### Lesson 10: Smart Pointers & Why Rust Isn’t C (Part 1)

* **Reading**: “The Rust Programming Language” Chapter 15.1-15.3 (Students may also choose to read this article from Medium: <https://perelyn-sama.medium.com/understanding-smart-pointers-in-rust-a-comprehensive-guide-cc06eb94a147> which may help summarize common Smart Pointers & their typical uses).
* **Assignments**: Read the appropriate sections of Chapter 15 & thoroughly understand (review if necessary) pointers in C
* **Preflight**: Read assigned sections in Chapter 15
* **Lesson Goals**:
  + Understand how smart pointers differ from “normal” pointers and C and references.
  + Understand how smart pointers are implemented.
  + Remember the most common standard library smart pointers and how they are used.
* **Motivation**: Programmers who come from a background in C or C++ need to understand that Rust’s concept of ownership and borrowing makes smart pointers different from “normal” pointers in C and even references in Rust.
* **Lecture**:
  + Review the way that pointers work in C
  + Cover Rust references & borrowing/ownership
  + Introduce & give examples of the Box<T> smart pointer being used in a simple program
* **Lab**: Create a Rust program to demonstrate basic knowledge of the Box<T> smart pointer. (Optional: Write a *short* description of the differences between smart pointers, pointers, and Rust references using original thought).