**Subject: Julio Pochet – Module 4.1 Discussion - Coding: Defensive and Secure.**

**Input Validation in Real Life Code**

Input validation is basically your app’s first line of defense. If we let just anything get processed—like text when numbers are expected—it could crash the program or open the door for attacks. That’s why checking what users enter before using it in your code is super important.

Here are three common input validation checks I’ve worked with:

1. **Type Check**  
   This makes sure the input is the right kind of data. If someone types “hello” when you're asking for an age, your program should catch that.

try {

int age = Integer.parseInt(userInput);

} catch (NumberFormatException e) {

System.out.println("Please enter a valid number for age.");

}

1. **Range Check**  
   This helps keep values realistic. For example, if you're asking for a rating between 1 and 5, a 10 shouldn’t be accepted.

if (rating < 1 || rating > 5) {

System.out.println("Rating must be between 1 and 5.");

}

1. **Whitelist Validation**  
   Instead of trying to block all the wrong things, whitelist validation focuses on allowing only what’s right. Like allowing only “admin” or “user” roles.

if (!role.equals("admin") && !role.equals("user")) {

System.out.println("Invalid role.");

}

Without proper validation, a user could accidentally (or intentionally) break the system. It’s not just about clean data—it’s about safety too. Validating inputs on both the frontend and backend is a habit every developer should build early on.