Teaching master’s degree students to read research literature
Experience in a programming languages course 2002–2017

Antti-Juhani Kaijanaho
University of Jyväskyla
Faculty of Information Technology
University of Jyväskyla, 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

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
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 "tentative 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

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

### Table 1: Summary of course instances

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

<table>
<thead>
<tr>
<th>Year</th>
<th>Pages expected</th>
<th>Students participating</th>
<th>Papers submitted</th>
<th>Effect on grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>2002</td>
<td>~10</td>
<td>29</td>
<td>10</td>
<td>major</td>
</tr>
<tr>
<td>2004</td>
<td>~10</td>
<td>11</td>
<td>5</td>
<td>minor</td>
</tr>
<tr>
<td>2007</td>
<td>5–10</td>
<td>4</td>
<td>4</td>
<td>pass/fail</td>
</tr>
</tbody>
</table>
(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) pro-
cess [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
countext.
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 ques-
tion affects results significantly. Too specific a ques-
tion 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:] Throughout the session, students mainly
responded to prompts from me, and did not speak
on their own initiative. This shows, I think, an un-
fortunate deference to me as the teacher […]. The
session did not seem much different from other sim-
ilar 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 meet-
ing 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 usu-
ally 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

2I have translated some of the quoted student answers from Finnish.
3While pairs are desirable, often the number of students present is not even.
[Student 2:] I further learned that many things that are presented as given 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 to develop my critical thinking. Scientific articles are critically assessed quite seldom at least in this level at the beginning.

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-PDL 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 concot 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