Teaching reflection

Hjálmtýr Hafsteinsson

This is a teaching reflection for Hjálmtýr Hafsteinsson, associate professor of Computer Science, University of Iceland. Teaching CV is in a separate document. Appendix with evidence and documentation is in a separate document.

Contents

Teaching reflection word count: 6932

Hjálmtýr Hafsteinsson - hh@hi.is

Department of Computer Science
School of Engineering and Natural Sciences
University of Iceland
107 Reykjavík
tel.: +354-525-4932

Teaching principles

Teaching is one of the most enjoyable things that I do. It is a wonderful feeling when you realize that a student is understanding something for the first time or is seeing a part of the world in a new light. This does not always work out, but when it does you sense it immediately and the feeling becomes quite addictive! It is probably the thing that inspires me the most in my teaching.

It is very important to me to have a relationship with the students that is built on mutual trust. I work towards that goal in several ways. The students have, almost without exception, come to the University to study a subject that they are interested in, or are at least curious about. I view it as my role to nurture that interest and show them how much fun the study of Computer Science can be. I try not to let undesirable behaviors (copying solutions, not show up for labs, etc.) of a handful of students ruin the enjoyment of the dedicated ones. So instead of having strict rules, I try to structure my courses such that certain behavior is expected and encouraged. I also seek to educate the students in how to deal with study difficulties so that they will be better equipped to handle them (see Case 2). Of course, some amount of framework is needed, but the rules cannot become the focal point in the course. By making the subject exciting and compelling to study the issue is usually automatically resolved. You can say that I tend to favor using the "carrot" rather than the "stick".

I believe that the most effective and enjoyable way to learn is by working on interesting and relevant projects at the proper difficulty level. This means active learning (Bonwell & Eison, 1991), which involves applying the ideas and methods ourselves. Only that way do they make sense to us. Nobody else can put that understanding into our heads. Thus, comprehension must be developed by each student individually. My task as a teacher is to make that process productive and rewarding. One way to do that (see Case 3) is to create projects that motivate the students to work on them. The effort involved in designing an interesting and an appropriate project pays substantial dividends in the amount of learning that the students gain from it. Another way (related to Case 1), is to have a judicious progression in the activities that result in the intended learning. For example, start with a simple retrieval practice, where you only need to recollect the new concept and apply it with little change, then you need to use it in a slightly more challenging situation, and finally in an interesting and rewarding project when you fully grasp the potential of the concept. There are many other possible ways to learn effectively, and I am always open to new ideas and willing to experiment with ways that contribute to this end.

Teaching biography

I did my Ph.D. studies in Computer Science (CS) at Cornell University in 1984-88. My first experience with teaching at University level came right after that in the CS department at the University of Iceland (UI). In fact, I had my Ph.D. defense in early September 1988 at Cornell and was teaching an undergraduate course on Algorithm Analysis one week later at the UI. I was literally thrown in at the deep end!

Over the years I have taught many courses in the CS department at the UI, both at undergraduate and graduate levels (see Teaching CV). Most of them have been on theoretical Computer Science or on computer organization. In 1995 I designed a new elective course on Computer Graphics. This semester I am teaching that course for the 20th time, but obviously the content has changed drastically over the years. Course material tends to change very rapidly in CS courses. In the last 10

years I have also become more involved in the first-year courses, especially the first programming courses, CS1 and CS1a.

The University Teaching Award is given to one teacher in the entire University per year. I received it in the year 2000 (see Appendix A01). That was only the second time it was handed out, but it has been awarded every year since then. I have taken two courses on university pedagogy from the Centre of Teaching and Learning (CTL) in their diploma program for university teachers. I took one course in 2012 and the other one in 2019. Both courses had a big influence on my perspective on teaching. Earlier in my teaching career I mostly modelled my teaching methods on one or two excellent teachers that I had as a student, but after taking the CTL courses I started focusing more on the students as active and independent learners and how I can facilitate that (Kugel, 1993). This can be compared to a successful football coach who has no knowledge of sport physiology. They can reach a certain level with enthusiasm and passion, but their lack of knowledge will always limit them. Similarly, if you want your students to reach their full potential you need to know more about pedagogy, cognitive psychology, and student learning.

In addition to the two courses from the diploma program, I have participated in many other events held by the CTL, both formal and informal. For instance, the monthly "Teaching Coffee Hour" has been a great venue to discuss teaching with people from other departments in a casual setting.

