Pass/Fail Grading and Educational Practices in Computer Science

Torstein J. F. Strømme
Department of Informatics, University of Bergen, Norway
torstein.stromme@uib.no

Abstract

Binary (pass/fail) grading have been shown to have benefits with respect to mental health and collaboration, and is argued to promote a deep approach to learning. However, diverging results with respect to academic achievement suggests that the full benefits of binary grading are contingent on underlying factors, such as how the teaching and learning activities in the course are designed.

We here present experiences and student feedback for an intermediate level course in computer science that is graded using pass/fail, and which is highly successful both in terms of of student enjoyment and academic achievement. Survey results also indicate that students apply a deeper learning approach towards the course than average.

Drawing on examples and findings from this course, we argue that the following three practices makes a binary graded course in computer science successful: a) a sufficiently high bar for passing, b) clear course requirements, and c) the use of formative assessment.

1 Introduction

The assessment is arguably the most important landmark that students use when deciding how to work with a class, and what to focus on in their studies [1]. How assessment is done has therefore attracted great attention in education research. For instance, highly regarded principles such as constructive alignment [2] and integrated course design [3] both place heavy emphasis on making assessments align well with the intended learning outcomes, in order that the perceived authenticity will motivate the students to go deeper with the material. It is also well established that formative assessment, the practice of letting assessment be an integral part in the process of learning and working with the material, carries significant benefits [4].

In addition to the assessment format and its frequency, there is also a question of grading, and in particular the grading scale. Across the world there are multiple traditions in higher education when it comes to grading scales, for instance the six-tier A-F scale used in the ECTS framework in Europe, or the five-tier scale (A/B/C/D/F) commonly used in the United States. However, in many of these systems there is also an option of using a binary (pass/fail) grading scale for individual courses. In Norway, Jørgensen and

This paper was presented at the NIK-2019 conference; see http://www.nik.no/.

Bråten [5] recently argued that a ternary (pass/fail/honors) grading scheme should also be made available as an option.

Proponents of binary (and sometimes ternary) grading scales argue that they give less stress and anxiety among students, increase collaboration and sparks a more intrinsic motivation than finer-grained scales do, all without a decline in academic achievement [5] (see also [6, 7, 8, 9, 10]). However, some studies also find that academic achievements actually do decline under binary grading schemes [11, 12], and that students with pass/fail grades suffer in the job market [13]. It is also argued that the stress caused by competition in the job market will cause students with only pass/fail grades to pick up more extracurricular activities in order to distinguish themselves in job applications, reducing the time available for studies [14, 15].

Most of the positive effects of binary grading described in recent literature have occurred within the context of medicine in North America, and it is possible that the particularly rigid traditions in medical studies versus other subjects explains why the results appears to be diverging. In any case, it is clear that simply changing the grading scale is not a magic wand one can use without regard to how the course is otherwise designed. The current paper is a first look into what can make a pass/fail -course successful in the context of computer science.

Aim: To describe educational practices which are favorable when designing a successful pass/fail course in computer science.

2 The Course

We begin by describing the course, called *Algorithms Engineering* (course code INF237). It yields 10 ECTS credits upon completion, and is taught every spring semester. It is the only regular course with a pass/fail grading scheme in the Department of Informatics at University of Bergen (UiB), Norway. The course is practically oriented and involves a lot of programming, however a good understanding of the theoretical algorithms is important. There are no formal requirements for the course, but it is highly recommended to have completed one or two courses in algorithms and data structures prior to enrolling. It is typically taken the last semester of bachelor studies, but almost 40% of students are on the master level. The course is not mandatory for any degree, however it is recommended for students pursuing a masters degree in algorithms.

There is no final exam in the course, but the final grade is decided by work submitted throughout the semester. All submissions are entirely auto-graded[1], so a student will at every point throughout the semester be aware of exactly how (s)he is doing within a few seconds after submitting a task on an assignment. Even though the feedback from the course judge is limited in its level of detail, the learning process is fully integrated with the assessment – in this sense the assessment is formative. There is no penalty for attempting the same task many times before getting it accepted.

