, science literacy, pyramid discussion, qualitative research, content analysis

ACM Reference Format:

Antti-Juhani Kaijanaho. 2017. Teaching master's degree students to read research literature: Experience in a programming languages course 2002–2017. In *Proceedings of Koli Calling 2017*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3141880.3141893

1 INTRODUCTION

University education has an element of training the student as a scholar and a scientist. This is most obvious in the PhD but it is also present, less prominently, in master's degrees and even in bachelor's degrees. For example, the Finnish government mandates [20] that a bachelor's degree holder should be equipped for scientific thinking and working habits, and a master's degree holder should be equipped for applying scientific methods and knowledge as well as for continuing studies toward a later research degree.

One essential scholarly skill is the ability to locate and critically read the research literature. It is needed even during studies, when preparing the capstone thesis, and I assert that it is one of the essential skills that should separate university trained professionals from trade school graduates in the same field. The current post-truth world especially needs it. In my personal experience, students are often expected to pick this skill up by themselves; even when reading is assigned, students are not given any support nor is success evaluated (beyond possibly a substantive exam). I am not myself innocent of this sin, but I have been working to do better.

In this paper, I discuss my experience in developing a substantive course toward providing explicit support and training for students in reading technical research literature critically. In addition to a report of my experience, I also discuss two explorative case studies based on archived and participant–observer generated data, aiming for preliminary guidance on where this process should go next.

2 BACKGROUND

I begin by discussing the related research and then continue by reviewing two background concepts central to my more recent attempts to teach literature reading: evidence-based practice and flipped classrooms.

2.1 Teaching research paper reading

The research on teaching research paper reading seems to concentrate on undergraduate capstone courses. Dekhane and Price [6], for example, present a capstone software development course that requires students to read a research paper, to find supporting research literature, and to write up a summary. Erkan and Barr [8] discuss their capstone course aiming to integrate the teachings of all previous courses by discussing research papers and making explicit how those papers used what the students have already learned. Neither course appears to provide much scaffolding for students in their reading of papers.

The literature also includes several tutorials for novice readers of research papers (e. g., [18]) and a line of research on the content and teaching of computing research methodology (e. g., [24]). While these provide background and context for research reading, they do not address the key question of teaching the skill.

2.2 Evidence-Based Practice

Evidence-based practice is an interdisciplinary umbrella term covering a number of research dissemination and utilization initiatives, first developed in medicine as Evidence-Based Medicine (EBM) [9]; it originated as a method of teaching medical students and resident

Koli Calling 2017, November 16–19, 2017, Koli, Finland

ACM ISBN 978-1-4503-5301-4/17/11…$15.00
https://doi.org/10.1145/3141880.3141893

Table 1: Summary of course instances

Year	Reg'd	Active	Passed	Pass rate	Literature element	Course format
2002	56	32	32	0.57–1.00	Simulated conference	Lectures and homework
2004	22	16	9	0.41–0.56	Simulated conference	Lectures and homework
2007	32	8	2	0.06–0.25	Simulated conference, voluntary reading assignments	Lectures and homework
2009	28	19	13	0.46–0.68	Voluntary reading assignments	Lectures and homework
2010	30	20	17	0.57–0.85	Voluntary reading assignments	Lectures and homework
2012	35	16	9	0.25–0.56	Voluntary reading assignments	Lectures and homework
2016	32	11	8	0.25–0.73	Voluntary reading assignments	Lectures and homework
2017	27	16	14	0.52–0.88	Required assigned reading	Flipped classroom

physicians to deal with an excess of medical research literature when trying to determine the best way to answer a question about, e. g., the diagnosis or treatment of a particular patient [1]. Central to the method is *critical appraisal*—a heuristic for quickly determining whether a particular research publication answers a particular question in a reliable way [10]. While EBM has since developed a significant institutional arm—including Cochrane reviews [23] and clinical practice guidelines [22]—it is this original practitioner-education model that is of relevance for this work.

Many disciplines have created variations; most notably for my purposes, Evidence-Based Software Engineering (EBSE) is mostly focused on producing systematic reviews [19]. Jørgensen et al. [15] discussed a course design for teaching an early practitioner-education model of EBSE, but more recent reports (e. g., [4]) of teaching EBSE seem to base themselves on teaching the institutions like systematic reviewing.