From 2012 to 2015 I was the head of the Teaching Committee for the School of Engineering and Natural Sciences (SENS). That also involved being a member of the University Teaching Committee. This was a very valuable experience and opened my eyes to the many different approaches to teaching in different disciplines.

Case 1: Lecture quizzes in large courses

Background:

The Computer Science department at the UI has two first semester programming courses: CS1 (for CS, math, and electrical engineering students) and CS1a (for other engineering students). Each course has between 150 and 300 students. During the fall semester of 2005 I taught CS1 together with another teacher, while the regular teacher was on a sabbatical leave. As we were only supposed to teach it this one time, we did not want to change much in the organization of the course. It was taught in the traditional way with 2x40min. lectures twice a week and weekly homework. The lectures were a slideshow with slides directly from the textbook. There were around 200 students in the course. This was the first time I had taught so many students in one class. The course went OK, but I found the teaching dull and felt something was lacking. In fact, I felt that I had let the students down and I could have done better.

The next time I taught one of those large classes was in the fall semester of 2012, when I was asked to take over CS1a, the programming course for engineering students. In the spring of 2012, I took a course on University Pedagogy run by the CTL, as mentioned in the Teaching biography above. I knew I would be teaching this large course in the fall and I was determined to a better job than last time. Therefore, I was particularly receptive to new ideas on teaching. The main goal that I decided to achieve was to make the students more active participants in the lectures. There were two concrete items that I implemented during the fall of 2012. One of them was the lecture quizzes that are the subject of this case and are described below. The other one was to make the lectures as "alive" as possible, so that most of the lecture time is spent looking at pieces of code, tracing

through them, changing them and seeing what happens. The lecture slides are the outline of the lectures, but we regularly jump over to the programming environment (Matlab at the time) to see how the sample programs really work. The students get these sample programs before the lecture and are encouraged to follow along and trace through them at the same time on their own computers. They get used to working with code in the programming environment and get a better feel for how the programs behave.

This case will consider lecture quizzes as a way to achieve active learning objectives in CS1/CS1a. It is well known that students learn more when they are actively engaged, rather than when they are passive recipients of knowledge (Bonwell & Eison, 1991) (Deslauriers, Schelew, & Wieman, 2011). In the lecture quizzes the students do simple exercises from the subject immediately after it is presented, thus making it stick better in their memory (Roediger & Karpicke, 2006). Some of the exercises require reflection, thus involving thinking at a higher level in the Bloom's taxonomy (Krathwohl, 2002).

Implementation:

Lectures at UI are usually 2x40 minutes with 10-minute breaks in between. They are twice weekly for 14 weeks. I allocated a 10-minute slot at the end of each 40-minute session for a quiz from the material presented in the first 30 minutes. Each quiz would have 3 exercises, so that each 80-minute lecture would contain 6 exercises for 20 minutes. The students would fill out answer sheets on paper or online (Google Forms or Canvas). Sample solutions of the quiz were posted soon after the lecture. The student solutions were not graded. I only registered whether a student had handed in a solution. To encourage students to do the quizzes (and show up for lectures!) we gave points for handing in solutions. They counted for up to 10% of the final grade. It was enough to do around 75% of the quizzes to get full marks. This quiz grade would only raise the final grade, so those who did not want to show up for lectures were not penalized. There is a strong tradition at the University of Iceland that lecture attendance is not mandatory, so by only using the quizzes to raise the grade we are not forcing the students to attend lectures, only encouraging them.

Result:

The results were very good from the beginning. They can be split into a few parts:

- I tried to have the exercises easy, but still from the key parts of the material. During the quizzes I often got questions from the students on topics that I thought I had explained quite well in the lecture. Thus, the quizzes served to reduce misunderstanding that could have made trouble for the students later on.
- Students were encouraged to work on the quizzes themselves but were also allowed to talk to their neighbors. As the solutions were not graded there was less pressure on them to get "the correct" answer from their neighbors. I kept reminding the students that they benefited much more from handing in a wrong answer, if it was their own, rather than a correct answer that they got from someone else.
- Lecture attendance improved quite a bit from previous versions of the courses. The 10% grade helped, but many students told me in person and in the teaching evaluations that they felt they now got more out of attending the lectures, rather than just reading the slides at home. For example, in 2016 (see Appendix A10): "*The lecture quizzes are a very good incentive for one to attend the lectures, which is great*"[1].