In order to pass the course, a student is required to successfully pass 13 assignments, one for each week of lecture. Typically, each assignment consists of four tasks, and the requirement for passing the assignment is to successfully solve at least two of them. The assignment for a particular topic is released the same day as the lecture on that topic, and is accepting submissions in a three to four week period.

In addition, there are two oral presentations in which the students will be asked to explain a chosen part of their submitted work. Some students perceive this oral

[1] Complete source code for each task is submitted to the online judge Kattis at https://uib.kattis.com.

presentation as an exam, however the primary purpose of the presentations is to detect cheating, and most students find the presentations quite easy in retrospect. During the presentation, students will be asked to present some of their submitted tasks, for instance solutions which the instructors find either suspicious or particularly interesting.

With respect to cheating, plagiarism detection is automatically conducted by the course judge. All submissions for a problem are checked against each other, including (if the task is publicly available) thousands of submissions from around the world[2]. Instructors are made aware of possible plagiarism cases in the graphical user interface of the course judge, and a few cheating cases have been detected each year.

Students in the course are also required to participate in at least two programming contests, and have the option to participate in four or five. If a student participates in more than two contests, (s)he will receive wild cards which can be used to skip one task in an assignment. Two wild cards may not be used towards the same assignment.

Over the years we cover in this paper (2017, 2018 and 2019), the course was taught by three different instructors and had different sets of exercises, but maintained essentially the same structure as described above. In 2018 the course had two different instructors due to paternity leave, where the author taught the first half of the course, and the same instructor as in 2018 taught the second half of the course. A third instructor taught the course in 2019.

3 Methodology and background data

A digital survey was distributed to all students who had successfully solved at least half the assignments in either 2017, 2018 or 2019 (respectively 16, 39 and 29 students, a total of 84). This includes students who ultimately failed or dropped the course due to either cheating or simply not finishing all the assignments (5 students). The survey was sent to all respondents during summer 2019, so those who took the course in 2017 and 2018 needed to recall experiences much further back than those who took the course in 2019.

The questionnaire consisted of three main sections, and there was an opportunity to make separate comments on each section of the submission. There was also a field where the respondents could provide general constructive feedback about the course if they wished.

In the first section we ask the respondents to compare the course to other courses they have undertaken in computer science. Claims (e.g. “The course is easy”) are presented to the respondents, and answers are reported on an modified Likert scale where the alternatives are:

(1) This statement applies much more to INF237 than to my other courses.
(2) This statement applies slightly more to INF237 than to my other courses.
(3) This statement applies roughly the same to INF237 as to my other courses.
(4) This statement applies slightly less to INF237 than to my other courses.
(5) This statement applies much less to INF237 than to my other courses.

In the second section, we ask a few items specifically on experiences with the pass/fail grading scale. Answers here are recorded on a standard Likert scale.

[2] While the course judge Kattis allow us to create tasks which are private to the UiB domain, most tasks are picked from a large database of problems open for everyone at https://open.kattis.com

The third and final section of the survey is an edited version of Biggs' revised two-factor study process questionnaire (R-SPQ-2F) [16], in which the students are asked to compare their study habits in Algorithms Engineering to their computer science studies in general. The intention is to get an indication for how the course itself affects deep and surface learning, as opposed to measuring the learning approach of the student mass in general.

In our edition of the R-SPQ-2F questionnaire, the statements respondents face are the same as in the original (except the question 18, which is not applicable for Algorithms Engineering and was removed), however the 5-point scale ranging from "always or almost always true of me" to "never or only rarely true of me" was replaced by a 5-point scale where the options were instead:

(1) This statement applies much more to INF237 than to my studies in other courses.
(2) This statement applies slightly more to INF237 than to my studies in other courses.
(3) This statement applies roughly the same to INF237 as to my studies in other courses.
(4) This statement applies slightly less to INF237 than to my studies in other courses.
(5) This statement applies much less to INF237 than to my studies in other courses.

