

Reflect on your interaction with bucket grading this semester. What were any positives, negatives... (r0_BucketGeneral)

I liked how it did not need to be perfect to get 100% (r1_BucketGeneral)

No negatives (r2_BucketGeneral)

I'd say positives, reduces the pressure to get a 100% (r3_BucketGeneral)

It pushed me to constantly try to get proficient instead of settling for ~80% (r4_BucketGeneral)

Generally fine! I don't think I had issues with it really, though it may have been nice to have been able to get a grade between 80 and 100 rather than just one or the other. (r5_BucketGeneral)

I think the bucket grading makes sense, as there isn't a "correct" way to implement a software. (r6_BucketGeneral)

It was kind of useless. It was really just a different way to show 40, 60, 80, 100. (r7_BucketGeneral)

not really (r8_BucketGeneral)

The system was good for focusing on learning outcomes rather than just points, but sometimes it was a bit unclear how to move up a bucket. (r9_BucketGeneral)

It made me more dedicated to get into the 100% grading bucket without fully focusing on getting everything 100% correct. (r10_BucketGeneral)

Sometimes I'm not sure the criteria for each bucket (r11_BucketGeneral)

I liked the bucket system! It allowed me not to worry about every single edge case which I often would do in 210 labs. Only two issues were in c2 with MIN and MAX behaving oddly as clusters and the post checkpoint feedback sometimes not feeling like it aligned with our cluster feedback before submission (r12_BucketGeneral)

in the beginning we struggled to understand the range for the buckets.. like was Proficient a guarantee your grade was between an 80-100. What caused more confusion is we tried to ask some lab and OH TAs for clarification and received some differing feedback but afterwards we figured that if we felt we had an extensive test passing test suite + were achieving Proficient we would be happy with our final grade (honestly I may still be confused if you can be Proficient and receive only an 84...) (r13_BucketGeneral)

I just didn't get the purpose, and when I don't understand the reasoning behind something like this, it really impacts my ability to approach it positively. My math teacher in grade 10 also did something like this, and I just purely don't understand? Is it supposed to help students? and how so? What research there behind bucket grading on students?? Or.. is this the research? Like am I right now apart of the research for bucket grading on students? Because let me tell you it was mostly negative. (r14_BucketGeneral)

It was an interesting grading scheme. I think percentages are more concrete and easier to understand, but I understand why it was done this way too (r15_BucketGeneral)

Feedback was somewhat unclear. (r16_BucketGeneral)

Not many confusions with the system but I did get frustrated with the lack of detail from the autobot when we received less than Proficient for a checkpoint. I found the speed of the Autobot to be great (apart from C0). (r17_BucketGeneral)

retry positive I could say (r18_BucketGeneral)

Didn't matter to me (r19_BucketGeneral)

It was good enough. (r20_BucketGeneral)

I guess we were uncertain the full meaning of proficient since our tests is still detecting the bugs, but once we are familiar with it, we are happy with it. (r21_BucketGeneral)

I wish we had a percent grade instead of buckets, its easier to know just how close you are then. (r22_BucketGeneral)

I felt it was very very helpful and successful. I felt I could spend more time writing good code because I didn't need to be scared of missing that one final edge case test that would stop me from getting 100%. (r23_BucketGeneral)

Having a bucket system help alleviated the pressure to need to pass all the smoke tests (r24_BucketGeneral)

Positives - you don't necessarily need to have 100% on your implementation to get Proficient. \Negatives - for c2 specifically, our group struggled to move from the Acquiring bucket to the Developing bucket, even though a majority of our tests were passing. (r25_BucketGeneral)

Good enough - accurate as to "real world" standards. (r26_BucketGeneral)

It was positives for me. (r27_BucketGeneral)

I liked the bucket grading because it didn't mean bashing my head against the wall trying to get from 92% -> 100% which I definitely would have done. (r28_BucketGeneral)

The bucket grading was not helpful. I'd rather have a grade that shows how many tests are passing to overall tests because that would be more useful than the bucket grading. (r29_BucketGeneral)

Positive since there is leeway however if we got less than that we have no idea how far off we would be. (r30_BucketGeneral)

Overall good, keep it imo (r31_BucketGeneral)

It was nice that it allowed some mistakes (failing tests) and still gave us proficient (r32_BucketGeneral)

It was a good system. Instead of a percentage, it made it nicer to read. Made me happier (r33_BucketGeneral)

It was a little confusing assessing how far off I am from the solution (r34_BucketGeneral)

I think it was good because we don't have to get 100% of the tests to pass to get a full mark (r35_BucketGeneral)

Overall, I found the bucket grading system to be positive. (r36_BucketGeneral)

Pretty good system I would say. (r37_BucketGeneral)