[1] Translated from "*Fyrirlestraæfingarnar eru mjög góð hvatning fyrir mann að mæta í tíma sem er frábært.*"

- In order to register which students had handed in solutions, I had to scan through all their solutions. A side benefit of that is that I often saw if the exercises were too difficult (many empty or wrong answers) or if many seemed to misunderstand the material. This gave me a chance to revisit a topic that many students did not understand. It has also helped with the development of the exercises for the future.
- In the teaching evaluations (see Appendix A11, A12) many students said that they focus better during the lecture part of the class, since they know that there will be a quiz at the end where they want to do well. Examples: from 2012: "*I have found the lecture quizzes very rewarding, concentration is maintained the whole time and by doing exercises immediately I remember the material better.*"[2]. Some students also mentioned that they found it easier to do the weekly homework as the material was fresher in their minds. For example, a comment from 2014: "*It is a good arrangement to have questions in the lecture, because then you remember the material much better and are more likely to be able use it to solve problems later."*[3].

Development:

The lecture quizzes quickly became very popular with the students and they were one of the most frequently mentioned items in the positive section of the teaching evaluations. Actually, students in the first year started mentioning them to other teachers that were teaching them, and the teachers then talked to me to ask what it was that the students were asking for! It gave me a good opportunity to talk pedagogics with fellow teachers. That can sometimes be tricky to do without seeming to be butting in on their courses. Many teachers in the CS department have now taken up some form of lecture quizzes in their courses, so most CS courses now have quizzes during the lectures. I have given presentations on this idea on several different occasions, for example at a teaching symposium in the School of Humanities at UI (see slides in Appendix A13) and a university-wide workshop for PhD students (see slides in Appendix A14).

I have used different versions of lecture quizzes in all my courses since the fall of 2012. I taught CS1a in 2012-14. Then I moved to CS1 and taught that in 2015-18. During that time, the structure has mostly been the same, but some development has taken place. For example, I have occasionally put in exercises from previous material from the course. This is in the spirit of spaced repetition (Dunlosky, Rawson, Marsh, Nathan, & Willingham, 2013) and should strengthen the learning of important ideas.

In the fall of 2020, I taught a part of the second-year course Information Engineering for industrial engineering students. There I tried having the quizzes in the middle of the lectures, immediately after I had introduced some topic. The students got 8-10 minutes to do the quiz and then I would spend 5-7 minutes discussing the solutions. There were usually two or three quiz sessions per 80-minute lecture, so that the quizzes took between a third and up to a half of the total lecture time. The students were very happy with this version (see Appendix A15). One comment was "*These little exercises that we did during the lectures made a big difference and made me learn the material really well.*"[4].

[2] Translated from "*Mér hefur þótt fyrirlestraræfingarnar skila miklu, einbeitingin helst þá allan tímann og með því að vinna strax verkefni úr glærunum festist efnið betur í minni."*

[3] Translated from "*Gott fyrirkomulag að hafa spurningar í fyrirlestrunum því þá man maður mun betur eftir efninu sem var fjallað um og er líklegri til að geta notað það til að leysa verkefni síðar."*

[4] Translated from "*Gerði svo mikið fyrir mann að hafa þessar litlu æfingar í fyrirlestrunum sem maður gerði jafnóðum meðfram kennslunni og lærði þetta því svo vel*"

One variation on the lecture quizzes that I tried recently in Analysis of Algorithms did not work out. It consisted of adding an exercise at the beginning of the lecture from the material of the upcoming lecture. The students got 5 minutes to work on the problem and then it would be resolved during the lecture. The idea, which is based on the Pre-testing Effect (Richland, Kornell, & Kao, 2009) was that the students would be more receptive to a topic after having tried to work it out for themselves. I explained the reasoning behind this experiment to the students and it started off well, but then as the semester wore on, more and more students started arriving 5 minutes late to the lectures. Towards the end, only a handful of students were seriously trying to solve the problem. Perhaps the exercises were too hard, and the students did not see an obvious benefit from them, as they could from the usual lecture quizzes, which were still at the end of the lectures.

I will continue to develop the format of the lecture quizzes in the future. Different courses benefit from different types of quizzes and you must adjust the setup accordingly. I would like to experiment with allowing students to work in groups on some of the quizzes. It would also be interesting to investigate the timing of feedback. Is it better to provide solutions immediately after each exercise, after the whole quiz, or even later? Further, what kind of feedback is appropriate?

