**GIT ASSESSMENT**

**Initialize a New Git Repository:**

* Create a new directory.
* Navigate into the directory and run git init to initialize a new Git repository.

**Create and Commit Initial Files:**

* Create a file called README.md.
* Add some content to the README.md.
* Commit the changes.

**Create a New Branch:**

* Create a new branch named feature-branch.
* Switch to the new branch.

**Make Changes in the New Branch:**

* Add a new file called feature-file.txt.
* Commit the changes.

**Switch Back to the Main Branch:**

* Switch back to the main branch.

**Create and Merge a Hotfix Branch:**

* Create a new branch named hotfix.
* Make a change in the README.md file.
* Commit the hotfix.
* Merge the hotfix branch into the main branch.

**Create a Merge Conflict:**

* Create a new branch named conflict-branch.
* Make a change to README.md in this branch.
* Switch back to the main branch and make a conflicting change in the same line of README.md.
* Attempt to merge conflict-branch into the main branch and resolve the merge conflict.

**Revert a Commit:**

* Identify the commit hash of the commit made in step 2.
* Revert the commit.

**Create a Remote Repository:**

* Create a new repository on a Git hosting service (GitHub, GitLab, Bitbucket).
* Link your local repository to the remote repository.
* Push Changes to Remote Repository:
* Push the main branch and the feature-branch to the remote repository.

**Fork a Repository:**

* Fork a public repository on GitHub.
* Clone Forked Repository Locally:
* Clone the forked repository to your local machine.

**Create a Pull Request:**

* Create a new branch in your forked repository.
* Make a change and commit it.
* Create a pull request to the original repository.

**Review and Merge the Pull Request:**

* Switch to the original repository.
* Review the pull request made from your fork.
* Merge the pull request.

**Fetch Changes:**

* Create a new branch in your local repository.
* Fetch changes from the original repository without merging them.

**View Commit History:**

* View the commit history of your local repository.

**Tag a Release:**

* Tag the latest commit as version v1.0.

**Amend the Last Commit:**

* Make a change to the last commit message.
* Do not change the commit content, only the commit message.

**Use Git Stash:**

* Create and apply a stash to temporarily save changes without committing.
* Apply the stash changes to your working directory.

**Clean Up:**

* Delete the feature-branch, hotfix, and conflict-branch branches.
* Remove the remote repository link.