Section 1: Git Basics

1. What is the purpose of the `git pull` command?
2. Explain the purpose of `git pull --rebase` and its implications.
3. How do you unstage changes in Git, and in what scenarios is this useful?
4. Describe the function of `git commit --amend` and when you should use it. What impact does it have on commit history?

Section 2: Git Merge Strategies

1. Describe what happens to commit history during a rebase operation.
2. Differentiate between a rebase and a merge in Git.
3. What is the purpose of interactive rebasing, and in what scenarios is it valuable?

Section 3: YAML

1. Define YAML anchors and explain how they are used in YAML documents. (Extra Credit)
2. Provide examples of YAML anchors in configuration files. (Extra... Extra Credit)
3. How are multi-line strings represented in YAML, and what are the common indicators for multi-line strings?

Section 4: CI/CD Basics

1. Define Continuous Integration (CI) and explain its importance in the software development process.
2. How does CI contribute to code quality?
3. Clarify the role of Continuous Delivery (CD) in software development and discuss the difference between CI and CD.
4. What is “Immutable Infrastructure” in the context of CI/CD, and why is it relevant in modern software deployment?

Section 5: Git Commits

1. Why are clean commits important in Git?
2. Describe the characteristics of a clean commit message.

Section 6: CircleCI Basics

1. Define CircleCI and explain its role in CI/CD.
2. Enumerate the benefits of using CircleCI.
3. Differentiate between a job and a workflow in CircleCI. Explain their significance in defining CI/CD pipelines.
4. What are “context” and “environment” in CircleCI, and how are they used to manage secrets and configuration?

Section 7: Canary Deploys

1. Define what Canary deployments are and why they exist.
2. What are the primary objectives of Canary deployments in the software release process?
3. Explain the concept of a control group in Canary deployments and its role in monitoring and testing. (I know we didn’t explicitly go over this but take your best guess. It’s fairly self explanitory)