|  |  |  |
| --- | --- | --- |
| SCHOOL OF INFORMATION AND TECHNOLOGY | | |
| NAME: FLEX D. PEDRO | DATE PERFORMED: 11-20-24 |  |
| Section: IDC1 | DATE SUBMITTED: 11-20-24 |

# SYSADM1 – Git Basics

Answer the following research questions about Git, GitLab desktop and GitHub.

1. What is Git, and why is it important in software development?

Git is a free and open-source distributed version control system that allows developers to track changes in their code and collaborate on projects. Its importance in software development lies in its ability to manage code revisions, enabling teams to work simultaneously without overwriting each other's changes. This facilitates better collaboration, code integrity, and project management

1. How does Git track changes in a project?

Git tracks changes by creating a snapshot of the project at each commit. Each snapshot contains a reference to the previous commit, forming a chain of history. This allows developers to see what changes were made, who made them, and when, making it easy to revert to previous versions if necessary

1. What is the difference between a local repository and a remote repository in Git?

A local repository is stored on your own machine and contains all the project files and history. In contrast, a remote repository is hosted on a server (like GitHub or GitLab) and allows multiple users to access and collaborate on the project. Changes made locally can be pushed to the remote repository, and updates from others can be pulled into your local repository

1. What are the basic Git commands?

git init: Initializes a new Git repository.

git clone: Copies an existing repository.

git add: Stages changes for the next commit.

git commit: Records the staged changes.

git push: Sends local commits to a remote repository.

git pull: Fetches and merges changes from a remote repository.

1. How do you check the status of a Git repository?

To check the status of a Git repository, you can use the command git status. This command provides information about the current branch, staged changes, and any untracked files, helping you understand what changes are ready to be committed

1. What is the purpose of branches in Git, and how do you create and switch between them?

Branches in Git allow developers to work on different features or fixes independently without affecting the main codebase. You can create a new branch using git branch <branch-name> and switch to it with git checkout <branch-name>. Alternatively, you can create and switch in one command using git checkout -b <branch-name>.

1. What are GitLab Desktop and GitHub, and how are they different from Git?

GitLab Desktop and GitHub are platforms that provide hosting for Git repositories, along with additional features for collaboration, project management, and CI/CD (Continuous Integration/Continuous Deployment). While Git is the version control system that tracks changes, GitHub and GitLab offer user interfaces and tools to manage repositories, track issues, and facilitate team collaboration.

1. How do you connect a local Git repository to a GitLab or GitHub repository?

COMMAND: git remote add origin <repository-URL>

This command sets the remote repository as the origin, allowing you to push and pull changes between your local and remote repositories

1. What are the steps to collaborate with others using GitLab or GitHub?

* **Clone the repository:** Use git clone <repository-URL> to get a local copy.
* **Create a branch:** Work on a new feature or fix in a separate branch.
* **Make changes and commit:** Use git add and git commit to save your changes.
* **Push your branch:** Use git push origin <branch-name> to upload your changes.
* **Create a pull request (PR):** On GitHub, submit a PR to merge your changes into the main branch.

1. How do you resolve merge conflicts in Git?

* Identify the conflicting files after a merge attempt.
* Open the files and look for conflict markers (e.g., <<<<<<<, =======, >>>>>>>).
* Manually edit the files to resolve the conflicts.
* After resolving, stage the changes with git add and complete the merge with git commit

1. What is a pull request, and why is it used in GitHub?

A pull request (PR) is a request to merge changes from one branch into another, typically from a feature branch into the main branch. It allows team members to review the changes, discuss potential modifications, and ensure code quality before integration. This process enhances collaboration and code review practices.

1. What are some best practices for writing commit messages?

* **Be concise and descriptive:** Summarize the changes in a clear manner.
* **Use the imperative mood:** Write messages like "Fix bug" or "Add feature" to convey actions.
* **Reference issues:** If applicable, mention related issue numbers for context.
* **Limit line length:** Keep the first line under 50 characters for better readability.