Case 2: Teaching students more effective study habits

Background:

When I started teaching the large first year courses, CS1 and CS1a in 2012 it really struck me how many students were not using good study methods. Some of their methods would have been enough for bright students in high school, but when the students started university the study methods were completely inadequate. For instance, many students counted on being able to cram for the exam. They would do very little during the semester, but in the last 1-2 weeks they would study all day. Another common mistake was spending most of the time reading the study material or watching YouTube videos of similar material, but avoiding work on exercises or projects. Computer Science studies, especially these early programming courses, have a high element of repeated training, like learning to play a musical instrument. This sort of skill is just not acquired by reading a book or watching a video.

I got the idea of trying to educate the students in some aspects of pedagogy, in order for them to see which study methods worked and which ones were not as good. The main point was to get the students to use effective study methods by explaining their pedagogical background. I suspect that the students have never been adequately taught good study methods, at least not by analyzing why some methods are better than others. Many students also have wrong ideas about the effectiveness of active study techniques, thinking them inferior when in fact they are learning more (Deslauriers, McCarty, Miller, Callaghan, & Kestin, 2019).

The same thing holds for students as for teachers: when they start reflecting on the purpose of individual aspects of learning, they become more effective and "professional" students, more likely to use their time productively. It is especially important to persuade the first-year students to use effective study methods, as this will shape the rest of their academic career. They will be less likely to drop out and will be more successful learners.

Implementation:

Initially I just added simple material on pedagogy and effective study methods to the first lecture (see slides in Appendix A20). This was mostly on the value of active learning and review. I also

connected these items to the structure of the course, so that the students would see the purpose of each element of the class. For example, the role of the lecture quizzes is to recall and reinforce the subject of the lecture. The lab exercises in the study sections (3-5 days after lecture) are simple drills, so they are both retrieval practice and active learning. The weekly homework (6-10 days after lecture) aims to give more practice and understanding of the material. In the fourth week of the course, I would revisit this pedagogic information for 5-10 minutes, just to remind the students of the purpose of the course elements.

It is well known that students pay more attention to the opinions of their peers than that of the teacher. Thus, in the last lecture of the course, I asked the students to give some good advice to the group of students who will be taking the course the next year. This was very successful, and I would get 100-200 comments each year. I would then select the most interesting tips and put one on the title page of each lecture (see first slide in A20). The name of the student giving the advice was included, but the students could ask for their advice to be anonymous.

When I started looking at the grade distribution based on the number of homework assignments and number of lecture quizzes it was obvious how important it is to work steadily through the semester. There are usually some rumors going around among the first-year students that some student never shows up for lectures and does no homework but gets 10 in every course. By showing the students the grade distribution based on handing in 3/4 af the assignments you can debunk this rumor. At least this student was not taking CS1 last year, since no student handing in fewer than 3/4 of the homework assignments got a 10! Showing this grade distribution is a very effective way to convince the students what works and what does not work ("Which distribution do you want to be a part of in December?!"). I always show them the graphs for last year in the first lecture and discuss them. This is especially important in CS1 and CS1a, since the students there are just starting their university studies and might not realize what is required to succeed.

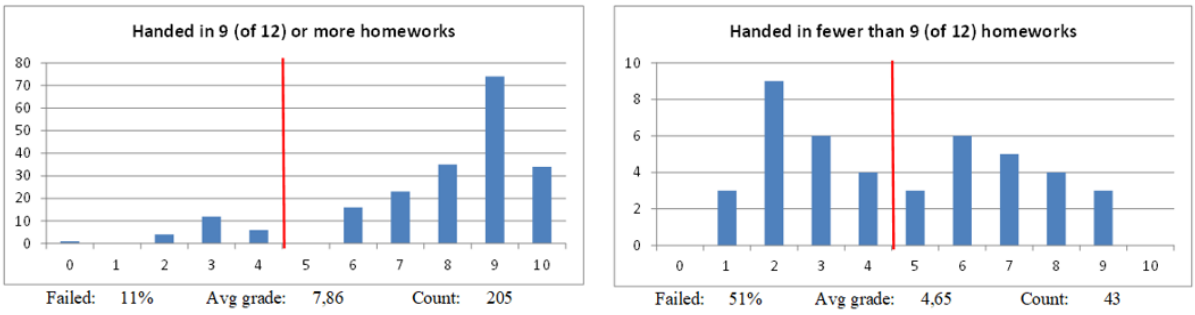

Figure 1. Grade distribution by number of submitted homeworks in CS1, fall 2017

Result:

The teaching evaluations have shown that the students appreciate that fact that there is some pedagogical thought behind the course structure. One comment on the midterm evaluation in CS1 in 2015: "*Very good when you find that the teacher cares about how the study material is delivered in the best possible way.*"[5]. Another comment from the second-year course Computer Systems, fall 2020 (see Appendix A21): "*I also really appreciate how he presents study methods for students and*

[5] Translated from "*Mjög gott þegar maður finnur að kennarinn pælir i þvi hvernig namsefnið kemst sem best til skila.*"

thinks about how best to teach the subject"[6]. The students are more satisfied with the implementation of the course when they understand the purpose of individual elements.

It is also evident that more students are handing in more homework assignments than before. In CS1 the proportion of those who hand in 3/4 of all homework (9 out of 12) has been going up. In 2015 there were around 68% of the students that handed in 9 homework assignments or more (158 out of 232). In 2016 the ratio had gone up to 78% (186 out of 238) and in 2017 there were 82% (205 out of 248) that handed in 9 assignments or more. The 2018 class was the largest one ever, but still 80% (247 out of 310) handed in 9 or more assignments. The grade distribution in this group of students has not altered significantly, so that means that there are more students getting good grades because of good study habits.

It has worked well to have study tips from former students on the lecture slides. The students really try to give useful advice to the next group of students. It is also motivating that they might get their advice on a lecture slide of the course and become "famous" in the course! The tips are often ones that I would never have thought of myself. Below are some examples:

- *If Donald Trump can become President of the United States, then you should be able to pass CS1.*
  Benedikt, 2016
- *Don't use* **`stackoverflow.com`**[7]*, it is useless.*
  Anonymous, 2017
- *Open the homework as soon as it arrives and start thinking about it. It often takes time to digest.*
  Kristín Fjóla, 2018

Development:

It was not easy to decide which pedagogical elements should be presented to the students. I became enthusiastic about many things after taking the Teaching studies course at the Centre for Teaching and Learning (CTL). However, I thought it best to concentrate on a few concrete practical elements, with some reference to the theory, rather than cover too many different options. I was only utilizing a fraction of the pedagogical techniques that I had learned in the teaching of CS1. Therefore, I concentrated on explaining and justifying the elements that I was using in the course.

When I have taught more advanced courses, I have also started adding these pedagogical elements, but there I feel justified in going slightly deeper into the theory. For example, in the course Analysis of Algorithms in the second year I have one or two problems on the first homework where the students must read a research article on a particular pedagogical matter (see homework in Appendix A22). I have then used that as an opportunity to discuss those issues. This material is of course entirely outside the usual syllabus of the course, but I feel justified in devoting a small amount of time to give the students an opportunity to reflect on how they conduct their studies and perhaps to develop themselves as learners. After all, we are in a university environment and all of us, teachers and students, should use scientifically sound methods for teaching and learning. It might even be considered a part of our role as teachers to guide the students on scientific methods in their studies.

---

[6] Translated from "*Kann einnig vel að meta hvernig hann tekur fyrir lærdómsaðferðir fyrir nemendur og hugsar um hvernig er best að kenna efnið.*"

[7] **`stackoverflow.com`** is a question and answer site for programmers. CS students often use it to look for, or ask for help with homework problems. The answers they get are of mixed quality.

In the future I would like to explore additional study techniques. This includes methods such as more structured distributed practice, reflective learning, and increased use of peer assessment and review.

Case 3: Creating interesting and relevant projects

Background:

When students are learning new material, it is important that they get a chance to apply it early in interesting and useful projects. Students are much more motivated when they are working on something that they like. If the project is appealing to the students, then the work on it is not a chore, but a fun challenge where they can lose themselves in finding a solution or creating an attractive result.

Most courses in the School of Engineering and Natural Sciences (SENS) have weekly homework. Their purpose is to train the students in the material and strengthen the learning. It can be compared with the drills that music students do. In math courses students need to be proficient in certain basic technical skills so they can apply them subconsciously to more complicated tasks. Similarly, in programming courses students will need to become so skilled in writing code that they can quickly put together a program to test a hypothesis or simulate some phenomenon.