For each of deep approach and surface approach, half the statements apply to them as described in [16]. In order not to confuse our results with those of a proper R-SPQ-2F questionnaire, we will simply report the average value answered to the statements in each of the two categories.

4 Results

Response rate for the survey was 52% (for years 2017, 2018 and 2019 respectively 56%, 54% and 48%). One response was removed from the data set, since the answers appeared inversely correlated to the written comments of the same submission. Hence, final sample size used in the analysis is $n = 44$, out of which 42 answered every multiple choice item.

Demography

Gender	86% male, 9% female, 5% prefer not to say
Age*	Average 23.3, SD 2.57, min 19, max 31
Course year	20% 2017, 48% 2018, 32% 2019
Towards degree*	39% master, 59% bachelor, 2% other

Table 1: Background of respondents. *At the time when course was taken.

There was no item on the survey where respondents across different years answer significantly different, except one: Respondents who took the course in 2017 recommend it slightly more (2017 average: 1.0, 2018 average: 1.57, 2019 average: 1.54).

Of students responding, 85% belong to the Department of Informatics at University of Bergen (UiB), whereas the others were equally distributed between the Department of Information Science and Media Studies at UiB and the Department of Computing, Mathematics and Physics at collaborating institution Western Norway University of Applied Sciences (HVL).

Claims about the course	Ave	1	2	3	4	5
The objectives of the course are clear.	1.61	50	41	7	2	0
The course requires a high cognitive level of reasoning.	1.70	43	43	14	0	0
The course requires memorizing many details.	3.52	5	16	25	32	23
The course is easy.	4.20	0	5	11	43	41
The course is fun.	1.34	73	20	7	0	0
The course has a high workload.	1.48	59	34	7	0	0
The course is relevant to my future career.	2.16	27	41	23	7	2
I learned a lot in this course.	1.52	52	43	5	0	0
I recommend this course.	1.44	67	26	2	5	0

Table 2: About the Course. The scale ranges from 1 ("this statement applies much more to INF237 than to my other courses") to 5 ("this statement applies much less to INF237 than to my other courses"). Values on the right are percentages.

About the course

Respondents generally report favorable views of the course, saying that the course has clearer objectives, is more fun, and more relevant to their future careers than a typical computer science course (see Table 2). 93% of respondents report that they recommend it more than other courses. It is also a pretty strong consensus that the course is hard, has a high workload, and requires a high cognitive level of reasoning, but also that one learns more in the course than what is normal. Some typical comments include:

> Toughest course I took in UiB but also the one I probably learned the most from and it was highly beneficial for my future career. (2018 master student)

> I would have recommended the course based on what we learn, but it requires a lot of work, so it can't be recommended for anybody. (2019 bachelor student)

> I liked the course, and i really liked the course style, it forces you to work with the course evenly over the whole semester, and that was good for me. And the topic of the course is really fun also! (2018 bachelor student)

> I felt the course was very good. Best class the entire semester. It was interesting and challenging, but not overly challenging either. (2019 bachelor student)

> This is one of top three courses I have taken at university (...). It was a very heavy subject, but it was luckily engaging enough that I never considered giving up. I believe, though, that I wouldn't have made it through without a group of people to work and discuss with. (2017 bachelor student)

> Love that you use everything you learn. It was my favourite course. Think I spent 90% of my time on this course, and winded up only getting 20 [ECTS credits] that semester. On the downside, I have nothing but a – pass – to show for all my efforts in it. (2018 master student)

