Assignment 1: Initialize a new Git repository in a directory of your choice. Add a simple text file to the repository and make the first commit.

Ans:

1. First, navigate to the directory where you want to create the Git repository:

```
cd /path/to/your/directory
```

2. Initialize a new Git repository:

```
git init
```

3. Create a simple text file (let's say `example.txt`) and add some content to it:

```
echo "This is a simple text file." > example.txt
```

4. Add the text file to the staging area:

```
git add example.txt
```

5. Commit the changes:

```
git commit -m "Initial commit: Added example.txt"
```

Now, you have initialized a new Git repository, added a simple text file (`example.txt`), and made the first commit with the message "Initial commit: Added example.txt".

Assignment 2: Branch Creation and Switching Create a new branch named 'feature' and switch to it. Make changes in the 'feature' branch and commit them.

Ans:

1. Create a new branch named 'feature':

```
git branch feature
```

2. Switch to the 'feature' branch:

```
git checkout feature
```

Alternatively, you can use a single command to create a new branch and switch to it:

```
git checkout -b feature
```

3. Make changes in the 'feature' branch. For example, you can edit existing files or create new ones.

4. Stage the changes:

```
git add <file1> <file2> ...  # Add specific files
```

Or to stage all changes:

```
git add .
```

5. Commit the changes:

```
git commit -m "Made changes in the feature branch"
```

Now, you have created a new branch named 'feature', switched to it, made changes, and committed them to the 'feature' branch.

Assignment 3: Feature Branches and Hotfixes Create a 'hotfix' branch to fix an issue in the main code. Merge the 'hotfix' branch into 'main' ensuring that the issue is resolved.

Ans:

1. Create a new branch named 'hotfix':

```
git checkout -b hotfix
```

2. Make necessary changes in the 'hotfix' branch to fix the issue.

3. Stage the changes:

```
git add <file1> <file2> ...  # Add specific files
```

Or to stage all changes:

```
git add .
```

4. Commit the changes:

```
git commit -m "Fixed the issue in the hotfix branch"
```

5. Switch back to the 'main' branch:

```
git checkout main
```

6. Merge the 'hotfix' branch into 'main':

```
git merge hotfix
```

7. Resolve any merge conflicts if they occur. Once conflicts are resolved, make a new commit to finalize the merge.

8. Push the changes to the remote repository if needed:

```
git push origin main
```

Now, the issue is resolved by fixing it in the 'hotfix' branch, and the changes are merged into the 'main' branch.