

**Experiment No.: 02**

**Title: Source Code Management: Git and GitHub**

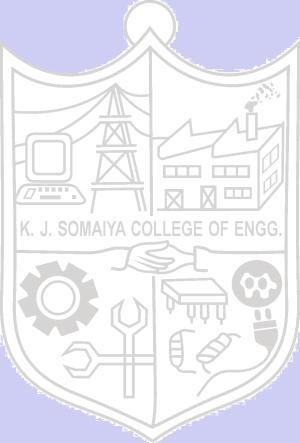
**Batch: A1 Roll No.: 1814068 Experiment No.:0**2 **Aim:** Installation and Integration of Source Code Management tools Git and GitHub

**Resources needed:** Internet

**Pre Requisite:** Git Commands and GitHub Account

## Theory:

**Introduction to Git**



Git is a version control system for tracking changes in computer files and coordinating work on those files among multiple people. It is primarily used for source code management in software development, but it can be used to keep track of changes in any set of files.

# Git Installation

1. Download Git for Windows.
2. Extract and Launch Git Installer.
3. Server Certificates, Line Endings and Terminal Emulators.
4. Additional Customization Options.
5. Complete Git Installation Process.
6. Launch Git Bash Shell.
7. Launch Git GUI.

## Git Commands:

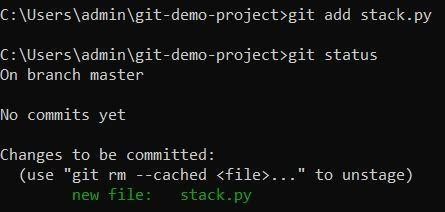
**Git Init**

The git init command creates a new Git repository. It can be used to convert an existing, unversioned project to a Git repository or initialize a new, empty repository.

https://lh3.googleusercontent.com/XhLyInQ_GOsoTTEh_z4ejKkpw-spfmMkfthRKpl4nC0B-WP8nvy4zrRNJUCQrss0zaNkUMZ7ndmh4gEJVOiWyFZbNZ0xM52B4BIwEe3H79cXYpLhk5UHDzkjc7FRlEk03jj2CEY

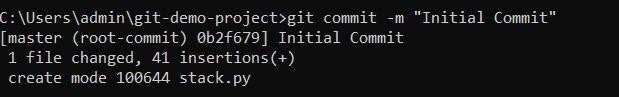
## Git Add

When we create, modify or delete a file, these changes will happen in our local and won't be included in the next commit (unless we change the configurations). We need to use the git add command to include the changes of a file(s) into our next commit. The git add command doesn't change the repository and the changes are not saved until we use git commit. The git add command adds new or changed files in your working directory.



## Git Commit

A commit, or "revision", is an individual change to a file (or set of files). It's like when you save a file, except with Git, every time you save it creates a unique ID (a.k.a. the "SHA" or "hash") that allows you to keep record of what changes were made when and by who. In version control systems, a commit is an operation which sends the latest changes to the source code to the repository, making these changes part of the head revision of the repository.



## Git- Push and Pull

**Push operation:** The git push command is used to upload local repository content to aremote repository. Pushing is how you transfer commits from your local repository to a remote repository. The command for push is git push remote –all.

**Pull operation:** The git pull command is used to fetch and download content from a remote repository and immediately update the local repository to match that content. The command is: git pull <remote>.

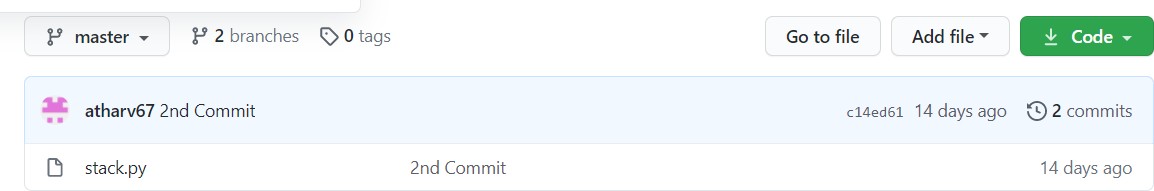
## Git Clone

Git clone is a Git command line utility which is used to target an existing repository and create a clone, or copy of the target repository. In this page we'll discuss extended configuration options and common use cases of git clone. Clones a repository into a newly created directory, creates remote-tracking branches for each branch in the cloned repository (visible using git branch --remotes), and creates and checks out an initial branch that is forked from the cloned repository’s currently active branch.