Positive: My first thought was that it didn't help, but looking back, if it had been actual percentages, I might have gone insane trying to hit 100%, so I think the buckets were a good way to lessen the pressure of hitting 100% (r38_BucketGeneral)

I think is good it evaluates everything holistically as it really shows how most software is not just perform query or add Dataset but how the whole product interacts (r39_BucketGeneral)

I liked it. It's hard to judge how "correct" your implementation was but it was alright (r40_BucketGeneral)

Only negative I had was that the feedback was very vague. (r41_BucketGeneral)

positive because less stress, but negative because we could be decreasing our grade and going in the wrong direction but we wouldn't be able to tell (r42_BucketGeneral)

we really tried to dial in on perfection, which was not helpful when we are busy with other courses (r43_BucketGeneral)

I liked it. We realized we had tests that didn't pass every single edge case, (the bug feedback that was released after checkpoints), but we were still able to get Proficient. I would say there are only positives with the bucket grading system :) (r44_BucketGeneral)

It was fine. (r45_BucketGeneral)

I would prefer if it was numbers / percentage - much easier to interpret. (r46_BucketGeneral)

It will be better if more details(e.g. what kinda test failing) can be given. otherwise people will just have no idea where it went wrong. (r47_BucketGeneral)

Positives: I didn't waste time trying to get the last 1% to get 100%. \Negatives: Not really a negative or confusion, but the bucket labels weren't really useful. The numbers spoke for themselves. (r48_BucketGeneral)

'- It was nice to know we didn't have to worry about debugging the last few percent points on a feature to get full marks! It significantly reduced stress around the project as I knew I wouldn't have to put in work with diminishing returns. (r49_BucketGeneral)

Experiences were positive when things were going well, but very negative when working hard at the cusp of reaching Proficient and being unable to direct our work efforts effectively from the grading information provided. Then it was very stressful. (r50_BucketGeneral)

There was no confusion with the bucket grading. :) (r51_BucketGeneral)

I think it was good, and gives you a sense of the quality of your code without assigning a concrete number grade. (r52_BucketGeneral)

I had no confusions, I think it's a better system than percentage grades for people who are aiming to do well, but I can imagine it was probably frustrating for those who were struggling. (r53_BucketGeneral)

Easy to get a reasonable mark even if only in the second bucket. (r54_BucketGeneral)

I think the buckets are an overall good idea to prevent getting stuck on one small thing that could stop a grade from going up. This being said, I wish the Beginning bucket could have had more info associated with it. (r55_BucketGeneral)

It would have been much more clear and helpful if we could just see our percentage rather than the bucket grade. (r56_BucketGeneral)

i feel that it was a better grading system then 100 on the checkpoint deliveries. no major complaints. (r57_BucketGeneral)

No, I think it was pretty clear how it operated. Overall I thought that the bucketed grading system was a huge positive. I don't think that having to pass every single test in a test suite would be an ideal way to evaluate projects. I can easily see this devolving into groups spending hours trying to get one or two final tests to pass. Although this sort of thing can be fun, it can also be a massive time sink of questionable value, especially since the odd failing test could be the result of different but reasonable interpretations of specs than an actual error. (r58_BucketGeneral)

The biggest confusion, although not from personal experience, could be that some unsuccessful fields are not directly related to the bug itself. This can result in wasting time on irrelevant fields. The bug could actually be in a field that passes the test, but is only revealed later in a different field. (r59_BucketGeneral)

The system contributed to giving me confidence on the project as largely autograded assignments sometimes feel like they require total perfection. (r60_BucketGeneral)

I liked the bucket grading system, maybe an extra level between developing and proficient would've been preferred (r61_BucketGeneral)

Fine. (r62_BucketGeneral)

The buckets saved us a lot of stress as perfectionists looking for the smallest bugs (r63_BucketGeneral)

I wasn't particularly confused by any part of the system, and I think it worked pretty well overall! I think my most negative experience was just seeing the bucket not change after I implemented a bunch of functionality XD. (r64_BucketGeneral)

Positives - don't have to pass all the tests to receive 100% \Negatives - statistics about bucket testing during class was not very helpful, especially when we were not doing too well (r65_BucketGeneral)

No complaints (r66_BucketGeneral)

It's nice not having to get all 100% of the tests passing to get a 100% on the code component. (r67_BucketGeneral)

The beginning bucket was the most confusing, as it was a range of grades. (r68_BucketGeneral)

I liked that as long as our implementation reached a certain proficiency we could achieve 100% for that phase. I think that makes more sense than

deducting some small amount of point for every reference test that was not passing. (r69_BucketGeneral)

Bucket grading felt confusing at times since I would always convert it to a percentage score, but otherwise it was useful to determine roughly how I was progressing in the checkpoint. (r70_BucketGeneral)

More detailed feedback was needed as to why it wasn't reaching a certain bucket. (r71_BucketGeneral)