I recently proposed Evidence-Based Programming Language Design (EB-PLD) modeled after the individual practitioner's view of EBM [16]. It is (quoted from Analysis 35 on p. 163; first-level item numbering elided and punctuation added; emphasis in the original)

> "a decision procedure to resolve uncertainty regarding a particular practical language design problem, applied by an individual language designer, consisting of five steps: (a) formulating a question, (b) locating evidence, (c) appraising the evidence, (d) applying the evidence, and (e) evaluating one's own performance. Here, *evidence* means relevant published studies."

I gave specific advice on each of these steps, including a "[t]entative quality appraisal checklist for comparative questions" (Table 19 on p. 168). As defined, EB-PLD is not a teaching method, but I hypothesize it could function as a scaffolding for students trying to learn to use the research literature: in such a case, the student would take on the role of a language designer in a simulated design exercise with an assigned (or a student-formulated) design problem.

2.3 Flipped classroom

The slogan of the flipped classroom is simple: what was homework is now in-class activity, and what used to be in-class activity is now homework [2, 21]. In practice, this means presenting substantive content in a form the students can access and absorb independently, typically online videos, and using class meetings for active exercises. Lage et al. [21] motivated it by a desire to provide appropriate forms of instruction for students with diverse learning styles. Bergmann and Sams [2] motivated their approach by their desire to accommodate different life situations of their students, many of whom missed classes regularly due to other commitments. More radically, a flipped course design at the university level can be motivated by a desire to enable student self-direction (e. g., [25]).

Systematic reviews in other fields of discipline-based education research [5, 17] suggest a lack of strong evidence base from which to make firm conclusions about the efficacy of flipped classrooms, but there may be a positive effect compared to traditional lecture-based education. A quasi-experiment in biology education [14] suggests that the benefits, if any, of a flipped classroom is properly accounted to the increase in active learning and not to the flipping aspect itself.

Bishop and Verleger [3] claim that a flipped classroom requires "direct computer-based individual instruction" so that, for example, merely assigning required reading to be completed before class would not qualify, but it seems to me that the key aspect (especially in light of the evidence) is the freeing of class time from lecturing.

3 THE COURSE

I discuss in this paper a master's level course on the principles of programming languages first taught in 2002 and taught eight times, most recently in 2017, at the University of Jyväskylä, Finland.[1] It has always had some component encouraging the students to read and decipher highly technical research literature.

A challenge for this course is that the technical literature about its subject matter has adopted an exceedingly formal style for describing and justifying their contributions (for a recent textbook, see [11]). In my experience, developing sufficient theoretical background to discuss modern formally oriented research papers on programming languages requires more than half of the available time in the whole course, and students tend to lose sight of why this is done long before I can start introducing real research papers that use this theory. Complicating the course further in recent years has been my desire to incorporate human-factors research in the course (see, e. g., Chapter 8 of [16]). This requires a different background, including experiment design and statistical analysis.

Counterbalancing these difficulties, this course has always been fairly small, allowing flexibility to the course design. Table 1 shows, among other things, the number of registered students (whether or not they withdrew later), the number of active students (based on the archival record), and the number of students who received

[1] See http://users.jyu.fi/~antkaij/opetus/okp/ for the publicly archived course materials.

Table 2: Summary of simulated conferences

Year	Pages expected	Students participating	Papers submitted	Effect on grade
2002	~10	29	10	major
2004	~10	11	5	minor
2007	5–10	4	4	pass/fail

a passing grade on the course. The number of registered students commonly overestimates the number of actually participating students; conversely, the active count can underestimate it because some activities (like attending lectures in all but the 2017 instance) do not leave any record. The range of pass rates was computed by dividing the number of passed students with either the number of active students or the number of registered students. Differences in the pass rates may, in part, reflect changes in the pass criteria, not necessarily the effectiveness of the teaching techniques.

For most of its history, the course followed a standard format for courses here. The substantive course content was presented in lectures, and homework exercises were set for each week. Once a week, there was an in-class session where the exercises were discussed, with one student called to present their solution and then the teacher commenting on it; these sessions were usually run by an assistant teacher. Generally, exercises were voluntary in this model, but a portion of the course grade was determined by how many exercises the student had reported as completed; the only time a teacher checked a student's work was when they were called to present their solution. At the end of a course, students were typically required to complete a larger capstone project in addition to passing an exam; in this course, the project was the simulated conference (which was later dropped) and the exam was often replaced by learning journals or essays.

