Version Control ASSESSMENT 1 answers

1. Describe version control. (2 marks)

Version control is a system that allows you to track changes to files over time. It is primarily used by software developers, but it can also be used by writers, artists, and other professionals who need to keep track of changes to their work.

2. Why do you need version control as a developer? (2 marks)

As a developer, you need version control to:

- **Track changes to your code:** Version control allows you to see who made changes to your code, when they were made, and what the changes were.
- **Revert to previous versions of your code:** If you make a mistake or introduce a bug, you can revert to a previous version of your code that was working.
- **Collaborate with other developers:** Version control makes it easy for multiple developers to work on the same codebase at the same time.
- **Manage your codebase:** Version control can help you organize your codebase and make it easier to find what you need.

3. Differentiate local repository from remote repository. (3 marks)

A **local repository** is a version control system that is installed on your local computer. It stores the history of changes to your code files. A **remote repository** is a version control system that is hosted on a server. It allows you to share your code with other developers and collaborate on projects.

4. Define the following terms: (5 marks)

- **Repository:** A repository is a collection of files and directories, along with their history of changes.
- **Terminal:** A terminal is a command-line interface that allows you to interact with your computer using text commands.
- **Branch:** A branch is a parallel version of a repository. It allows you to work on new features or bug fixes without affecting the main codebase.
- **Metadata:** Metadata is data about data. In version control, metadata includes information such as the author of a change, the date and time the change was made, and the commit message.
- **Commit:** A commit is a snapshot of the current state of your repository. It includes the changes you have made since the last commit.

5. In addition to keeping track of code changes, version control also provides a range of other benefits, state four (4) benefits of using any version control. (4 marks)

- **Collaboration:** Version control makes it easy for multiple developers to work on the same codebase at the same time.
- **Backup:** Version control can be used as a backup system for your code.
- **Auditability:** Version control allows you to track who made changes to your code and when they were made.
- **Experimentation:** Version control allows you to experiment with new features or bug fixes without affecting the main codebase.

6. How have version control systems transformed software development practices and collaboration? (2 marks)

Version control systems have transformed software development practices by making it easier for developers to collaborate on projects and manage their codebases. They have also made it possible to develop large and complex software projects.

7. What are some common challenges or pitfalls that developers may encounter when using version control systems? (4 marks)

- **Merge conflicts:** When two developers make changes to the same file at the same time, it can lead to merge conflicts. These can be difficult to resolve.
- **Branching and merging:** It can be difficult to manage branches and merge them back into the main codebase.
- **Learning curve:** Version control systems can have a steep learning curve.
- **Performance issues:** Large repositories can be slow to work with.

8. What are the fundamental differences between local, centralized, and distributed version control systems? (6 marks)

- **Local version control systems:** These systems store the history of changes to your code on your local computer. They are simple to set up and use, but they are not well-suited for collaboration.
- **Centralized version control systems:** These systems store the history of changes to your code on a central server. They are easy to collaborate with, but they can be a single point of failure.
- **Distributed version control systems:** These systems store the history of changes to your code on every developer's computer. They are highly resilient and well-suited for collaboration.

9. In a local version control system, how is the history of changes tracked and managed? (2 marks)

In a local version control system, the history of changes is tracked and managed using a database that is stored on your local computer.

10. What are the primary advantages and disadvantages of using a centralized version control system? provide two (2) for each. (4 marks)

Advantages:

- Easy to collaborate with.
- Single point of truth. **Disadvantages:**
- Single point of failure.
- Can be slow to work with large repositories.

11. How does a distributed version control system differ from a centralized one in terms of data storage and collaboration? (4 marks)

In a distributed version control system, the history of changes to your code is stored on every developer's computer. This makes it highly resilient and well-suited for collaboration. In a centralized version control system, the history of changes is stored on a central server. This can make it a single point of failure.

12. Differentiate working directory from staging area in Git architecture. (4 marks)

The **working directory** is where you edit your files. The **staging area** is a temporary area where you can stage changes before committing them.

13. What is the role of a remote repository in Git, and how does it facilitate collaboration among team members? (4 marks)

A remote repository is a repository that is hosted on a server. It allows you to share your code with other developers and collaborate on projects. Team members can push their changes to the remote repository and pull changes from the remote repository.

14. What is the purpose of a pull request, and how does it contribute to code quality and collaboration? (4 marks)

A pull request is a request to merge a branch into another branch. It allows you to review code changes before they are merged into the main codebase. This can help to improve code quality and prevent bugs from being introduced.

15. Let us imagine a company operating in many countries with many employees; in order to work they need to come to work in different departments. What happened during the COVID 19 pandemic? Are employees still working as usual? We all know it is not possible, because many companies were forced to close their doors because of the stay-at-home order.

- **State different 4 problems that may be raised, among them.** (4 marks)
    - **Reduced productivity:** Employees may find it difficult to be productive when working from home.
    - **Communication problems:** It can be difficult to communicate effectively with colleagues when working remotely.
    - **Technical difficulties:** Employees may experience technical difficulties when working from home.
    - **Isolation:** Employees may feel isolated when working from home.
- **How would the company's employees continue to work and be aware of the changes in their company?** (2 marks)
    - The company could use a version control system to allow employees to collaborate on projects and be aware of changes to the codebase.
- **How would they know about the changes in their institution?** (2 marks)
    - The company could use a communication tool, such as email or Slack, to keep employees informed about changes in the company.
- **How would they keep up with the past?** (2 marks)
    - The company could use a version control system to keep track of changes to the company's policies and procedures.