I thought it was alright, it gave a good sense of where I might be at approximately (If i didn't have bugs that stopped me from actually being in a higher bucket) (r72_BucketGeneral)

The bucket grading didn't really affect me. (r73_BucketGeneral)

Overall I really liked the bucket grading because it felt more similar to how things might work in the real world. It's probably unrealistic to make code completely devoid of bugs, so I appreciated a grading test suite that accommodated this. However, it was a little frustrating that failing an edge case could push my grade from one bucket to another. Without knowing how much more I needed to fix, I sometimes was unsure if it was worth the time to debug it. (r74_BucketGeneral)

I did not like it as it obscured a lot of the details of the project and made us confused on what exactly was wrong with our project (r75_BucketGeneral)

It was good that after getting a certain grade that we were guaranteed a certain mark. However with how the work is separated amongst partners, sometimes majority of the marks seem to depend on the other partner and it ended up being very frustrating for one of the partners who finished their work flawlessly and still have to depend on their other partner to hand in their project on time and done to receive a decent mark. (r76_BucketGeneral)

I think it was nice, since I didn't have to worry about getting 100% of edge cases
(r77_BucketGeneral)

Again, there were times where the grading was confusing as it was not specific enough or misleading due to criteria dependencies. I had to go to TAs at least once per checkpoint to clarify those issues. (r78_BucketGeneral)

I liked that you didn't have to worry about making it perfect. I enjoyed the bucket grading.
(r79_BucketGeneral)

I liked how you didn't have to get the implementation completely perfect to get 100% however sometimes it felt like you were stuck on one bucket for a long time and even with changes, the bucket wouldn't change and it was unclear if it was worth working on it more to attempt to get out of the bucket or not. (r80_BucketGeneral)

positives: feels nice and feels that we're progressing
negative+confusion: doesn't have a clear clue on how much we're getting
(r81_BucketGeneral)

I think it was pretty easy to understand
(r82_BucketGeneral)

I thought it worked well. I was always trying to get Proficient/Extending, and it was really nice to not have to even think about 85 vs 90 vs 100.
(r83_BucketGeneral)

at the start but things made sense near the end.
(r84_BucketGeneral)

I don't like the bucket grading because it only benefits the proficient bucket. For C1, I got ~86% according to my TA but because of the bucket grading it would round down my grade.
(r85_BucketGeneral)

I don't really feel any particular way about it. I would probably rather have a percentage, since it is easier to see how close you are to the final

solution. It can be a bit confusing when you make a lot of changes to a program, and fix a lot of bugs, but end up with the same grade - it can cause you to doubt whether you actually fixed any bugs. (r86_BucketGeneral)

I think if we were able to see a grade computed by the number of tests passed, it would be really too stressful as any change would change the grade, so on that note, I do like the concept of bucket gradings. However, I feel that linear buckets would be better, with each being 20%, or maybe assign a letter A-, A, A+ for example. (r87_BucketGeneral)

Not really any issues I think it was fine
(r88_BucketGeneral)

Positives, I guess it was nice to see your grade move up the ladder. Negatives, adding unnecessary complexity, just give a number grade.
(r89_BucketGeneral)

I like it. One positive is that you don't need absolute perfection to get 100%, which I appreciate. (r90_BucketGeneral)

I didn't really like it, as it didn't help me pinpoint what bugs I had. I know that means I need more local tests, but I thought my tests were thorough enough, so I don't even know what other cases to test to catch the bugs, so it caused me to give up sometimes. (r91_BucketGeneral)

I have mentioned in the previous part. Where we were 0/1 for SUM but have no idea about how to fix it. (r92_BucketGeneral)

I think it made it easier for us, but I would really like TAs, instructors, slides and website to tell the same story. (r93_BucketGeneral)

I found the bucket grading system to be useful in figuring out which part of our project was failing.
(r94_BucketGeneral)

It reduces Anxiety a lot from my perspective, and somehow it encourages me more.
(r95_BucketGeneral)

I loved this grading system. I feel that it really took a lot of the stress away from developing the project, as you weren't always stressed trying to get everything exactly perfect so that you get 100%. There are a lot of small things that could maybe go wrong, but don't necessarily impact the core functionality of the program, and I feel that bucket grading takes this into consideration. If you have a program that works, and follows the spec requirements, then you get the maximum grade, which I think is incredibly fair to everyone. (r96_BucketGeneral)

I think straight percentages (or "proportion of tests passed") is probably more helpful than buckets, although I like the idea of getting 100% on a checkpoint even with some tests still failing. (r97_BucketGeneral)

It was slightly confusing to figure out if we were at the higher end of the bucket or the lower end and how much work needed to be done before we move to the next bucket level. (r98_BucketGeneral)