4 PAST ATTEMPTS AT TEACHING LITERATURE READING IN THIS COURSE

I first discuss two attempted techniques for teaching literature reading on the course, which I no longer use. The data for this section are purely archival, supplemented by my memory; I generated no research-specific data about these techniques at the time.

4.1 Simulated conference (2002–2007)

The first three times, a mandatory part of the course was a simulated conference, with students writing (in teams) papers discussing some technical topic in the subject area of the course, presenting their paper to an audience consisting mostly of other students in the course, and each team serving as an opponent to one other team. Table 2 summarizes the three instances of the simulated conference.

In 2002 and 2004, students were instructed to prepare a paper discussing a specific programming language, the appearance of a specific feature in several languages, some specific aspect of the theory of programming languages, or a specific topic in the history of programming languages. The paper was required to be predominantly based on research papers, supplemented by the technical documentation of the languages under discussion. Both years, a proceedings volume was prepared, and a multi-session single-track simulated conference was held. In 2007, the simulated conference presented difficulties. Initially, the instructions were the same as before, but because so few people participated actively in the course, works were to be prepared alone. No proceedings volume was prepared, and there are very few archived records about the actual conference.

There is a very limited amount of data available at this time to make meaningful assessment of the simulated conference. Clearly, the 2002 instance was very successful; reading the proceedings volume 15 years later fills me with a sense of awe at the skill and level of achievement of the students who took part. To some extent, a novelty effect must have been present, as it was the first such course at this department, and the simulated conference was also a new idea here. In contrast, the 2007 conference was a clear failure: about half of the students who had been active during the lecture part of the course failed to even specify a topic. At this point, I concluded that the simulated conference model was no longer viable, and I dropped it from the course concept.

4.2 Voluntary reading assignments (2007–2016)

In 2007, I added new exercises that called the students to read a specific research paper and to prepare a 10 minute presentation about it. In the exercise session, one of the students was called by the session teacher to present, and the teacher then led discussion about the article. Oral feedback from students and from the assistant teacher suggested that this was a bit too demanding a task, and so mid-course in 2009, the reading assignment was changed, first to just ask the student to prepare to discuss, and a couple of weeks later, to specifying explicit prompts.

For example, I assigned the classic TOPLAS paper on Featherweight Java [13], or its earlier OOPSLA version, each year near the end of the course. At first, the assignment specified nothing more. In 2010, I prompted the students to think about the purpose of the formalism discussed, and to figure out the meaning of two particular formal rules. In 2012 and 2016, I gave the following prompts (translated here to English): "a. Why do the authors think Featherweight Java is useful? b. What is *stupid cast*, how does FJ differ from Java regarding it, and why? c. Are there any surprises in the formal rules of FJ?" A number of other, mostly classic but also some contemporary papers were also assigned at various times.

I have very little data on which to base any evaluation of this practice, but my memory suggests that it did not encourage the students to go beyond the trivial and the surface.

5 RECENT ATTEMPTS

In this section I detail two recent attempts to teach research literature reading. In both cases, some systematic data generation occurred. The data are qualitative, in the form of text written by the participants themselves. Beyond the generated data, the discussion here is based on my records and memory. The data analysis, conducted as a conventional content analysis [12], was guided by the following questions:

(1) What aspects of the studied intervention helped or hindered students learning to read research literature?
(2) What subjective difficulties can be observed in the students' learning to read research literature?

(3) What issues in course design emerge from the data?

Limited space precludes a full exposition of the analysis here.

5.1 The EB-PLD trial (2016)

In the 2016 course, as part of the weekly voluntary exercises in the final three weeks of the course, students were asked to complete the evidence-based programming language design (EB-PLD) process [16]. The main difference to my original EB-PLD was that no language design context was present: instead, the process was used to answer questions regarding programming languages in a generic context.

Data generated from this trial comprise the answers submitted by the students as well as my participant-observer notes as the teacher, written shortly after each session where trial exercises were discussed. Participation was voluntary, and all participants gave informed consent.

