

REVIEW FEEDBACK

Josh Blumberg 12/03

12 March 2020 / 11:00 AM / Reviewer:

Steady – You credibly demonstrated this in the session.

Improving – You did not credibly demonstrate this yet.

GENERAL FEEDBACK

Feedback: Thanks for a really good review. We had a small blip with internet connectivity near the beginning, but quickly reconnected. Overall you are pretty solid against the course goals – well done. You are very user focussed and showed great consideration to the client/reviewer – thank you. This came across very well. You are extremely methodical, have a good debugging process and a good TDD process. My only real comment is that possibly you could have started with an even simpler case. While this may seem silly for such a simple task, it is good practice for bigger and more complex systems. But otherwise excellent review, well done!

I CAN TDD ANYTHING – Steady

Feedback: I think that you could still work on starting with a simpler test, such as an input with a single value only. The first/second tests were a bit complex because they required that you introduced both if statements for checking the bounds as well as iteration over the array. While this is a very simple task, which you seemed to almost instantly see how to approach, it is good practice for bigger and more complex systems.

Other than this, you seem to understand TDD and its value. I really liked that you adjusted your approach when you realised that your tests were not going to drive the implementation forward. You laid out nice tests laid out in the readme. A couple of these may be unnecessary ones, duplicate testing features, but your justifications for them were excellent and they provided good coverage.

I CAN PROGRAM FLUENTLY – Steady

Feedback: You seemed quite comfortable navigating your terminal and editor while setting up your environment. You demonstrated knowledge of both core syntax and more advanced syntax such as for the map function, which was an excellent choice. I was also pleased to see that you made use of the Ruby docs in addition to other internet resources. You seemed quite comfortable with programming.

I CAN DEBUG ANYTHING – Steady

Feedback:

I CAN MODEL ANYTHING – Steady

Feedback: You modelled your solution in a single method which I felt was a nice and simple implementation and provided a good place to start.

I CAN REFACTOR ANYTHING –Improving

Feedback: There are some issues with your Red-Green-Refactor cycle, where you tend to try to find a *solid*(refactored) solution to the currently failing test instead of finding the quickest solution during the Green step – this is a common problem with the strict TDD process amongst stronger programmers and this sort of methodology tends to produce over-engineered solutions and a fair amount of debugging. Try to find the simplest and quickest solution during the Green step and then use the Refactoring step to solidify/neaten/generalise the solution. This will be crucial in your next review.

I HAVE A METHODICAL APPROACH TO SOLVING PROBLEMS – Steady

Feedback: I was really impressed with the methodicalness of your whole approach. Your information gathering process demonstrated this nicely, and you made lovely notes and ticked off features in your IO table when they were

completed. I liked that you used two panes during development so that you would refer back and forth between them. Your TDD process moved along nicely, punctuated with git commits. I was very pleased to see that you checked your logic before running Rspec. My only comment here is that to be even more methodical, your tests should start off even simpler so that your implementation can also start off really simply (no iterating at the beginning) and that you should try to take the simplest approach to get your tests to pass first, and then follow this with a distinct refactor step.

I USE AN AGILE DEVELOPMENT PROCESS – Strong

Feedback: You have developed a really nice and methodical information gathering process – well done! You asked questions about the input and output type and about formatting, established the main program function and considered several edge cases. You developed some good expected input and output cases to test later, though you could consider some examples that are even simpler. You had excellent client interaction throughout the information gathering session and showed consideration for the client. Perhaps if you could go through this process a little faster, you would have more time for programming, but your process is really excellent here, so don't compromise on it too much.

I WRITE CODE THAT IS EASY TO CHANGE – Steady

Feedback: I was pleased that you had your test suite properly decoupled from your implementation by making sure the tests were based solely on acceptance criteria, and not reliant on the current implementation. This makes changes to the code much easier.

I was very happy with your Git usage. You initialised Git early on, committed after every test went green and supplied useful commit messages, making it easy for future developers to come in and change a feature.

I CAN JUSTIFY THE WAY I WORK – Strong

Feedback: You had very good user interaction and vocalisation throughout the review. You were very user-focussed, which was nice, let me know what to expect, why errors were caused and provided good justifications for decisions.