It could potentially be helpful for those who are somewhat behind, as they won't necessarily get a failing grade for mistakes that could cause them to fail many of the automated tests. I think it makes it less stressful for those who are 'behind' on the project by giving a guarantee of sorts for a grade rather than a 0. (r99_BucketGeneral)

Positive was that it allowed an imperfect, but otherwise complete implementation to achieve proficient. Negative perhaps was that the bucket label names didn't have much effect at least for me. (r100_BucketGeneral)

Positives: Can get 100% grade without perfection.
Cons: You can have no idea what your exact % will be which kinda sucks if you are like aiming to get like a 75% and that's all you want, but you don't know. Like maybe you are at risk of failing and you just need to get a 75% on this checkpoint and ur fine, but the buckets make it impossible to tell. (r101_BucketGeneral)

It helped to not force yourself to try and get 100%, as long as you get proficient it is good enough. Saves a lot of unnecessary time and energy to get the last couple of percentages (r102_BucketGeneral)

I just ignored this system. (r103_BucketGeneral)

I liked the bucket grading, as we didn't have to worry about marks being taken off here and there for little mistakes (r104_BucketGeneral)

Knowing how close I was to the next bucket would have been helpful but having the bucket system also helped in time organization for the course as if I reached my target bucket then I would not try to keep adding more to get a few percent marks higher. (r105_BucketGeneral)

Sometimes I get all proficient for all the points, and one acquiring, and my overall grade is capped at developing, and it's confusing because I don't know why. And putting it into buckets sometimes make me wonder if my grade is like 90%, will it be always capped at 80% because it's not 100% kind of thing (r106_BucketGeneral)

can make it difficult to gauge progress but was good to not solely focus on getting everything perfect (r107_BucketGeneral)

I think the labelling has been very good and clear. Before shown the label (i.e. before implementing) we focused on making sure the whole program works well. Then after shown the label, we also get feedback on a precise grade, which is helpful. (r108_BucketGeneral)

We thought initially that it was a baseline, ex. Developing meant at least 80%. But since it's bucketed it made us waste a lot of time trying to fix small issues so we could try to get 90% (r109_BucketGeneral)

I thought it was fine, didn't have any feelings towards it in any way (r110_BucketGeneral)

I think how the grade was actually calculated was confusing, but I don't mind the bucket grading system. I didn't really need to know what percent I got, I just wanted to know what it meant for the project (like if I had to do more work)
(r111_BucketGeneral)

I think it is good, since I don't have to meet obscure edge cases to get 100
(r112_BucketGeneral)

Some of the instructions gave the impression that Extending was reachable through autotest alone, causing some confusion. (r113_BucketGeneral)

Smoke tests passing but not having proficient was confusing (r114_BucketGeneral)

I was frustrating to put in all this effort and see more aspects of our implementation working but still remain in the same bucket. It would have been nice to see even a small change in our grade to make us feel like we were still making progress.
(r115_BucketGeneral)

No complaints. (r116_BucketGeneral)

Seems like a un-needed level of abstraction. Students care about percentages.
(r117_BucketGeneral)

No strong opinions, but I liked that there was some leniency (being able to get full marks despite having a few bugs remaining).
(r118_BucketGeneral)

Bucket grading is cool (r119_BucketGeneral)

It was just demotivating to see our grade not move one bit until we hit the higher bracket.
(r120_BucketGeneral)

The positives were that I wasn't so fixated on getting a certain percentage like the first checkpoint. A negative is the other side of things, where since the scoring is general it's hard to see if I should continue to improve the code. Additionally, I was confused with how the bucket grading correlated with grade percentages.
(r121_BucketGeneral)

I don't have many interactions with bucket grading, I usually got Proficient when my code finished (r122_BucketGeneral)

The bucket grading system is a huge positive, as it provides feedback to students about where they are at in a project. The specs are a little bit of a mess at times, as they can be unclear and potentially contradictory. One of the simplest examples of this is the 1900 rule for C1 where the year is to be set to 1900 if the field Section = "overall". The field Section is not listed in the list of required fields, so there was quite a lot of confusion on whether we should check for that. This question was also asked but not properly addressed on Piazza, which happened quite a few times with other issues. In the end, we usually made an assumption and moved on with the implementation, which is far from ideal.
(r123_BucketGeneral)

Positive as we would get full marks even if we had small hiccups in our code. Saved us a lot of time when we didn't have to track super minor mistakes (r124_BucketGeneral)

i hate being graded on a black box, no way of checking what im wrong (r125_BucketGeneral)

The autotester has some generally great feedback but was not particularly helpful or sometimes was not correct in pointing out some issues with our code. (r126_BucketGeneral)

Sometimes it is really confusing. I will suggest just put the percentage grade. (r127_BucketGeneral)

It was extremely frustrating and demotivating. We would be at developing without much feedback to know what to improve on, then spend a whole week improving our tests to end up with the exact same grade. (r128_BucketGeneral)

negative (r129_BucketGeneral)