Four students submitted answers to the trial exercises. They reported mainly difficulties in formulating a useful question:[2]

> [Student:] Especially defining the scope of the question affects results significantly. Too specific a question risks that no fitting answer can be found by the searches and "almost good" results are excluded.

The process was also seen as time-consuming, but worth the effort. One student noted that the lack of statistical training made it hard for them to critically appraise the human-factors experiments that they were reading.

I noted that an exercise session of the sort used in this course seems to induce a power hierarchy:

> [Teacher:] Troughout the session, students mainly responded to prompts from me, and did not speak on their own initiative. This shows, I think, an unfortunate deference to me as the teacher [. . .]. The session did not seem much different from other similar sessions I have ran on this course and in other courses.

My impression from the data and their analysis is that those students who did take part found the actual reading instructive but time-consuming. The teaching method—lectures and exercises—seems to be a hindrance, however.

5.2 Flipped classroom with assigned reading (2017)

In the 2017 course instance, I adopted a flipped classroom model, replacing lectures and homework. The students were expected to study certain assigned readings before each in-person course meeting and to be prepared to discuss the material in class; video lectures were not used. Each course meeting was generally structured as peer discussion, based on teacher-assigned prompts, following usually a pyramid model (see, e. g., [7]): students are first assigned to work in pairs or in groups of three;[3] after a while, groups are merged to form 2–3 larger groups; and the exercise is ended by a plenary discussion. One round of discussion from pairs to the plenary can easily take 45–90 minutes, depending on the topic. The role of the teacher is to set the assigned reading and the discussion prompts, to direct group formation, to circulate among the student groups listening to their discussions, and to take action in the case of problems in the discussion process.

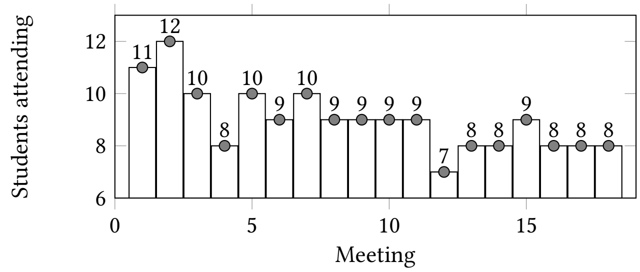

Figure 1: Attendance in flipped classroom meetings (2017)

The in-class discussions had deliberately low stakes. Attendance, or completing a corresponding distance learning exercise, was required for each course meeting, but the in-class discussions did not affect the passing grade; only the attendance of each student was recorded (see Figure 1). The goal was to enable students to throw themselves into the discussion without a fear of failure; my observation is that this seems to have succeeded. The passing grade was determined by other assignments.

My original intent was to use published research literature as the assigned readings as much as possible, but it quickly became obvious that the materials were not suitable for introducing the more technical material. Instead, I wrote a series of notes which I then assigned as readings when the highly technical topics were under discussion. Similarly, while the pyramid model was often used in the more technical meetings as well, exercises were used instead of discussion prompts, and sometimes the pyramid progressed straight from the pair stage to the plenary. In some cases, I decided to lecture for 15–30 minutes at the end of a meeting, after it became clear that the students did not grasp the essential points from the readings.

At the end of the course, I assinged the students a voluntary (but compensated) written task, where the students were asked to provide (non-anonymous) feedback to me as the teacher, guided by open-ended questions. Four students submitted answers to this task; all consented to their answers being used as data and none objected to discussion in a paper.

Several students commented how what they learned on this course was not what they had expected:

> [Student 1:] I feel my mind has kind of broadened but not in the way I expected.

Several also commented on how reading actual papers was good but sometimes too hard; the course was considered hard work, or perhaps even too much work at times. Some students would have wanted more lectures or a textbook to follow alongside. The discussions were seen as a good feature of the course, except that sometimes they felt that other students had not prepared themselves sufficiently. The course's focus on critical reading was seen as a good, perhaps even unique, feature:

[2] I have translated some of the quoted student answers from Finnish.

[3] While pairs are desirable, often the number of students present is not even.

