## Experiment No :1

1. **Setting up and Basic Commands**

Initialize a new Git Repository in a directory. Create a new file and add it to the staging area and commit the changes with an appropriate commit message.

**Step 1: Initialize a new Git Repository**

- Open a terminal or command prompt.
- Create a Directory
  **mkdir first**
- Navigate to the directory where you want to create your new Git repository.
  **cd path/to/your/directory**
- Initialize a new Git repository.
  **git init**

**Step 2: Create a New File**

- Use any text editor of your choice to create a new file in the repository directory. For example, let's create a file named sample.txt.
  **vim sample.txt**
- Open sample.txt in your text editor and add some content.

**Step 3: Add the File to the Staging Area**

- Add the newly created file to the staging area.
  **git add sample.txt**
  This command stages the changes in sample.txt for the next commit.

**Step 4: Commit the Changes**

- Commit the changes to the repository with a meaningful commit message.
  **git commit -m "Initial commit: Added sample.txt"**

**Experiment No :2**

**Creating and Managing Branches**
Create a new branch named "feature-branch." Switch to the "master" branch. Merge the "feature-branch" into "master."

**Step 1: Initialize a new Git Repository**

- Open a terminal or command prompt.
- Create a Directory
  **mkdir second**
  - Navigate to the directory where you want to create your new Git repository.
    **cd path/to/your/directory**
  - Initialize a new Git repository.
    **git init**

**Step 2: Create a new branch named "feature-branch"**

- **git branch feature-branch**
  This command creates a new branch named "feature-branch".

**Step 3: Switch to the feature-branch**

- **git checkout feature-branch**
  This command switches to feature-branch from master branch.

**Step 4: Make changes in the "feature-branch":**

- Make the necessary changes to your code in the "feature-branch."
- Edit the file using the editor and make appropriate changes to the source code.

**Step 5: Commit the changes in "feature-branch":**

- **git add .**
- **git commit -m "Implement new feature in feature-branch"**
  This command stages and commits your changes in the "feature-branch."

**Step 6: Switch back to the "master" branch:**

- **git checkout master**
  This command switches back to the "master" branch.

**Step 7: Update the "master" branch (optional):**

- It's a good practice to ensure your "master" branch is up-to-date before merging changes. Fetch the latest changes from the remote repository:
  **git pull origin master**

**Step 8: Merge "feature-branch" into "master":**

- **git merge feature-branch**
  This command merges the changes from "feature-branch" into the "master" branch.