Claim	Ave	1	2	3	4	5
I would have preferred an A-F grading scheme in INF237 over the current pass/fail scale.	4.07	2	9	11	34	43
I would have preferred a pass/fail grading scheme in the computer science core courses (for example INF100/101/102) over the current A-F scale.	3.80	5	7	34	14	41
I prioritized other courses than INF237 because they used an A-F grading scheme	4.30	0	9	7	30	55
If all my other courses also had a pass/fail grading schemes, I would have spent more time on INF237.	3.88	5	12	16	26	42
For a course with a pass/fail grading scheme to be successful, it is important that it is challenging to receive a passing grade.	1.88	37	42	16	5	0
For a course with a pass/fail grading scheme to be successful, it is important to receive continuous feedback on how you are performing throughout the semester.	1.60	53	33	14	0	0

Table 3: About the pass/fail grading scale. The scale ranges from 1 ("I strongly agree") to 5 ("I strongly disagree"). Values on the right are percentages.

About the pass/fail grading scale

Respondents generally agree that pass/fail is a suitable scale for the course, even though 13% of respondents would have preferred an A-F scale. The main criticism of the binary grading scale is that a pass/fail does not reflect the work they put into the course when calculating grade averages (in Norway, pass is treated like a C when calculating grade point average). Consider these comments:

> I thought the pass/fail was unsuitable for inf237. The amount of work that I put in was probably enough to get an A or a B, but instead I got a pass... The amount of work put into inf237 also affected the grades in my other courses, so my average became lower than I wanted. (2018 master student)

> The only downside of pass/fail, is that it counts as a C when calculating average grades. It is therefore hard to justify using that much time on the class. The reason I still did was because I enjoyed it and found it challenging. (2017 master student)

On the other hand, a significant majority (76%) prefer binary grading in the course. Yet, when it comes to the introductory core courses in computer science, the picture is entirely different; only 11% of respondents report that they would like them be graded pass/fail, whereas 57% prefer the finer grained A-F scale. There is in other words a huge gap between the support for binary grading in Algorithms Engineering versus in the core courses in computer science.

As for traits that the respondents believe are important in order to make a course with binary grading successful, 78% agree that it should be challenging to receive a passing grade, and 87% agree that it is important to receive continuous feedback on performance throughout the semester. A recurring comment was also the importance of clear and objective criteria for passing:

> Very clear conditions for passing the course [makes a course suitable for pass/fail grading]. Students are pretty much hardwired to optimize for good grades, and in courses where what constitutes a pass is more nebulous, you spend less time learning and more time obsessing over what the instructor is looking for, and what is enough to pass. I've occasionally spent entire days on such discussions, and it is very counterproductive. It worked well in INF237, because the objectives were very clear (...). (2017 bachelor student)

> The course objectives should be unambiguously measurable, meaning the criteria must be clearly defined. INF237 does one thing correctly, the fact that with each task the student knows if he/she will pass or fail the course. This is good because the student knows what he/she did not understand early on the course. (2019 bachelor student)

Other comments mention that classes without final exams are a good fit for binary grading, but that good systems for detecting plagiarism are important, and that many small assignments is more suitable than a few large ones. We also note that students generally do not down-prioritize the course compared to courses with fine-grained grading scales. See Table 3.

Study habits

$n = 42$	Score	SD	Min	Max	α
Deep approach	2.42	0.51	2.26	2.57	0.77
Surface approach	3.57	0.47	3.43	3.72	0.74

Table 4: Study habits. The scale ranges from 1 to 5, where values below 3 indicate that this approach is relatively more applied in Algorithms Engineering, and values above 3 indicate that this approach is relatively more applied in other courses. Min and max values indicate the 95% confidence interval for the score, and α is the Cronbach alpha -value.

Our modification of Biggs' R-SPQ-2F questionnaire is not well suited to analyze the actual level of deep learning and surface learning of the respondents, nor is it the intended purpose. It rather attempts to measure how a particular course influence these learning strategies. We find that a deep approach is applied more in Algorithms Engineering than in other courses, whereas a surface approach is applied more in other courses, see Table 4. The Cronbach alpha for deep approach was 0.77, and for surface approach was 0.74 (slightly higher than those reported by Biggs for the R-SPQ-2F, respectively 0.73 and 0.64 [16]).