I really like this idea. This makes things so much less stressful and helped me learn in a more healthy manner (if that makes sense) (r130_BucketGeneral)

I can see how it may be quite unforgiving (particular with the checkpoint 3), even if you have a good amount of implementation completed, but I think it's overall well fleshed out. (r131_BucketGeneral)

Overall, it was good, but like mentioned I think it should give more detailed feedback and also when a checkmark is put, it means its proficient in that section, rather than giving checkmarks for everything and then still not being proficient (r132_BucketGeneral)

I loved it; it saved us a lot of time not worry about a super small edge case that otherwise would have taken us hours to figure out. (r133_BucketGeneral)

I liked that we didn't need a perfect implementation for full marks, but I can imagine that having student A pass only one more test than student B, but student A getting a 20% higher grade potentially being a bit unfair. (r134_BucketGeneral)

while we may have gotten proficient and moved on with our lives we still have no idea whats wrong with our sort. (r135_BucketGeneral)

I am ok with bucket grading but felt the feedback was too vague (r136_BucketGeneral)

overall neutral. i usually interpret grades in percent form and each bucket had an associated percent equivalent so it was not a major change. (r137_BucketGeneral)

it was okey (r138_BucketGeneral)

Not really sure why a percent system could not have been used instead. (r139_BucketGeneral)

I did not care about the system, just tell me if I get a 100% or not. (r140_BucketGeneral)

I don't like how we could be at almost proficient, essentially completing the checkpoint - maybe like 90% - and still only given 80% if we're in the developing bucket. (r141_BucketGeneral)

It is confusing. Needing to memorize new words to figure out what the grade even means (even though it's basically just a percentage grade) does seem kind of unnecessary. If you're going to make grades up at least use something fun, like Outstanding, Exceeds Expectations, Acceptable, Poor, Dreadful, and Troll. (r142_BucketGeneral)

I still don't understand how how it works lmao (r143_BucketGeneral)

Negatives, I thought that the feedback was unclear and could be improved. (r144_BucketGeneral)

I think knowing exactly where we were was hard. For example, we could submit a check and it return acquiring, then work on the project and implement a lot of changes and features, check it again and still have it returning. I think bucket scale is fine but maybe a percentage system would be better as we could tell if our new additions were actually working or not (r145_BucketGeneral)

It has its pros and cons compared to the point / percent scale. Its feedback is not as clear as the latter, but it reduces some of my anxiety. When I see the label "proficient" I tend to stop working, but if there is some scores like 95%, I might be uncomfortable about it since it's not full marks. (r146_BucketGeneral)

not really helpful but not harmful either (r147_BucketGeneral)

At most times it confused me. When I spent a lot of time fixing a bug but it turns out I'm still in the "beginning" level, you may imagine how struggle I was. I hope it could be more specific and gives details. (r148_BucketGeneral)

It was fine, but I always just used it to convert to percent. I really only cared for percent. Although seeing "acquiring" or "developing" over and over got to be a little frustrating. (r149_BucketGeneral)

The positive side was that it gave us a vague estimation of our progress on our tasks; it offered grades with a moderate level of tolerance so that we did not have to spend too much effort on ensuring one or two missing aspects were covered. The possibly negative side was that we might actually miss some key aspects of the stage we were working on, and we could not make sure if these deficiencies would affect the next stage's behavior. It somewhat transferred a certain level of burden to the next stage. (r150_BucketGeneral)

Might confuse u because not as precise as % grade (r151_BucketGeneral)

Confusion occurs when there's no error in local, but auto tests fail (happened to MIN/MAX) (r152_BucketGeneral)

It is pretty clear (r153_BucketGeneral)

I like how I didn't need to do 100% of the tasks to get 100% :D (r154_BucketGeneral)

Definitely a good idea - I would have spent a lot of time trying to get all of the tests in C0 if they weren't bucket graded. (r155_BucketGeneral)

I think I had a positive interaction, and I appreciate it being implemented because it encourages students to spend time creating quality code, rather than spending a lot of time over small edge cases that don't contribute to their achievement of the learning objectives. (r156_BucketGeneral)

The bucket grading does not really help us to improve our code or to check which stage we are at. A simple percentage grade would have sufficed. (r157_BucketGeneral)

sometimes unfair cuz maybe my score is pretty close to proficient, but I still only get the points of developing bucket. (r158_BucketGeneral)

I think it might not be the fairest way to allocate grades. I think it allowed for some incomplete or incorrect submissions to be given full marks. (r159_BucketGeneral)

As previously mentioned (r160_BucketGeneral)

The comments within the bucket grading are oftentimes so vague. I would not know which area I should be looking more into and/or why my grade is in a particular grading bucket. Sometimes, adding more valid (and passing) tests would lower my grade in the grading buckets. But overall, it provided me with the motivation I need to know where I am in a grading criteria. (r161_BucketGeneral)

