Subject

Name

:

**Source**

**Code**

**Management**

Subject

Code

:

**CSE**

**0106**

**24**

Cluster

:

**Alpha**

Department:

**DCSE**



**Submitted By: Submitted To:**

Name: Vikrant Garg Dr. Renu Popli

2410990158 Department of Computer

G02 Science & Engineering

Chitkara University Institute of

Engineering and Technology Rajpura,

# Index Task 1.1

|  |  |  |
| --- | --- | --- |
| **Sr.**  **No.** | **Program Title** | **Page No.** |
| **1.** | **To install and configure Git Client on your local system** | **3-4** |
| **2.** | **Setting up GitHub Account and Adding Collaborators on**  **GitHub Repository** | **5-6** |
| **3.** | **To merge two branches within a Git repository.** | **7-9** |
| **4.** | **To demonstrate push and pull operations in Git.** | **10-12** |



# Practical No.: 1

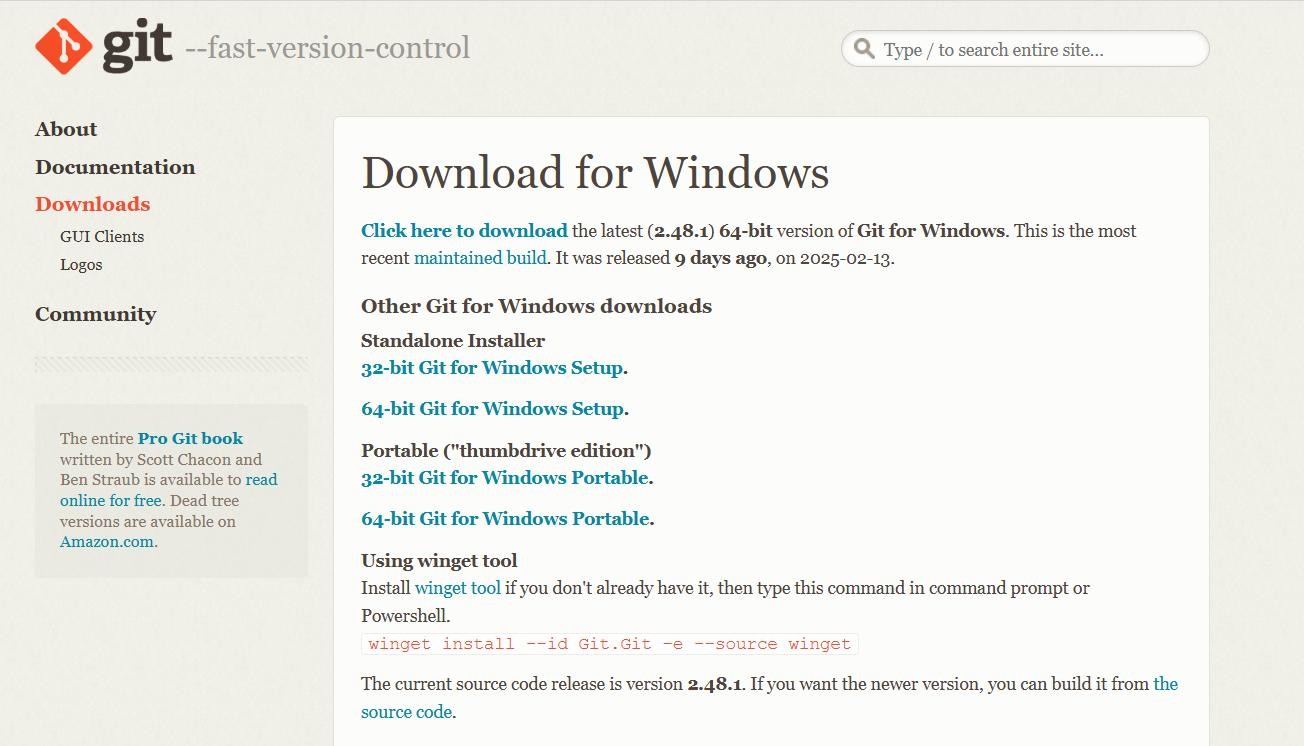
**Aim:** To install and configure Git Client on your local system.

**Theory:**

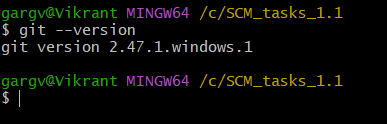
Git is a distributed version control system used to track changes in source code. This practical focuses on setting up Git on your local system for effective version control.

