Hands-On Activity #7:

Cultivating the Course Garden

Objective:

To apply software maintenance and improvement techniques directly to the course repository. This activity will help you practice diagnosing code and documentation decay in a live codebase that you and your peers use regularly. You'll learn how to identify areas needing attention to make it more sustainable for current and future students.

Instructions:

1. **Observe**
   - **Identify Issues**: Look for any code, configuration files, or documentation that might have become "stale" or could benefit from updates (e.g., obsolete functions, outdated resources, unclear documentation). If you have difficulty here, consider:
     - What could be enhanced about the course to make it better than it is now?
     - What "bugs" are there in the code which misdirect or break anything?
   - **Log Observations as Issues**: Open issues for sections that could use attention. Be sure to label the issue and consider providing a user story if applicable to help describe what needs to be done (e.g. "As a <type of user> I need <x> so I can <y>.").
   - **Add labels to the issue**: Add a label to the issue to help classify how the issue falls into a specific grouping. For example, would this be an enhancement or is it a bug?
   - **Provide feedback on other issues**: look for other issues your colleagues have posted and provide feedback of some kind as a comment to help foster the discussion.

Deliverables:

GitHub Contributions:

- Open issues documenting course improvements. Include both course content (e.g., GitHub materials, slides) and delivery.

Resources:

- About GitHub issues

Tips:

- Consider how the issues might be read by someone else, potentially a long time from when the issue is submitted (what information might someone need in order to address the challenges or needs?).
- Keep in mind with communications: is it true, is it kind, is it necessary, is it helpful?