It was quite helpful overall for initial implementations, but could be seen as annoying when we were "right on the border" of buckets, seemingly. However, it was a lot nicer than how brute forcing missing one test (ex. 63/64 passing) would be. (r162_BucketGeneral)

positives: it motivated me to keep going.
Negatives: i felt disheartened when i was stuck in a bucket for too long after implementing my code, passing tests locally, but getting a low grade from the autobot and not getting any helpful feedback (r163_BucketGeneral)

Don't like the system. (r164_BucketGeneral)

Didn't really affect me much either way (r165_BucketGeneral)

It should be easier to understand the actual percentage grade. (r166_BucketGeneral)

Not really, preferable if we had % but it is what it is (r167_BucketGeneral)

It is relatively positive. I'm glad we don't need to solve all the problems with our codes to get 100%. :) (r168_BucketGeneral)

I think it was a mostly positive experience, however I liked CPSC317's system better where it was still 0-100 scale but weighted in a way where if you go most of the features you were around 70-80% and the 90-100% was only if you wanted to perfect it in a sense. (r169_BucketGeneral)

We had times where we got proficient in all except 1 and still received overall developing, so we didn't think that was fair. (r170_BucketGeneral)

I thought that the bucket grading system was better than percentage-based system for this project. (r171_BucketGeneral)

I actually enjoyed it a lot tbh. It took stress off of seeing the numbers not being 100 for example, and let me feel a little more sane. Some confusions was when it wasn't clear how to make it to the next bucket when the smoke tests seemed to indicate that we were doing well? It also felt kinda harsh to be grouped in a bucket that felt a little lower to what I felt we accomplished. (r172_BucketGeneral)

No confusion. Just had to 'spend time' searching for the conversion to percentages. Also, the criteria to declare something to be 'developing', 'proficient', 'acquiring' etc was vague, especially when the portions/tasks that were being graded seemed quite simple. Like in some cases, a straight percentage based on the number of tasks completed made more sense, and it seemed that this bucketing system was moreso pushed/forced through to the course, although the idea 'sounds good' in terms of motivating students, but also in terms of 'sounding fancy' (just like people in academia/computer science have the tendency to do so). \\\ feel these buckets would be more subjective, rather than just being determined by the 'number of checkboxes' being checked by some lines of code. Like I said, it sounds fancy, and seems appropriate, but it actually currently just seems like a fancy way to communicate the grade by just covering the number of tasks done out of the total tasks (the percentage) with a name. (r173_BucketGeneral)

I wasn't the biggest fan, I would have preferred just a percentage (r174_BucketGeneral)

Minor issue, but the Extending category was never shown anywhere explicitly so I didn't feel anything special about it. (r175_BucketGeneral)

I think it's good to use the bucket system, so students can get 100% even when a few cases were not covered (r176_BucketGeneral)

Maybe it can show more about where it gets a full mark where it gets no mark to give a better sense of which area to be more focus on. (r177_BucketGeneral)

really liked it... allowed us to account for some mistakes while still getting overall functionality (r178_BucketGeneral)

I thought the system was great and helped us know when it was time to stop implementing or else we would just keep going right until the deadline. (r179_BucketGeneral)

it was good, helped give a high level assessment of our progress without having to calculate exactly what our progress was (r180_BucketGeneral)

Some frustration when trying to move up a bucket grade between acquiring and developing if I could tell that we were super close to making it to the developing bucket which would come with a decent grade boost (r181_BucketGeneral)

It helped better understand our progress throughout the checkpoint. It was a bit confusing to convert to a percent grade. (r182_BucketGeneral)

There weren't any confusions as it was well explained. It didn't affect my approach to the project since I was going for full marks. (r183_BucketGeneral)

It was confusing because my partner and I got a 1/1 for most things on c1 but we got a bad grade because of the confusion with the bucketed grading system. (r184_BucketGeneral)

I felt like bucket grading was a fair and easily interpretable grading scheme (r185_BucketGeneral)

Definitely positive. Saved lots of debugging time as we did not have to get 100% exactly. (r186_BucketGeneral)

I really liked it suprisingly, it helped me focus less on the grade and more on larger progress mechanisms. (r187_BucketGeneral)

From my understanding, with bucket grading, you could reach proficient even if you didn't pass all tests. I think this is detrimental because students won't be able to pinpoint what exactly they're missing (r188_BucketGeneral)

I think overall is good. (r189_BucketGeneral)

I think it's more helpful to have a percentage, and just give full marks if u get above 80%, for example (r190_BucketGeneral)