> [Student 2:] I further learned that many things that are presented as givens regarding languages [...] are mainly based on various anecdotes and not scientific research.
> [Student 3:] it was interesting to read about the experimental side of software engineering research [...] reading about a lot of the experiments left me feeling a bit uncertain about the actual quality of their designs.
> [Student 4:] I suspect that this course at least a little helps me develop my critical thinking. Scientific articles are critically assessed quite seldom at least in this level at the university.

I recall one student commenting in the final meeting (not recorded formally, so this is a fallible memory) that they knew less at the end of the course than in the beginning.

It seems clear to me, in light of my experience and the data, that the flipped classroom model works well in teaching critical reading but is challenging with highly technical material.

6 DISCUSSION

I believe that university students, even at the bachelor's and master's degree phases, should be given sufficient tools to follow the research literature of their specialty. In this paper I have described my attempts to make this belief a reality in one master's level course. Because what I describe is past lived experience and not controlled experimentation, I do not claim to have any good answers. What I do have, however, is a bundle of observations that may or may not be transferable to other courses and universities.

First, while throwing our students to the water and expecting them to swim can sometimes work (witness the excellent simulated conference experiences in 2002 and 2004), it is not a reliable method (witness the dismal simulated conference in 2007). Further, like in all teaching, scaffolding is essential. Third, teaching students to critically appraise articles, either using something like the evidence-based practice model or having students read and discuss with scaffolding and encouragement, can lead to pleasant surprises to the students.

It is a bit unfortunate that so few students participated in the EB-PLD trial or submitted feedback when asked; though that might be a reflection on the interventions at issue, it is in line with my larger experience—most of our students do not volunteer their opinions and do not provide feedback to us even when asked.

I welcome attempts to test these ideas rigorously, but I have a hard time coming up with good experiment designs: for example, how does one measure a change in critical reading skills? Perhaps one could do a cohort followup several years later, and see how they did in their theses; but the problem will then be how to concoct a valid control.

In any case, the problem of teaching critical reading remains. I hope my experiences can inspire others in the community to do better than I did.

ACKNOWLEDGMENTS

Thanks to Ville Tirronen for help and Tommi Kärkkäinen for encouragement. The anonymous reviewers gave useful feedback.

REFERENCES

[1] Eric G. R. Barends and Rob. B. Briner. 2014. Teaching Evidence-Based Practice: Lessons from the Pioneers. *Academy of Management Learning & Education* 13, 3 (2014), 476–483. https://doi.org/10.5465/amle.2014.0136

[2] Jonathan Bergmann and Aaron Sams. 2012. *Flip Your Classroom: Reach Every Student in Every Class Every Day.* ISTE & ASCD, Eugene, OR.

[3] Jacob Lowell Bishop and Matthew A. Verleger. 2013. The Flipped Classroom: A Survey of the Research. In *120th ASEE Annual Conference & Exposition.* ASEE, Washington, DC, Article 6219, 17 pages. https://www.asee.org/public/conferences/20/papers/6219/view

[4] Daniela Castellucca and Giuseppe Visaggio. 2013. Teaching Evidence-Based Software Engineering: Learning by a Collaborative Mapping Study of Open Source Software. *ACM SIGSOFT Software Engineering Notes* 38, 6 (2013), 1–4. https://doi.org/10.1145/2532780.2532803

[5] Fei Chen, Angela M. Lui, and Susan M. Martinelli. 2017. A systematic review of the effectiveness of flipped classrooms in medical education. *Medical Education* 51, 6 (2017), 585–597. https://doi.org/10.1111/medu.13272

[6] Sonal Dekhane and Richard Price. 2014. Course-Embedded Research in Software Development Courses. In *SIGCSE 2014.* ACM, New York, 45–48. https://doi.org/10.1145/2538862.2538927

[7] Arthur K. Ellis. 2010. *Teaching, Learning, & Assessment Together: Reflective Assessments for Elementary Classrooms.* Routledge, London.

[8] Ali Erkan and John Barr. 2016. Algorithms + Organization = Systems. In *ITiCSE 2016.* ACM, New York, 65–70. https://doi.org/10.1145/2899415.2899458

[9] Trisha Greenhalgh. 2014. *How to Read a Paper: The Basics of Evidence-Based Medicine* (5 ed.). Wiley, Chichester.