As useful as the drills are, they are not very motivating for the students. They have difficulty in seeing their purpose and then it just becomes practice for its own sake. This can be compared to an athlete who is only allowed to train but can never compete in games or tournaments.

Implementation:

In most of the courses I teach I use the weekly homework to give the students the technical practice that they need, but also add two or three larger projects. Those projects are designed such that they *i*) build on the material that has been covered and *ii*) are interesting and stimulating for the students. The goal is that the projects are so compelling that the students will really want to work on them. This can be some sort of simulation, where the result is useful or interesting. Another popular type of project is a computer game that applies the technical material that has been studied.

The project solutions are graded normally, but I have also picked a few (3-5) of the most interesting solutions and published them on the course website, with the permission of the authors. The students allow publication almost without exception. This publication is a pat on the back of the students that have spent a lot of time and effort on the project, but also a kind of feedback for the other students. They can see different ways of approaching the project. The selected solutions are not necessarily the ones with the highest grade. I try to select solutions that show creativity and innovative techniques.

The projects are obviously very different between courses. Below are some examples:

- In Computer Graphics I usually have one of the projects to write a 3D version of well-known old computer game. Some examples include 3D Tetris (see solutions), 3D version of the Game of Life (see solutions) and a 3D Snake game (see solutions). The projects always change between years and the students can see old projects and solutions for comparison.
- Discrete Mathematics (has been a while since I taught that course) gives an excellent opportunity to mix mathematics and programming. There I designed a project that involves cracking a simplified version of the cryptography method RSA. Our method had key length of

164 bits instead of the usual 1024 or 2048 bits (see project in Appendix A30). At the time RSA was often in the news, as web-based online shopping was becoming more common. Another project was to calculate when it is worthwhile playing in the Icelandic Lotto (see project in Appendix A31). Only 45% of the income goes into prize money in the Lotto, but when the first prize is not won, it carries over to the next week and the total amount of prize money thus becomes more than 45%. Sooner or later the total amount of prize money on offer is higher than the money spent on tickets that week. However, more people tend to buy tickets when the jackpot has carried over several times, so the students must make some assumptions on how many tickets are sold each time. This is an example of a project where the result is interesting and useful for others and the students liked being able to share the result with friends and relatives.

- In Analysis of Algorithms there have been many different types of projects, but I have tried to design projects that are related to something currently happening in the field. For example, in 2011 Java SE 7 replaced their built-in sorting algorithm for arrays of non-primitive types (changed from Merge sort to Timsort). This change did not get much publicity, so the project here was to consider this change, compare the algorithms, and especially look at Timsort, which is not a well-known sorting algorithm (see project in Appendix A32). Another project in this course was to write code to play the board game Battleship, using some simplified assumptions. The main challenge was to find an algorithm to play the game such that the number of explosions was minimized (see project in Appendix A33).

Result:

The students really like the change of pace from the weekly homework over to getting two weeks to concentrate on one specific challenging project. There have been several comments to that effect in the teaching evaluations. For instance, in Analysis of Algorithms in 2019 (see Appendix A34): "*The classes and projects were interesting, I was even often excited to work on certain projects e.g., to analyze the sorting algorithms, it was a bit like solving a mystery*"[8]

Students with good solutions to the projects have sometimes used them on their CV when looking for work, especially if their solution has been highlighted on the course homepage. The homepages are open and can be presented as proof of that to potential employers.

The projects are often one of the main things that students remember from the course. I have for example spoken to former students of mine and they will recall how much fun it was to tackle the RSA project in Discrete Math or build the computer game in Computer Graphics.

Development:

Different students are motivated by different types of projects. Some like programming or computation, while others find it more interesting to write a summary of a particular topic. I soon started offering a choice between two or three types of projects. For example, in Analysis of Algorithms when one option was to write a program to play the Battleship game (see above), another possibility was to study and write a survey paper on Bloom filters, which is an interesting probabilistic data structure that has recently become popular in bioinformatics. In this case around half of the students chose each option.

