SE 3318 SOFTWARE CONSTRUCTION

Week 12

May 27, 2025

1 Problem description

You have joined a software development team working on a fast-growing e-commerce platform. The platform processes thousands of customer orders, user registrations, and payments daily. However, the system was quickly developed without following proper programming best practices.

The backend code has critical problems:

- It fails when given invalid or unexpected inputs.
- It lacks proper conditional logic and error handling.
- It assumes everything will always work perfectly.

Your task is to act as a code quality and safety analyst. You will review and improve a faulty part of the platform that includes:

- Order Processing
- User Validation
- Payment Handling

These features currently contain deliberate violations of conditional usage and defensive programming principles.

2 Goals

You've been given four main classes:

1. App.java – Entry point that simulates a real-world scenario.
2. OrderProcessor.java – Contains logic for processing orders and items.
3. UserValidator.java – Supposed to validate user details, but doesn't do it correctly.
4. PaymentService.java – Processes payments with assumptions that could crash the system.

You should

- Identify and document issues with current logic
- Refactor the code to properly handle invalid input
- Use clear, structured conditionals
- Apply defensive programming techniques

- Maintain clean and readable code

Submission Instructions

- Add your new project to your new branch named Week-12 Maven project.
- Commit changes to your local Git repository.
- Push your changes to the remote Week-12 branch using SourceTree.
- Use a versioned commit message like: "Add Week-12 project"