**Week 1 Goal Plan**

**Plan**

TDD

1. **Improve breadth and depth of knowledge about TDD**

I have improved knowledge of TDD through [**The 4 phases of a test**](https://robots.thoughtbot.com/four-phase-test)

I have improved knowledge of TDD through [**The three rules of TDD**](https://gist.github.com/R4wm/a0ea5cba323d945625141e0c98238c64)

I have improved knowledge of TDD through [**Testing Behaviour**](https://github.com/makersacademy/skills-workshops/blob/main/practicals/testing/behaviour_not_state.md)

1. **Produce unit tests that are useful, precise, and fit for purpose**

I have practised unit testing through [**Test Driving Exercises**](https://github.com/makersacademy/skills-workshops/blob/main/practicals/test_driving.md)

I have practised unit testing through [**Test Driving Practice**](https://github.com/makersacademy/birthdays)

I have practised unit testing through **Dice Rolling** exercise

I have practised unit testing through the [**Boris Bikes**](https://github.com/makersacademy/course/tree/main/boris_bikes)challenge

I understand the difference between ‘require’ and ‘require\_relative’

I have requested a **code review** on at least one of the above and noted FB

1. **Become more familiar with Rspec matchers**

I have researched and used at 10 different matchers in my work [**Rspec Matchers**](https://relishapp.com/rspec/rspec-expectations/docs/built-in-matchers)

Pair Programming

1. Let partner know what I need to pair program effectively

I have asked my partner for 5 mins to digest instruction material for each step

I have asked my partner for time to check any hints and additional material

I have agreed the timing/pattern for switching roles with my partner

1. Ask partner what they need to pair program effectively

I have asked about and noted anything my partner needs from me to make our pairing more effective

1. Becoming a more active driver and navigator

Driver - I say what I am doing and **why**

Navigator - I research peripheral objectives & troubleshoot

New Concepts

1. Understand the basics of encapsulation and SRP

I have googled encapsulation and SRP

I have read about encapsulation on Wikipedia

I have learned about [**Encapsulation & Cohesion**](https://en.wikipedia.org/wiki/Encapsulation_%28computer_programming%29) with guided resources (if time)

JavaScript

1. Maintain & build knowledge of JS syntax & technologies

I have completed 1 JS katas (<=6) on [**codewars**](https://www.codewars.com/dashboard)

I have completed the APIs module on [**codecademy**](https://www.codecademy.com/learn) Fullstack course (if time)

Ruby

1. Maintain & build knowledge of ruby syntax & technologies

I have completed 1 Ruby kata (<=7) on [**codewars**](https://www.codewars.com/dashboard)

I have reviewed at least 2 Ruby-related knowledge pills from the [**makers pre-course**](https://makersacademy.teachable.com/courses/enrolled/256825)