**Michael King**

**CS-499**

**Journal 2-1**

**Part 1:**

1. **What is code review?**  
   Code review is a systematic process where one or more developers examine another developer's code to identify potential errors, inconsistencies, or improvements. It's a collaborative approach that fosters knowledge sharing, enhances code quality, and improves team cohesion.
2. **Why is it an important practice for computer science professionals?**  
   Code review is crucial for several reasons. It helps catch bugs and potential issues early in the development process, preventing costly problems. It also promotes knowledge sharing among team members, fostering a culture of continuous learning. Additionally, code review ensures that code adheres to coding standards and best practices, improving code quality. Finally, it strengthens team bonds and fosters a collaborative environment where developers can learn from each other and improve their skills.
3. **What are some code review best practices that you read about in the resources that are crucial to include in a code review? Include when a code review should occur in the development process with a rationale as to why.**  
   Code review best practices include focusing on clarity and readability, checking for correctness, assessing maintainability, reviewing in small chunks, providing constructive and respectful feedback, and encouraging open discussions. Code reviews should be integrated into the development process as early as possible, ideally before code is merged into the main branch. This allows for early detection of issues, prevention of defects, improved code quality, and enhanced knowledge sharing among team members.

**Part 2:**

1. **What software have you chosen to use to record your code review**?  
   I plan to use GitHub Review
2. **Describe your approach to creating an outline or writing a script for your code review for each of the three categories that you will be reviewing based on the rubric as well as the code review checklist.**I'll break down the code into smaller parts and examine each part carefully. I'll make sure the code does what it's supposed to do, runs efficiently, and is easy to understand. If I find any problems, I'll suggest ways to fix them and work with the developer to make the code better.