**CS 499 Journal – Module Two: Code Review**

**Part 1:**  
 Code review is a collaborative process where developers examine each other’s code to catch bugs, improve code quality, and ensure adherence to standards before merging changes into a shared repository. It’s a critical practice for computer science professionals because it helps maintain clean, maintainable, and secure codebases while fostering team knowledge sharing.

Some best practices from the resources include reviewing small chunks of code at a time, being constructive in feedback, and maintaining a clear checklist for consistency. Reviews should focus not just on syntax but also on logic, security, performance, and clarity. Code reviews should occur **before code is merged** into the main branch—typically after development and before final testing—so that problems are caught early, minimizing technical debt and reducing costly rework.

**Part 2:**  
 For recording my code review, I plan to use **Windows + G (Xbox Game Bar)**, a built-in Windows tool that allows easy screen and voice capture without the need for additional software. It’s lightweight, convenient, and produces clear recordings suitable for walkthroughs.

To stay organized, I’m creating a basic script and outline divided into three review sections based on the rubric:

1. **Software Design and Engineering:** I’ll explain how I refactored my Android app using Clean Architecture, MVVM, and Jetpack Compose, showing how this structure improves scalability and testability.
2. **Algorithms and Data Structures:** I’ll walk through the logic and performance benefits of replacing Dijkstra’s algorithm with A\* search in my pathfinding project.
3. **Databases:** I’ll discuss how I enhanced my bookstore database by normalizing it, implementing stored procedures, and creating SQL views for reporting.