Even though the majority report having a deeper than average approach to Algorithms Engineering, it is not the case for everyone. A respondent writes:

> The main reason I answer in the direction of not spending time on learning topics in depth or spending time on studying topics on your own is that for me 237 was a course where my time was stretched to my very limit (...). If 237 was slightly easier, I would definitely spend more time learning more about everything from the course. (2018 master student)

There were also a few respondents that were confused about the questionnaire itself, since the modification of the answer alternatives did not allow the respondent to express their full view on the issue. For instance, we received this feedback:

> [This section was] a bit hard to answer, since the scale does not take into account whether I agree to the claim at all (that is, neither in INF237 nor other subjects). I therefore answered 3 both when I agreed and disagreed to a claim, if the claim applied equally much to INF237 as other courses. (2017 bachelor student)

Course statistics

There are a few statistics available from the online course judge (Kattis), indicating how students have worked with the course. We report here the extent to which students work more to improve on an assignment after it is already accepted, as this indicates that the student is using a deep approach to their studies. This is something every single student have actually done to some extent, though exactly how much varies. There are two basic ways to work on a submission after it is already accepted:

The first way is to submit improved versions of already accepted assignment tasks. On average, a student submits such improved versions at least 10.8 times during the semester. We remark that this is merely a lower bound, since students also have the option of submitting their code to a public judge for which we do not collect statistics. Both on the public and on the course judge there is a scoreboard ranking the best solutions by execution time, adding an element of gamification some students appreciate.

The second way to go beyond the requirements on an assignment, is to solve more than the required number of tasks. This requires significant extra effort, as the difficulty of the extra task is typically higher. A median student solves 3 tasks (average: 3.8) more than the minimum during the course, which corresponds to roughly 11% more work than necessary (though this is hard to quantify in terms of working hours). Only 5% of students passing the course have not done any excess tasks.

5 Discussion

Algorithms Engineering is a binary graded course where students have frequent and regular assessments with automatic grading, which we regard as a type of formative assessment. The bar for passing the course is high and rigid, and for many students this means that relatively many working hours are required to pass. However, the failure rate is low - among students who completed at least half the assignments (that is, who do not drop the course early), only 3 out of 83 students have been unable to successfully complete the course.

What separates strong from weak students is thus not primarily what is learned, but how much time is spent learning it. In this sense, the course achieves the elusive goal of outcome-based education: By the end of the educational experience, every student have achieved the intended goal [17]. A "pass" in the course is hence not merely a diploma for effort and participation, it is a true certificate that the candidate knows the topic.

The course appears to encourage a deep approach among students, and discourages a surface approach (see Table 4). This is also supported by the extra work students put into the course beyond what is required; every single student in the course have done some unnecessary improvement to their submission at some point throughout the semester, and only 5% of passing students solved only the minimum required number of tasks. Since this also requires solving conceptually harder tasks than necessary, it demonstrates a genuine interest and a deep approach to the course.

We believe the course offers some valuable ideas for designing courses that work well with a pass/fail grading:

A challenge. In order to facilitate a good learning experience, it is important that the learning outcomes of the course are (perceived to be) significant. This view is also supported by a large majority of the students who took the course (Table 3). While binary grading may under some circumstances make students sloppy, our experience is that it can also sharpen the students when the bar is high.

Clear requirements. While clear requirements is an advantage in all courses, it is of particular importance when the difference between two adjacent grades is so large. If a student receive "fail", the student him/herself should ideally know so before the instructor. Students repeatedly mention this as an important trait of a well-designed pass/fail -course. Clear requirements in combination with effective strategies to detect cheating also means that a course is perceived as fair.

Formative assessments. Students need to always be aware of how well they fare compared to the course requirements. This is important for continuous motivation throughout the semester, and for a good experience as a whole. An overwhelming majority of students agree (Table 3).