It can be quite a challenge to design interesting and appropriate projects each time, especially when you need to have more than one version. However, I take this responsibility very seriously and

[8] Translated from: "*Tímarnir og verkefnin voru áhugaverð, var meira að segja oft spennt að fá að vinna viss verkefnin t.d. að greina sort reikniritin, það var smá eins og að leysa ráðgátu*"

do spend a considerable amount of time to make sure that a project is both fun and instructive for the students. I am always on the lookout for project ideas. Whenever I hear of an interesting puzzle or a fun game, or if I read about a new discovery or scientific breakthrough, I will wonder if it can be made into an interesting project. This can easily get out of hand, but can also be quite enjoyable.

Case 4: Using TA meetings to discuss pedagogy

Background:

Larger courses in the School of Engineering and Natural Sciences (SENS) will usually hire teaching assistants (TAs) to help with homework grading and oversee lab sections. In the CS department most of the TAs are older undergraduate students or sometimes master's students. Most of them have never taught before and are often unsure of how to handle classroom teaching. When I headed the SENS teaching committee in 2012, we decided to hold a course in pedagogy for TAs at the beginning of the fall semester. This was just one 3-hour evening class that Guðrún Geirsdóttir from CTL taught. This class was very popular, but for various reasons it was only held two or three times. By that time, I was teaching the large first year courses CS1a and CS1, which needed between 5 and 10 TAs each time. I realized that it was important that the TAs have some knowledge of pedagogy and preferably have similar views on how to apply it. A large part of the students' contact time in the course is through the TAs. Therefore, it is important that they also promote the course vision of encouraging active learning and continuous study. In 2014 I decided to use the weekly TA meetings to discuss various aspects of pedagogy that I considered important in teaching the course. Additional benefits were to empower the TAs to become better teachers, provide a support structure for them, and build a strong teaching team for the course.

Implementation:

In CS1a and CS1 we would hold a meeting with all the TAs at the end of each week. The meeting would be around 45 minutes and would have two parts: First the subject material of the current week and the associated homework. The TAs would get sample solutions of the homework and directions on how to grade and give feedback in the study sections. The other part of the meeting would be on pedagogy. I would bring up a pedagogical topic and we would discuss that until the end of the meeting. We often got very lively and useful discussions in this part. The topics were quite diverse and consisted mostly of things that I myself had recently studied in the CTL course in the spring of 2012. Some of the topics were: Formative assessment and the value of feedback, Active learning, Cheating and how to handle it, Deep vs. surface learning, Study methods, such as distributed and interleaved practice. I tried to link the topics to the organization of the course and what they, as TAs, could do to support that.

Result:

Those who are chosen to be TAs are usually among the best students in the second or third year of the bachelor studies. We get between 20 and 30 applications to be a TA in CS1, and we pick the best 5 – 10 of them. Therefore, the meeting participants were very good students, who had recently taken the course, and were very receptive to ideas on how to teach it in the best possible way. They were also full of ideas themselves and I would often get excellent suggestions on course improvements in the meetings.

I have little concrete evidence on what this pedagogic element of the TA meetings changed in the implementation of the course. However, the average TA grade in the teaching evaluations did improve after I started using the TA meetings in this way. For example, in 2012 and 2013 the

average TA grade was 7,9 in both years. After adding this feature in 2014 the average TA grade was 8.6. In 2015 it was 8,5, then 8,7 in 2016, 8,0 in 2017, and 8,9 in 2018 (see details in Appendix A40). My own teaching grade did not change very much through the period, so that was not a factor in the increasing TA grade.

I have met again some of the students that were TAs with me in CS1 and they have told me that for them these meetings were almost equivalent to a seminar course. They learned many new things about the course material and pedagogy.

Development:

I did not write down anything about the pedagogical topics that we discussed in the meeting. The meetings were informal, and the topics changed slightly between the years. My experience with this has therefore not been so beneficial to other teachers in my department. I have mentioned this idea to a few teachers informally, but for this approach to be successful the teacher would probably have to have some pedagogical training.

I have used this type of TA meetings in other smaller courses, with one or two TAs, after I stopped teaching CS1. The pedagogical part would be even less formal there. However, I think it is important for the teacher and the TAs to have similar pedagogical vision for a course to be successful, no matter how many students are in it.

As a result of writing up this case for the Teaching reflection I have started thinking about how the idea can be developed in the future. It has the greatest effect in the large first-year courses that employ several TAs. One option would be to work with teachers from the math department, who use TAs in a similar way. We could write up a list of possible pedagogic elements to cover in TA meetings with a short description and talking points for each one. Then this initiative would be able to survive in the courses even though the teachers are replaced. Each teacher could even add or modify points to suit their teaching style.

