**Git Assignment Solutions**

**1. What commands would you use to stage all the changes and commit them with a meaningful commit message?**

sql

CopyEdit

git add .

git commit -m "Add meaningful description of changes"

* git add . stages all modified and new files.
* git commit -m "Your message" commits changes with a clear message, e.g., "Fix navbar styling and update README".

**2. You have committed changes to a wrong branch. How would you move these commits to the correct branch?**

**Step 1:** Switch to the correct branch:

nginx

CopyEdit

git checkout correct-branch-name

**Step 2:** Cherry-pick the commits from the wrong branch:

php-template

CopyEdit

git cherry-pick <commit-hash>

If multiple commits:

bash

CopyEdit

git checkout wrong-branch

git log # Copy commit hashes

git checkout correct-branch

git cherry-pick <hash1> <hash2> ...

**3. Create a new branch, make changes, and push the branch to remote**

pgsql

CopyEdit

git checkout -b feature-branch-name # Create and switch to new branch

# Make your changes

git add .

git commit -m "Add new feature"

git push -u origin feature-branch-name # Push branch to GitHub

**4. Contribute to an open-source project on GitHub**

1. Fork the repository on GitHub.
2. Clone your fork:

bash

CopyEdit

git clone https://github.com/your-username/project-name.git

cd project-name

1. Create a new branch:

css

CopyEdit

git checkout -b feature-branch

1. Make changes and commit:

sql

CopyEdit

git add .

git commit -m "Describe your contribution"

1. Push to your fork:

perl

CopyEdit

git push origin feature-branch

1. Open a Pull Request on GitHub from your fork to the original repository.

**5. Resolve merge conflicts between your branch and the main branch**

css

CopyEdit

git checkout main

git pull origin main # Update local main branch

git checkout your-branch

git merge main # Merge main into your branch

If conflicts occur:

* Open conflicted files, fix conflicts (look for <<<<<<<, =======, >>>>>>>).
* After resolving:

sql

CopyEdit

git add .

git commit -m "Resolve merge conflicts with main"

**6. Create a feature branch based on the latest changes in the main branch**

css

CopyEdit

git checkout main

git pull origin main # Update main branch

git checkout -b feature-new-feature # Create and switch to new branch

**7. Revert to a specific commit, discarding all changes made after that commit**

pgsql

CopyEdit

git log # Find commit hash to revert to

git reset --hard <commit-hash> # Reset to that commit, discard later changes

**Warning:** This discards all changes after that commit permanently.

**8. Restore a deleted file from a previous commit**

pgsql

CopyEdit

git log -- <filename> # Find commit where file existed

git checkout <commit-hash>^ -- <filename> # Restore file from previous commit

git add <filename>

git commit -m "Restore deleted file"