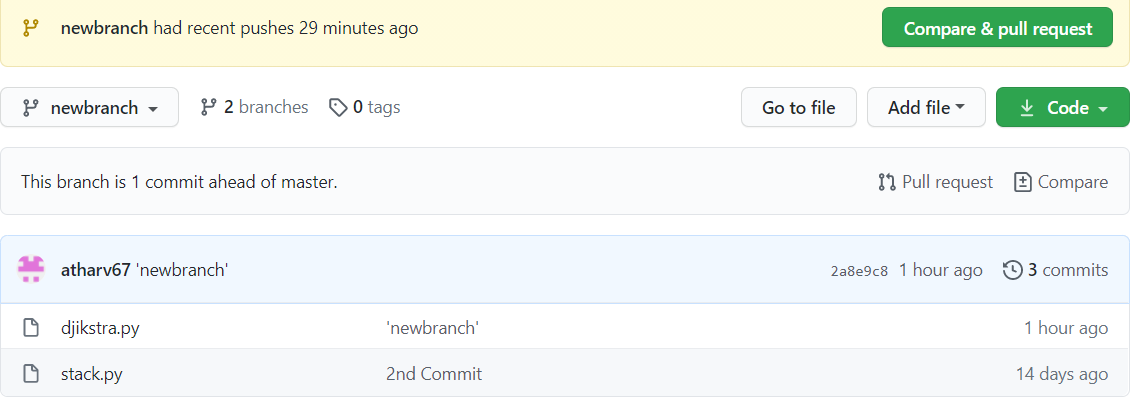
**Branches:** Git branches are effectively a pointer to a snapshot of your changes. When you want to add a new feature or fix a bug, no matter how big or how small, you spawn a new branch to encapsulate your changes. This makes it harder for unstable code to get merged into the main code base, and it gives you the chance to clean up your future's history before merging it into the main branch.

A branch represents an independent line of development. Branches serve as an abstraction for the edit/stage/commit process. New commits are recorded in the history for the current branch, which results in a fork in the history of the project.

Files in master branch:



The files in the master branch are merged with newbranch.



## Procedure:

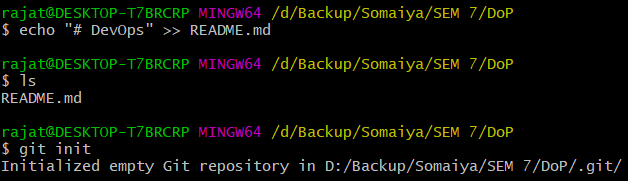
1. Download Git tool from official Git website: <https://git-scm.com/downloads>
2. Create GitHub Account <https://github.com/join>
3. Execute Git Commands: version, global, init, add, commit, checkout, push, pull etc.

## Results: (Document with screenshots)

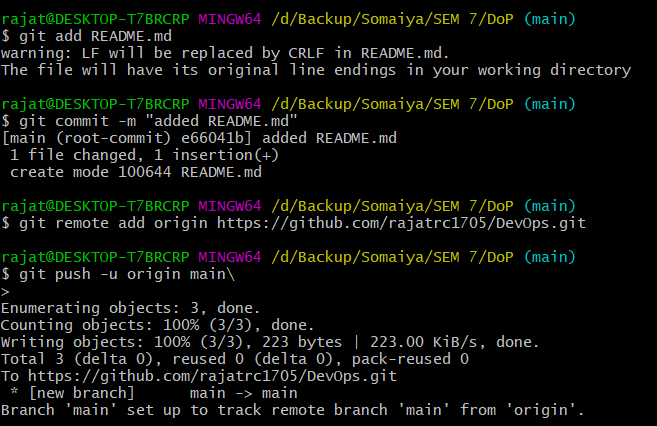
1. All Git commands
2. Working with branch and Master
3. Create a file on Git and Push it on GitHub
4. Pull files from GitHub