At first I was slightly confused about what percentages each of acquiring, developing, etc. corresponded to. But I understood that after a short while. (r191_BucketGeneral)

most time positives I would say, sometime confusions, like I check marked all the feature but still in Developing sometime. (r192_BucketGeneral)

its nice to get a decent grade quite fast, but can be demotivating to get a 100% since the effort required to fix all the other bugs can sometimes feel not worth it. (r193_BucketGeneral)

I did not mind the bucket grading. It would have been nice if I could get an actual %, but it did not effect the project for me. (r194_BucketGeneral)

Negative: not enough detail for error discourage the usage as a tool for feedbacks (r195_BucketGeneral)

None really, I think the explanation on the site made sense to me. I liked the margin for error we were given, it really helped get me to work and made me feel significantly less frustrated than in other classes where you have to get everything correct. (r196_BucketGeneral)

It's effective. (r197_BucketGeneral)

Loved it.. doesn't make me want to focus on just getting a 100 but to improve other stuff too (r198_BucketGeneral)

Some negative because if you are just missing one small thing you lose a lot. (r199_BucketGeneral)

Sometimes we ran out of ideas but still didn't know anything else we could improve on the not-proficient cluster. (r200_BucketGeneral)

Negative. Developing is 80, but Proficient is 100. This is a big gap, the percentage score based on the percentage of passed tests will be fairer. (r201_BucketGeneral)

When we went from Beginning to Acquiring, it was the best thing that happened to me that day. After acquiring, everything normally comes out pretty fast and decent. (r202_BucketGeneral)

It was pretty good I think. (r203_BucketGeneral)

i think it was good as it would have been impossible to get 100% of the tests passing where as with the bucket grading you would get full marks as soon as you reached proficient (r204_BucketGeneral)

I would rather have more details about my grade than just be put in a bucket (r205_BucketGeneral)

Good! (r206_BucketGeneral)

It was the same as giving a point scale (r207_BucketGeneral)

It's nice that we dont have to get everything to achieve proficient, and the test feedbacks do in most cases help us move into the next bucket which is awesome. (r208_BucketGeneral)

The only issue I faced would be limited feedback. There were random issues, for example, we got a grade for one of the features and reached Acquiring. We then added a new feature to see if it works and got it graded, without making any changes to the tests. On running yarn cover, the coverage was well over 80% and yet AutoTest failed to submit a feedback, showing coverage less than threshold. On asking the TA, no help was provided since the TA also couldn't understand the source of the error. (r209_BucketGeneral)

Getting stuck in the bucket fora while is very annoying, but it is nice that you can get 100% and still have an issue with one edge case. (r210_BucketGeneral)

I think overall the bucket system has merit. I think at first it's a little confusing, especially since every other course uses a grade based system. But I do admire the motivation behind it. I like the gradual shift from core functionality to complex functionality to edge cases and so on, and I believe it reflects how software is built in the real world. So I think it should still be used. (r211_BucketGeneral)

I didn't really care (r212_BucketGeneral)

personally did not like, very vague, more used to a point system. Seemed like it was easier to get 100% though so I am not complaining :) (r213_BucketGeneral)

I really didn't like the idea that someone who worked on the project for maybe 5 hours could get to developing, and someone who worked on the project for 25 hours and is a couple tests away from proficient would get the same mark at the end. It feels odd not being rewarded for getting more tests and overall I'm not sure I liked the bucket grading system. It was frustrating and confusing at times, and although I know in the real world there is no autograder, I think there are more effective ways to mimic real world software engineering practices. For example, not even allowing for grading to occur if you don't have a certain amount of coverage, or placing a bigger focus on writing tests. (r214_BucketGeneral)

I think my overall impression is positive. (r215_BucketGeneral)

To me, it was neutral, but I believe it has positive benefits of being more forgiving than pure percentage grading! (r216_BucketGeneral)

The bucket scale was a bit vague on how exactly grades are assigned but otherwise complete serviceable. (r217_BucketGeneral)

It was fine. I didn't really think much of it. It made it harder than just seeing which tests were passing/failing (r218_BucketGeneral)

I think it was nice that it didn't need to be absolute perfect to get proficient, but also there were some parts that we may have been able to work on more to improve extra if we knew to fix it earlier (r219_BucketGeneral)

Only what I have previously stated. (r220_BucketGeneral)

The bucket grading system provided structure and clarity, but there were moments of uncertainty, especially regarding reaching 100% proficiency within the Proficient level. However, it did encourage me as the bucket grading shows level goes up. (r221_BucketGeneral)

Didn't care much for the other labels, only cared to try our best to get the highest grade (r222_BucketGeneral)

didn't like it buckets were too wide (r223_BucketGeneral)

I think it was good since it didn't put pressure on me to try and get as many points out of 100 as possible and spend too much of my time on small optimizations. (r224_BucketGeneral)

The bucket grading worked fine, I had no issues with it and it provided a nice compartmentalization of the grading scheme. (r225_BucketGeneral)