While it is not directly investigated in this paper, we also believe that the instructor and course staff taking a supportive role "helping the students fight the inflexible requirements" is a big positive. Due to the rigidity of the course requirements it has indeed been possible to assume this role in the vast majority of student-instructor interactions. We also see that many students appreciate the supportive atmosphere in the course, where discussing the exercises with one another is encouraged. A competitive gamification element in the course still exists, but it is detached from the final grade.

The above mentioned traits are not equally present everywhere in computer science education, and the wide gap between the support of binary grading for one class versus others (see Table 3) suggests that not all courses are created equal. If we still wish to pursue the benefits of binary grading (for instance with respect to mental health, deep approach learning and collaboration), then a complete redesign might be necessary in many courses. On the other hand, the traits that makes pass/fail work well for Algorithms Engineering can certainly have positive effects in the context of finer grading scales as well; *any* course can be challenging, have clear requirements and use formative assessment.

Binary grading does, however, make it easier and more natural to follow those principles. For instance, clear course requirements are much easier to formulate for a binary scale than for a finer scale (though not impossible, as seen for instance in Nilson's specifications grading [18]). By lightening the burden of marking assessments, pass/fail can also free more resources to provide formative assessment. Lastly, we believe a certain level of challenge arise naturally when combining frequent formative assessment with binary grading.

And while binary grading supports a high bar for passing, clear course requirements and formative assessment, the converse is doubly true: A binary graded course without a high bar can make students uninterested and prioritize other subjects perceived to have higher learning outcomes. A binary graded course without clear requirements and formative assessment can leave the student flustered, frustrated and unmotivated. We believe these principles and binary grading are mutually supporting each other.

Limitations

The sample size for this survey is only $n = 44$, though the trends seems strong enough that the reported results are still significant.

Our modification of the R-SPQ-2F is not a validated instrument, and we do not know how the scores will look if students in other courses were to answer a similar survey. The results are therefore merely an indication that deep approaches are employed more. Moreover, there was a slight confusion over how the questions should be answered.

Students electing to take the course are often aware beforehand what they are signing up for, including that the course has a high workload and very stringent requirements for passing. This self-selection process will bias any results we record, and it is hence not straightforward to determine which effect the course format itself is responsible for. It is, for instance, entirely plausible that the particular format of the course attracts specifically students who appreciate pass/fail as a grading scale. However, the wide gap in support for binary grading in Algorithms Engineering versus the introductory core courses indicate that it is not obviously so.

It is also plausible that the reputation of being a work-intensive course combined with offering small rewards (or even punishment) in terms of calculated grade point average attracts a type of student that is naturally predisposed to deep learning approaches. However, if the deep and surface approaches were merely traits of the individual student, one would expect them to employ deep learning approaches equally much in all courses.

Conclusion

Binary graded courses in computer science can be successful with respect to both student enjoyment and academic achievement, and also stimulate deep approaches to learning by employing the following educational practices:

- Set the bar for passing sufficiently high.
- Provide clear course requirements.
- Use formative assessments.

We argue that there is a synergy between these principles which work especially well with pass/fail grading. The combination of formative assessments and clear course requirements ensures that every student can closely monitor his/her own progress and allocate the required resources to work on the topic. This allows raising the bar for a pass without losing many students, ensuring high academic achievement. After all, students appreciate challenges as long as they overcome them.

Further work

The course discussed in this paper is a free elective with a certain reputation among students, giving rise to a bias in the student mass. It would be interesting to draw experiences from a mandatory course and investigate to which extent the current findings still hold. A next step might be a qualitative study which spans across multiple classes that has binary grading; this would provide a better background for further research in the area.

Another open question is the relationship of stress/anxiety and difficulty. While existing literature have established that binary grading reduce stress and anxiety [10], it is not entirely clear how this relates to the perceived difficulty of obtaining a pass. What happens when the challenge becomes too challenging?

