**Exercise 1: Initialize a New Repository and Make First Commit**

**Scenario:**

You are starting a new project and need to create a Git repository.

**Solution:**

mkdir my\_project

cd my\_project

git init

echo "My First Project" > README.md

git add README.md

git commit -m "Initial commit with README"

**Exercise 2: Clone an Existing Repository**

**Scenario:**

Your team has an existing repository on GitHub. You need to clone it locally.

**Solution:**

git clone https://github.com/user/repository.git

cd repository

**Exercise 3: Create a New Branch and Switch to It**

**Scenario:**

You want to work on a new feature without affecting the main branch.

**Solution:**

git branch feature-branch

git checkout feature-branch

# Or using the newer command:

git switch -c feature-branch

**Exercise 4: Merge a Branch into Main**

**Scenario:**

You completed your feature in feature-branch and need to merge it into main.

**Solution:**

git checkout main

git merge feature-branch

git branch -d feature-branch # Delete the merged branch

**Exercise 5: Push Changes to GitHub**

**Scenario:**

You made some changes locally and want to push them to GitHub.

**Solution:**

git add .

git commit -m "Updated feature"

git push origin main

**Exercise 6: Pull Latest Changes from GitHub**

**Scenario:**

Your teammate has pushed new changes, and you need to update your local repository.

**Solution:**

git pull origin main

**Exercise 7: Resolve Merge Conflicts**

**Scenario:**

You tried merging a branch, but a conflict occurred.

**Solution:**

Open the conflicting file.

Manually resolve the conflicts.

Mark it as resolved:

git add conflicted\_file.txt

Commit the resolved changes:

git commit -m "Resolved merge conflict"

Continue merging if needed:

git merge --continue

**Exercise 8: Revert the Last Commit**

**Scenario:**

You accidentally committed changes and want to undo them.

**Solution:**

git revert HEAD

git push origin main

**Exercise 9: Stash and Apply Uncommitted Changes**

**Scenario:**

You made changes but need to switch branches without committing them.

**Solution:**

git stash

git checkout main

# Switch back and apply stashed changes

git checkout feature-branch

git stash pop

**Exercise 10: Fork and Contribute to a Repository**

**Scenario:**

You want to contribute to an open-source project on GitHub.

**Solution:**

**Fork** the repository from GitHub.

Clone the forked repository:

git clone https://github.com/your-username/repository.git

Create a new branch for your changes:

git checkout -b my-contribution

Make changes, commit, and push:

git add .

git commit -m "Added new feature"

git push origin my-contribution

**Create a Pull Request** on GitHub.