**Julio Pochet – Refactoring: Cleaning Up Code for Better Performance**

**Why I Chose This Topic**

I picked **Refactoring** because it’s something every developer has to do at some point. When writing code, the first version is usually messy, repetitive, or not as efficient as it could be. Refactoring is about **cleaning up the code** without changing what it actually does. I’ve noticed that it’s easy to write long, repetitive parsing functions when working with JSON data in Python. Refactoring helps simplify that and makes the code **easier to read and maintain**.

**Summary of Main Points**

Refactoring is all about making code **cleaner, shorter, and more efficient**. Instead of having huge, confusing functions, refactoring helps break things down into **smaller, reusable pieces**. It also removes duplicate code, making everything **run smoother and faster**. One key idea from *Pragmatic Programmer* is that refactoring should be **an ongoing habit**, not just something you do when things break. Keeping the codebase clean from the start makes future updates easier and prevents bugs from piling up.

**Supporting Resource**

A great website that explains refactoring well is [Refactoring.Guru](https://refactoring.guru/refactoring). They break down different ways to clean up code with **real examples** and **before-and-after comparisons.**