Output:

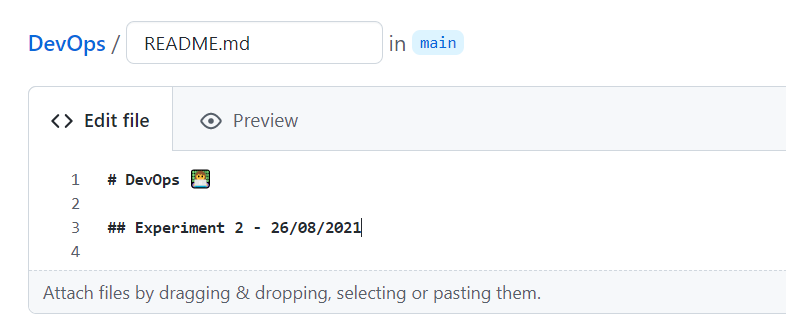
Initialising Git Repository



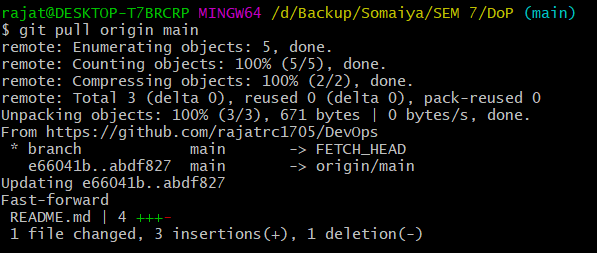
Adding Readme to the repository



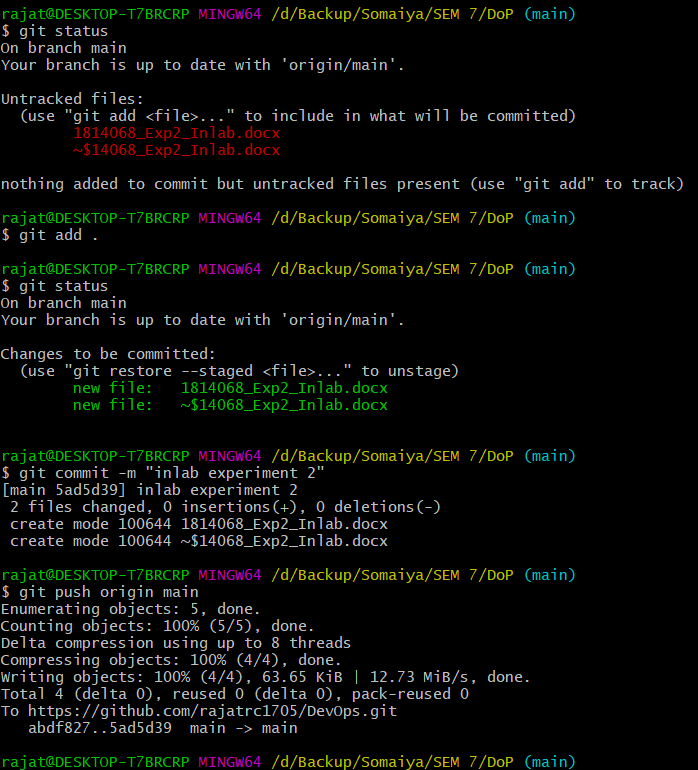
Make changes in the Readme on the repository and **pull them to local repository.**