Future teaching development

Distance learning challenges:

With the advent of Covid and online teaching our students have mostly lost out on one of their most valuable resource: the informal interaction and support from fellow students. We need try to offer some alternative means of informal interaction between the students. One avenue would be to get the assistance of the student unions to build informal study groups for each class. In our courses we can also use various methods to encourage collaboration between students in solving class projects.

There are some ideas that I have used and will continue to try out over the coming semesters. A popular one is to use the "breakout room" facility in Zoom to force the students to get to know each other and get support from one another on the quizzes. Group work can work as an incentive in the Zoom classes, especially now when the students are not meeting each other much. There are also interesting online tools emerging that try to mimic the experience of being in a classroom, such as Gather.town, Mozilla Hubs, and Wonder. These are all tools that I intend to explore in coming semesters.

Lecture recordings:

I have recorded the lectures in most of my courses in Panopto since 2015 and shared them with the students on the University course management system. The students have been very thankful for that (see teaching evaluations). However, I have misgivings about how much the recordings contribute to their learning. Students that follow the lectures and then use the recordings to revisit a particularly difficult topic benefit from having the recordings. However, it is tempting for less disciplined students to use their existence as an excuse to skip the lectures and attempt to scan quickly through the recordings later. This is fine for students that are experienced and know their own capabilities and limitations, but first year students tend overestimate their capabilities and underestimate their limitations.

It would be useful to find ways to get the benefits of the lecture recordings without the disadvantages. So far, I have been using several methods to encourage the students to work continuously during the semester (see Case 2) and I will keep on trying different ways to accomplish that goal.

Encouraging colleagues to study pedagogy:

When I took the second CTL class on university pedagogy in 2019, I was surprised to note that, out of around 25 teachers in the course, I was the only teacher from the School of Engineering and Natural Sciences (SENS). I also realized that I am the only teacher in the Computer Science department that has taken any of those courses. I am not quite sure if SENS is underrepresented in the group of teachers that have taken the CTL courses. I do know several colleagues in other departments within SENS that have finished all three courses of the CTL diploma program. However, I also know many more who have not taken any of the courses. Considering how valuable the courses have been to me, I really want to step up efforts to get more teachers to take at least one of the pedagogy courses. That could be done by having the school offer a variety of incentives to make it worthwhile for teachers in SENS to take such a course. I am convinced that it would be one of the most effective ways to improve the quality of teaching in engineering and natural sciences at the University of Iceland.

References

Biggs, J., & Tang, C. (2011). *Teaching for quality learning at university* (4th útg.). Open University Press.

Bonwell, C. C., & Eison, J. A. (1991). *Creating excitement in the classroom.* Washington D.C.: ASHE-ERIC Higher Education Reports. Sótt frá https://files.eric.ed.gov/fulltext/ED336049.pdf

Deslauriers, L., McCarty, L. S., Miller, K., Callaghan, K., & Kestin, G. (2019). Measuring actual learning versus feeling of learning in response to being actively engaged in the classroom. *Proceedings of the National Academy of Sciences, 116*(39), 19251-19257.

Deslauriers, L., Schelew, E., & Wieman, C. (2011). Improved learning in a large-enrollment physics class. *Science*, 862-864.

Dunlosky, J., Rawson, K. A., Marsh, E. J., Nathan, M. J., & Willingham, D. T. (2013). Improving students' learning with effective learning techniques: Promising directions from cognitive and educational psychology. *Psychological Science in the Public Interest, 14*(1), 4-58.

Karpicke, J. D. (2012). Retrieval-based learning: Active retrieval promotes meaningful learning. *Current Directions in Psychological Science, 21*(3), 157-163.

Krathwohl, D. R. (2002). A revision of Bloom's taxonomy: An overview. *Theory into Practice, 41*(4), 212-218.

Kugel, P. (1993). How professors develop as teachers. *Studies in Higher Education, 18*(3), 315-328.

Richland, L. E., Kornell, N., & Kao, L. S. (2009). The pretesting effect: Do unsuccessful retrieval attempts enhance learning? *Journal of Experimental Psychology: Applied, 15*(3), 243-257.

Roediger, H. L., & Karpicke, J. D. (2006). Test-enhanced learning: Taking memory tests improves long-term retention. *Psychological Science, 17*(3), 249-255.