1.Clone, Create, Commit, and Push to Remote

```
git clone https://github.com/guptadeepak0405/testing.git
cd testing
git checkout -b feature-branch
echo "Hello from feature-branch!" > newfile.txt
git add newfile.txt
git commit -m "Add newfile.txt with initial content"
git push origin feature-branch
```

2. Create and Merge Branches

```
git clone https://github.com/guptadeepak0405/testing.git
cd testing
git checkout -b my-feature
echo "This is a new line added from my-feature branch." >> README.md
git add README.md
git commit -m "Update README.md from my-feature branch"
git checkout main
git merge my-feature
```

Aim 3: Revert a Commit

```
# 1. Clone the repository
git clone https://github.com/your-username/your-repo-name.git
cd your-repo-name
# 2. Make 3 commits
echo "Hello World" > file1.txt
git add file1.txt
git commit -m "Commit 1: Add file1.txt"
echo "Second line" >> file1.txt
git add file1.txt
git commit -m "Commit 2: Add second line to file1.txt"
touch file2.txt
git add file2.txt
git commit -m "Commit 3: Add file2.txt"
# 3. View commit history
git log --oneline
# (Note the commit hash of the one you want to revert, e.g., d4e5f6g for Commit 2)
# 4. Revert the specific commit
git revert d4e5f6g
# 5. Confirm the revert worked
git log --oneline
cat file1.txt
```

Aim 4:anmed

```
# 1. Clone the repository
git clone https://github.com/your-username/your-repo-name.git
```

```
cd your-repo-name

# 2. Create a file and make an initial commit
echo "Initial content" > file.txt
git add file.txt
git commit -m "Initial commit mesage with typo"  # ← intentionally with a typo

# 3. Amend the last commit message
git commit --amend -m "Initial commit message with typo corrected"

# 4. Verify the updated commit message
git log --oneline

5.stash
git clone https://github.com/your-username/your-repo-name.git
cd your-repo-name
echo "Some temporary changes" >> file.txt
git stash
git checkout other-branch
git checkout main
git stash apply   # or git stash pop
git add file.txt
git commit -m "Apply stashed changes"

6.
# 1. Clone the repository
git clone https://github.com/your-username/your-repo-name.git
cd your-repo-name

# 2. Create branch-1 and make an initial commit
git checkout -b branch-1
echo "This is the initial content of file.txt" > file.txt
git add file.txt
git commit -m "Commit on branch-1"

# 3. Create branch-2 from branch-1
git checkout -b branch-2

# 4. Modify file.txt on branch-1
git checkout branch-1
echo "This is a change from branch-1" > file.txt
git add file.txt
git commit -m "Change in file.txt on branch-1"

# 5. Modify file.txt on branch-2 (same line)
git checkout branch-2
echo "This is a conflicting change from branch-2" > file.txt
git add file.txt
```

```
git commit -m "Change in file.txt on branch-2"

# 6. Attempt to merge branch-2 into branch-1
git checkout branch-1
git merge branch-2

# You will see a conflict in file.txt
# Now manually resolve the conflict in file.txt by editing it
# For example, combine the changes like so:
# This is a change from branch-1
# This is a conflicting change from branch-2

# 7. After resolving the conflict, add the file
git add file.txt

# 8. Commit the resolved conflict
git commit -m "Resolved merge conflict in file.txt"

# 9. Verify the merge was successful
git log --oneline

# Check the content of file.txt to confirm the merged changes
cat file.txt
```

7.

```
# 1. Clone a repository with submodules
git clone --recursive https://github.com/your-username/your-repo-name.git
cd your-repo-name

# 2. Initialize and update submodules
git submodule update --init --recursive

# 3. Modify a file in the submodule
cd path/to/submodule
echo "Some change in the submodule" > file-in-submodule.txt
git add file-in-submodule.txt
git commit -m "Modified file-in-submodule.txt in submodule"
git push origin main

# 4. Commit the changes to the main repository
cd ../..
git add path/to/submodule
git commit -m "Updated submodule to latest commit"
git push origin main
```

8.

```
# Clone the repository
git clone https://github.com/your-username/your-repo-name.git
cd your-repo-name

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

# Modify a file and commit the changes (repeat this for at least 2-3 commits)
echo "First change in feature branch" > file.txt
git add file.txt
git commit -m "First commit on feature-branch"

echo "Second change in feature branch" >> file.txt
git add file.txt
git commit -m "Second commit on feature-branch"

echo "Third change in feature branch" >> file.txt
git add file.txt
git commit -m "Third commit on feature-branch"
```