# Intermediate Version Control: Knowledge-based Questions

**Assessment Resources:**

|  |
| --- |
| Marking key available for lecturer via Blackboard.  Students may refer to the lecture material in formulating their answers.  Students may use Git Bash or Git CMD to help answer questions  You must refer to the provided North Metro Software Pty Ltd Git Onboarding Guidelines |

**Assessment Instructions:**

|  |
| --- |
| Students must attempt all questions.  Answer clearly and concisely using full sentences.  At most, two paragraphs are expected per answer.  All answers must be at the student’s own words – use of ChatGPT or other AI tools is **strictly** prohibited.  Read each question carefully before answering.  Ensure that you address all parts of each question.  If you need clarification on any question, please ask your assessor. |

## Scenario

You have recently joined North Metro Software Pty Ltd as a junior software developer. The company uses Git for version control and follows specific workflows and conventions as outlined in their [Git Onboarding Guidelines](https://github.com/NM-TAFE/ipriot-nms-org-template/tree/main/org).

Your team lead has asked you to demonstrate your understanding of the company's version control practices and how they align with industry standards.

**Ensure you have read the onboarding guidelines**

Answer **all** the questions:

### Questions

a. Compare Git, a distributed version control system to a centralized version control system you have used or heard of in the past. What are the key differences?   
b. What instructions in the onboarding guide relate most strongly to the distributed nature of git and why?

Explain how GitHub, as a web-based DevOps lifecycle tool, facilitates version control and collaboration in the software development process. In your answer, outline specific capabilities it has on top of Git that are relevant to North Metro Software.

Your team lead has asked you to set up a new repository for a project. Describe the steps you would take to create a new repository and configure it according to North Metro Software's guidelines. Include the specific Git commands you would use.

North Metro Software follows a specific branch naming convention. Explain this convention and provide three examples of correctly named branches based on the guidelines. Why are these naming conventions important for collaborative development?

You're working on a feature branch and need to merge your changes into the main branch. Describe the process you would follow, including:  
a. How you would create a pull request  
b. What you would include in the pull request description  
c. How you would handle any merge conflicts that arise  
d. Relate your answer to North Metro Software's workflow as described in the Git Onboarding Guidelines.

North Metro Software emphasizes the importance of clear, conventional commit messages. Explain the commit message format specified in the guidelines and provide two examples of well-formatted commit messages for different types of changes. Why is this standardization important?

Your colleague has made several small, related changes and is unsure whether to make multiple commits or a single commit. Advise your colleague on the best practice according to North Metro Software's guidelines, explaining the concepts of atomic commits and when it might be appropriate to squash commits.

North Metro Software uses GitHub Actions for CI/CD. Explain what CI/CD means and how it relates to version control. Describe two benefits of using automated CI/CD in the software development process.

**Assessor Guidelines**

When assessing this KBA, consider the following:

* Answers should demonstrate a clear understanding of version control concepts and practices as they apply to North Metro Software's specific context.
* Students should show they can relate their knowledge to real-world scenarios in the company.
* Correct use of Git terminology and command syntax is important.
* Answers should reflect an understanding of both the technical aspects and the collaborative nature of version control systems.

To achieve a Competent (CO) result, the student must:

1. Demonstrate a clear understanding of North Metro Software's specific practices
2. Show they can apply version control concepts to real-world scenarios

If these criteria are not met, the result is Not Yet Competent (NYC).

**Alignment with Knowledge Evidence**

This assessment addresses the following knowledge evidence requirements from the unit:

* Distributed and centralised version control systems (Q1)
* Web-based DevOps lifecycle tools (Q1, Q7)
* DevOps automation tools (Q7)
* Branching, pull and push commands local/remote (Q2, Q4)
* Repositories and working copies (Q2)
* Merging and merge tools (Q4)
* Working and indiscriminate commits (Q5, Q6)
* Resolving conflict and backout changes (Q4)
* Principles and techniques of creating repositories and branch workflows (Q2, Q3, Q4)