Auto test feedback is ambiguous and often lead us to the wrong place to look. (r226_BucketGeneral)

I like bucket grading in a way that even if we implemented some of the work, we still got rewarded for that part. (r227_BucketGeneral)

Negatives: wait times for feedback, lack of specific feedback, counter-intuitive grading (why not just tell students the percentage of tests failing in each bucket?)\\Positives: the categories of tests are nice (r228_BucketGeneral)

It was nice because it simplified the grading system and made it clear where we were going wrong. (r229_BucketGeneral)

I prefer actual grade point tbh (r230_BucketGeneral)

I had overall positive interactions with the autograding and the bucket system (r231_BucketGeneral)

I like it better than percent (r232_BucketGeneral)

I personally would've preferred a percentage; it seems a lot simpler and easy to understand. (r233_BucketGeneral)

Confusion on how buckets are calculated from test performance. (r234_BucketGeneral)

It was OK to not have the exact percentage mark in my opinion, because we should be doing our best to raise the mark to higher groups or percentages anyways, and we only need to know where we are roughly. However, I can see how some students would disagree if they need a specific mark for something. (r235_BucketGeneral)

Not many feelings, I thought it was pretty cool overall. (r236_BucketGeneral)

positive: give lower bound of grades, negative: can't really know the exact grade we get (r237_BucketGeneral)

It's okay except for some confusion all is good. (r238_BucketGeneral)

I do like how it stopped us from spending a ridiculous amount of time chasing a few potentially missing percentage points at the end of a checkpoint. I also like how it was more of a progress report than a grade. (r239_BucketGeneral)

positive that even if not all are passing, you get a decent grade. But negative that we don't know the exact percentages the bucket corresponds to so kinda frustrating at times. (r240_BucketGeneral)

There were some confusions as most of the time we would pass all of the smoke tests but not reach proficient yet, so didn't really know what was missing. (r241_BucketGeneral)

The bucket grading removes the stress from the numbers. (r242_BucketGeneral)

I found it somewhat punishing for small mistakes. (r243_BucketGeneral)

sucks when i have beginning and do lots of work and get more passes on smoke clusters but dont gain the next bucket (r244_BucketGeneral)

a little bit of confusion (r245_BucketGeneral)

It was mostly neutral, it was confusing at first, but it helped simplify reading the feedback on the checkpoints. It was also a good thing that percentage grades were not given as feedback for c1, c2, & c3, since I feel like the exact percentages would distract me, and draw too much attention to the slowly incrementing grade as I submitted my work. \One draw back of the bucket grading would be that it makes the work progress seem stagnant at times, when you submit a grade and still see the grade to be in the same bucket despite doing so much more work can be a bit frustrating, but otherwise there are always was to convert it to a percentage if needed. (r246_BucketGeneral)

I had no confusions. I believe the bucket grading system is nice for teams who perform well as it gives a solid stopping point when the proficient bucket is reached. However, for teams struggling, it can be difficult to reach the next bucket. (r247_BucketGeneral)

It was honestly great. I might have liked percentages listed next to the bucket name, but that might be counter to the goals of bucket grading (r252_BucketGeneral)

I personally find it a bit confusing since the Beginning bucket has a much wider range (0-55) than the other buckets, so it's a bit difficult to estimate my progress. I find bucket grading more useful for non-code submissions as the description and rubric is clearly stated. (r248_BucketGeneral)

The wording might be too vague. A percentage range would be more helpful. (r249_BucketGeneral)

Overall, I was very happy about this system, the feedbacks are very useful and precise. (r250_BucketGeneral)

It was initially confusing as I thought the grade would have smaller increments based on progress, but I got used to it after. I would like smaller increments though because sometimes one small issue prevents the grade jump from 80% to 100%. (r251_BucketGeneral)

It is a good system. However, i feel like giving a % makes me more certain about how i am doing, and whether this and the previous commit i made, improved or decreased by grade. (r253_BucketGeneral)

I thought the bucket grading was good, it allowed me to know what to be focused on. (r254_BucketGeneral)

Very neutral about it. It's neither a particularly good nor particularly bad system. (r255_BucketGeneral)

I have no complaints nor feedback really. The bucket grading system felt easily convertible to a point or percent system so I just interpreted it as is (r256_BucketGeneral)

They were confusing as at first I though of them as 20, 40, 60, 80 , and 100 percent, when in reality I feel like it was more like 50, 60, 70, 80 ,90+ percent (r257_BucketGeneral)

I think it is actually good since it reflects the industry well (r258_BucketGeneral)

I like the bucket grade at the end - i.e. developing is better than a 68, but while trying to improve the score, a more specific grade is more useful. (r259_BucketGeneral)

There was an extreme wait time at the end (r260_BucketGeneral)

I really don't think it matters to have percentage grade, or bucket grade. The aim should always be high on projects. (r261_BucketGeneral)