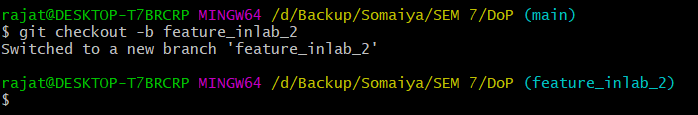
Pulling the changes with pull command



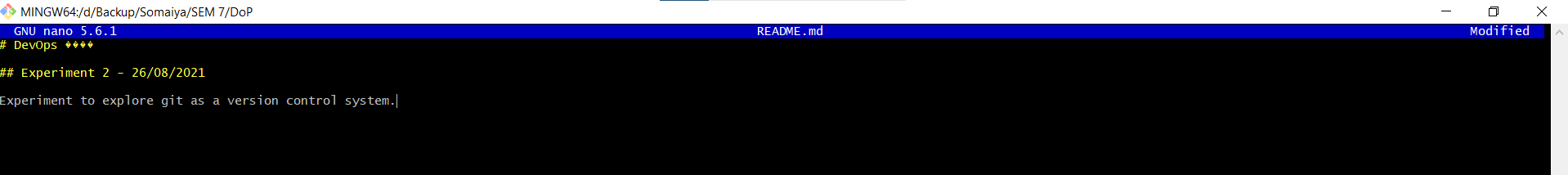
**Push** the inlab writeup to the git repository using **add, commit, push** commands.



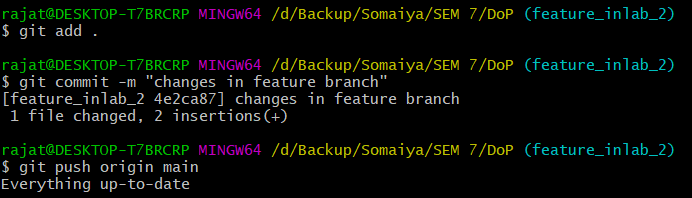
Creating new feature branch



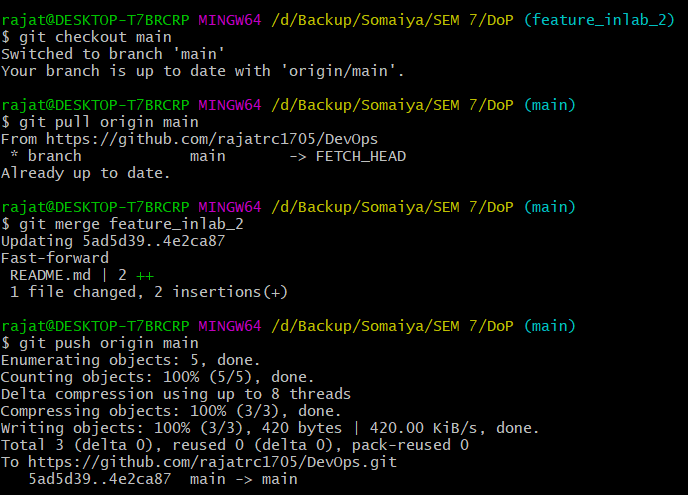
Making changes to readme



Pushing the changes to github



Merging feature branch with main branch



## Questions:

1. What is difference between Git and GitHub
2. What is Master and Branch

**Answer 1:**

**Git**

Git is a popular version control system that is used by developers to maintain the code files for their projects. Git is installed and maintained on your local system (rather than in the cloud) and gives you a self-contained record of your ongoing programming versions. It can be used completely exclusive of any cloud-hosting service — you don’t even need internet access, except to download it.

**Github**

Github is basically a cloud-based hosting system that is used to manage the Git repositories for your project.

**Answer 2:**

Master branch is the main and the default branch of Git that the developer maintains the code on initially.

Branches are the pointer to the commits that are made to a repository in git. There can be multiple branches during the development of a project.

Generally, the features for a project are developed on a separate feature branch and then after the feature is developed fully, it is merged into the ‘master’ or the ‘main’ branch of the git repository for the project.

## Outcomes:

**Conclusion: (Conclusion to be based on the Results and outcomes achieved)**

Through this experiment we revised the concepts of Github and revised the various commands that Github provides for version control of our projects.

**Grade: AA / AB / BB / BC / CC / CD /DD Signature of faculty in-charge with date**

**References:**

**Books/ Journals/ Websites:**

* 1. https://guides.github.com/introduction/git-handbook/
  2. Effective DevOps: Building a Culture of Collaboration, Affinity, and Tooling at Scale, Jennifer Davis, Ryn Daniels, O'Reilly Media June 2016.
  3. Practical DevOps: Implement DevOps in your organization by effectively building, deploying, testing, and monitoring code,Joakim Verona, Packt Publishing, 2nd Edition,2016