REVIEW FEEDBACK

Kenny Kehinde Peter Olofinmoyin 12/12

12 December 2019 / 09:00 AM / Reviewer: Robert Murdoch

Steady – You credibly demonstrated this in the session.
Improving – You did not credibly demonstrate this yet.

GENERAL FEEDBACK

Feedback: Overall this was a great improvement over the last session there are only a few things that still need to be ironed out namely your refactoring process which is benefited by simpler test progressions and more regular refactors for generalisation, go ahead and practice these over the year end break. Unfortunately the recording bugged out and I was only able to record the last few minutes of the session but I have uploaded it anyway.

I CAN TDD ANYTHING – Improving

Feedback: Although your test progression was a bit off the test themselves were based on acceptance criteria that you had developed, overall you tend to write meaningful tests and definitely focus on a test first approach in your dev cycle.

I CAN PROGRAM FLUENTLY – Steady

Feedback: You seemed quite comfortable navigating your terminal while setting up your environment. I feel that you have a good understanding of basic and intermediate Ruby syntax and generally seem comfortable in the language. I liked your use of if statement to separate out different cases that you programme would function under.

I CAN DEBUG ANYTHING – Nothing here

Feedback:

I CAN MODEL ANYTHING – Steady

Feedback: You modeled your solution in a single method which I felt was a nice and simple implementation, which provided a good place to start for this type of a programme.

I CAN REFACTOR ANYTHING –Improving

Feedback: This was the main issue with your process as everything else was quite serviceable, what you want to be doing with refactors is taking certain piece of code and replace/generalise it with some other piece of code , the main rule with refactoring is that we should never start a refactor when any tests are failing, because we always want to start a refactor from a space where the implementation is fulfilling its contracts with the test suite. When refactoring we are not so much focused on changing the logic as we are on changing the way that the logic operates to make the code objectively better (readability, efficient data structure, good algorithms, non repeating code) Refactoring is a beast on its own that takes time to master, but I just want to see that you have it integrated into your development cycle. Try and look for refactors every time you hit green across all tests, whether because of a simple passing test or because of a previous refactor that you completed.

I HAVE A METHODICAL APPROACH TO SOLVING PROBLEMS – Steady

Feedback: Overall you seem to have a pretty good methodical process during the session where you were prioritizing tasks that provided immediate value the user. I liked that you were prioritizing acceptance criteria over edge cases. You worked through tasks in a logical manner starting in the most simple of spaces and moving towards more complex spaces, however there were w few jumps in complexity which might have made refactors less apparent.

I USE AN AGILE DEVELOPMENT PROCESS – Steady

Feedback: Your requirements gathering was very good and comprehensive you asked many questions about the input and output of the programme, as well as the few edge cases that the programme was meant to handle. I liked that you provided a comprehensive list of inputs and outputs however you did miss a few of the very important simpler cases. when writing out acceptance criteria the simplest cases are the most important as they (through TDD) determine the core functionality of your programme.

I WRITE CODE THAT IS EASY TO CHANGE – Steady

Feedback: Overall you write code that is easy to change and maintain, you had your test suite properly decoupled from your implementation by making sure the tests are based solely on behaviors this makes changes to the code much easier.

I CAN JUSTIFY THE WAY I WORK – Steady

Feedback: Throughout the session, you were vocalizing your thought process, as a result I was able to clearly follow you through the session. You seemed to be able to justify your decisions for doing things. You were regularly updating the product owner on your progress through the task.