[10] Gordon Guyatt et al. 1992. Evidence-Based Medicine: A New Approach to Teaching the Practice of Medicine. *JAMA* 268, 17 (1992), 2420–2425. https://doi.org/10.1001/jama.1992.03490170092032

[11] Robert Harper. 2013. *Practical Foundations for Programming Languages.* Cambridge University Press, Cambridge.

[12] Hsiu-Fang Hsieh and Sarah E. Shannon. 2005. Three Approaches to Qualitative Content Analysis. *Qualitative Health Research* 15, 9 (2005), 1277–1288. https://doi.org/10.1177/1049732305276687

[13] Atsushi Igarashi, Benjamin C. Pierce, and Philip Wadler. 2001. Featherweight Java—A Minimal Core Calculus for Java and GJ. *ACM Transactions on Programming Languages and Systems* 23, 3 (2001), 396–450. https://doi.org/10.1145/503502.503505

[14] Jamie L. Jensen, Tyler A. Kummer, and Patricia D. d. M. Godoy. 2015. Improvements from a Flipped Classroom May Simply Be the Fruits of Active Learning. *CBE—Life Sciences Education* 14, 1, Article 5 (2015), 12 pages. https://doi.org/10.1187/cbe.14-08-0129

[15] Magne Jørgensen, Tore Dybå, and Barbara Kitchenham. 2005. Teaching Evidence-Based Software Engineering to University Students. In *METRICS 2005.* IEEE, Washington, DC, 24–31. https://doi.org/10.1109/METRICS.2005.46

[16] Antti-Juhani Kaijanaho. 2015. *Evidence-Based Programming Language Design—A Philosophical and Methodological Exploration.* University of Jyväskylä, Jyväskylä. http://urn.fi/URN:ISBN:978-951-39-6388-0 PhD diss.

[17] Aliye Karabulut-Ilgu, Nadia Jaramillo Cherrez, and Charles T. Jahren. 2017. A systematic review of research on the flipped learning method in engineering education. *British Journal of Educational Technology* early view (2017), 1–14. https://doi.org/10.1111/bjet.12548

[18] S. Keshav. 2007. How to Read a Paper. *ACM SIGCOMM Computer Communication Review* 37, 3 (2007), 83–84. https://doi.org/10.1145/1273445.1273458

[19] Barbara Ann Kitchenham, David Budgen, and Pearl Brereton. 2016. *Evidence-Based Software Engineering and Systematic Reviews.* CRC, Boca Raton, FL.

[20] Krista Kiuru and Laura Hansén. 2013. Valtioneuvoston asetus yliopistojen tutkinnoista annetun valtioneuvoston asetuksen muuttamisesta. *Suomen säädöskokoelma* 1039 (2013), 1–14. http://www.finlex.fi/fi/laki/kokoelma/2013/sk20131039.pdf

[21] Maureen J. Lage, Glenn J. Platt, and Michael Treglia. 2000. Inverting the Classroom: A Gateway to Creating and Inclusive Learning Environment. *Journal of Economic Education* 31, 1 (2000), 30–43. https://doi.org/10.1080/00220480009596759

[22] Marjukka Mäkelä and Ilkka Kunnamo. 2001. Implementing evidence in Finnish primary care: Use of electronic guidelines in daily practice. *Scandinavian Journal of Primary Health Care* 19, 4 (2001), 214–217. https://doi.org/10.1080/02813430152706701

[23] Mark Starr et al. 2009. The origins, evolution, and future of The Cochrane Database of Systematic Reviews. *International Journal of Technology Assessment in Health Care* 25, s1 (2009), 182–195. https://doi.org/10.1017/S026646230909062X

[24] Matti Tedre, Danny Brash, Sirkku Männikkö-Barbutiu, and Johannes Cronjé. 2014. Towards Identification and Classification of Core and Threshold Concepts in Methodology Education in Computing. In *ITiCSE 2014.* ACM, New York, 237–242. https://doi.org/10.1145/2591708.2591758

[25] Ville Tirronen and Ville Isomöttönen. 2011. Making Teaching of Programming Learning-oriented and Learner-directed. In *Koli Calling 2011.* ACM, New York, 60–65. https://doi.org/10.1145/2094131.2094143