Thanks

Many thanks to Daniel Lokshtanov and Jan Arne Telle who were the other instructors of the course during the period of this study. We would also like to thank the many instructors before them who helped develop Algorithms Engineering into what it is today.

References

[1] Arild Raaheim. Eksamensrevolusjonen. råd og tips om eksamen og alternative vurderingsformer. *Nye perspektiver på evalueringsformer i universitetspædagogik*, page 87, 2016.

[2] J. Biggs and C. Tang. *Teaching For Quality Learning At University*. SRHE and Open University Press Imprint. McGraw-Hill Education, 2011.

[3] L. Dee Fink. *Creating significant learning experiences: An integrated approach to designing college courses*. John Wiley & Sons, 2013.

[4] Neal Kingston and Brooke Nash. Formative assessment: A meta-analysis and a call for research. *Educational measurement: Issues and practice*, 30(4):28–37, 2011.

[5] C. Jørgensen and H. Bråten. Can 'passed with distinction' as a new grading scale favour the transition towards formative assessment? *Nordic Journal of STEM Education*, 3(1):7–11, 03 2019.

[6] Daniel E. Rohe, Patricia A. Barrier, Matthew M. Clark, David A. Cook, Kristin S. Vickers, and Paul A. Decker. The benefits of pass-fail grading on stress, mood, and group cohesion in medical students. *Mayo Clinic Proceedings*, 81(11):1443 – 1448, 2006.

[7] Robert A Bloodgood, Jerry G Short, John M Jackson, and James R Martindale. A change to pass/fail grading in the first two years at one medical school results in improved psychological well-being. *Academic Medicine*, 84(5):655–662, 2009.

[8] Darcy A Reed, Tait D Shanafelt, Daniel W Satele, David V Power, Anne Eacker, William Harper, Christine Moutier, Steven Durning, F Stanford Massie Jr, Matthew R Thomas, et al. Relationship of pass/fail grading and curriculum structure with well-being among preclinical medical students: a multi-institutional study. *Academic Medicine*, 86(11):1367–1373, 2011.

[9] Lynne S Robins, Joseph C Fantone, Mary S Oh, Gwen L Alexander, Marshal Shlafer, and Wayne K Davis. The effect of pass/fail grading and weekly quizzes on first-year students' performances and satisfaction. *Academic Medicine*, 1995.

[10] Laura Spring, Diana Robillard, Lorrie Gehlbach, and Tiffany A Moore Simas. Impact of pass/fail grading on medical students well-being and academic outcomes. *Medical education*, 45(9):867–877, 2011.

[11] Richard M. Gold, Anne Reilly, Robert Silberman, and Robert Lehr. Academic achievement declines under pass-fail grading. *The Journal of Experimental Education*, 39(3):17–21, 1971.

[12] David E Suddick and Robert E Kelly. Effects of transition from pass/no credit to traditional letter grade system. *The Journal of Experimental Education*, 50(2):88–90, 1981.

[13] John A. Dietrick, Michael T. Weaver, and Hollis W. Merrick. Pass/fail grading: A disadvantage for students applying for residency. *The American Journal of Surgery*, 162(1):63 – 66, 1991.

[14] Ray Guo and Jennifer Muir. Torontos honours/pass/fail debate: The minority position. *University of Toronto Medical Journal*, 85(2):95–98, 12 2008.

[15] Jan Frich, Knut E.A. Lundin, and Ingrid Os. Karaktersystemet avveining mellom ulike hensyn. *Tidsskrift for Den norske legeforening*, 134(1):14–15, 01 2014.

[16] John Biggs, David Kember, and Doris YP Leung. The revised two-factor study process questionnaire: R-spq-2f. *British journal of educational psychology*, 71(1):133–149, 2001.

[17] William G Spady. *Outcome-Based Education: Critical Issues and Answers.* ERIC, 1994.

[18] Linda Nilson. *Specifications grading: Restoring rigor, motivating students, and saving faculty time*. Stylus Publishing, LLC, 2015.