**Procedure:**

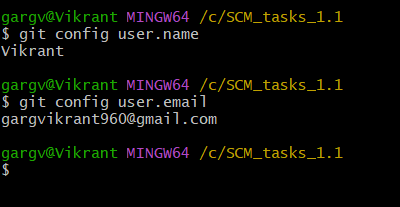
1. Download Git from [git-scm.com](https://git-scm.com/) .



1. Install Git by following the setup wizard.
2. Open Git Bash and verify installation using the command: git --version.



1. Configure user details using the commands:
   * git config --global user.name "Your Name"
   * git config --global user.email "Your Email"



# Practical No.: 2

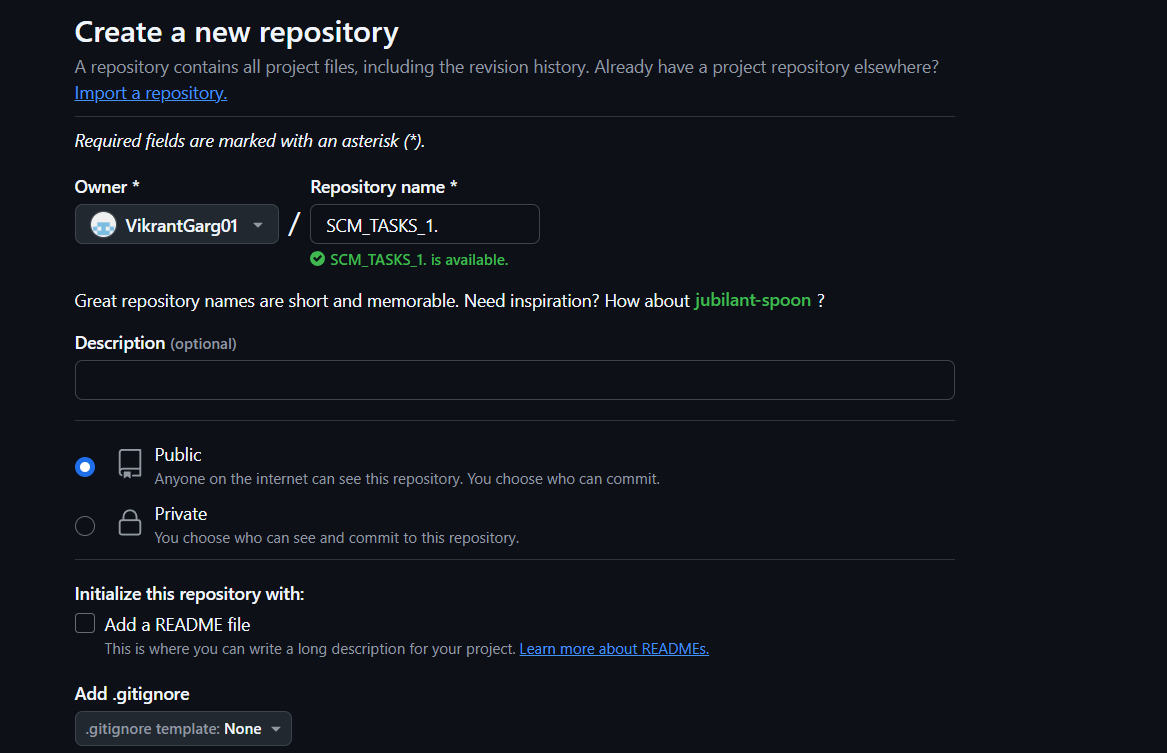
**Aim:** To add collaborators to a GitHub repository for collaborative work.

**Theory:**

Collaborators are individuals with write access to a repository. They can contribute to the project by pushing changes and merging pull requests.

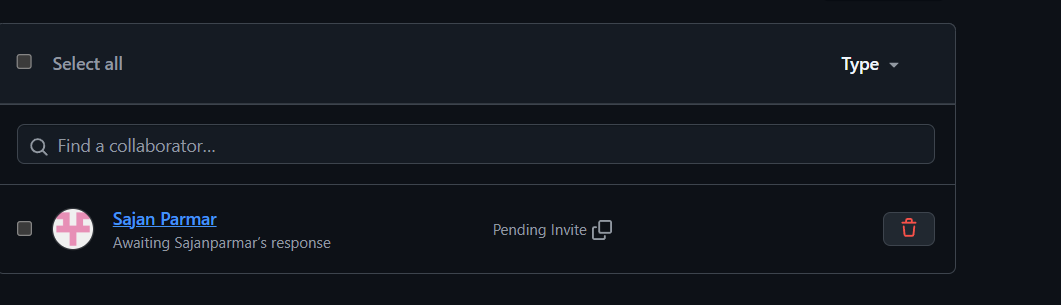
**Procedure:**

1. Log in to your GitHub account and create a new repository.

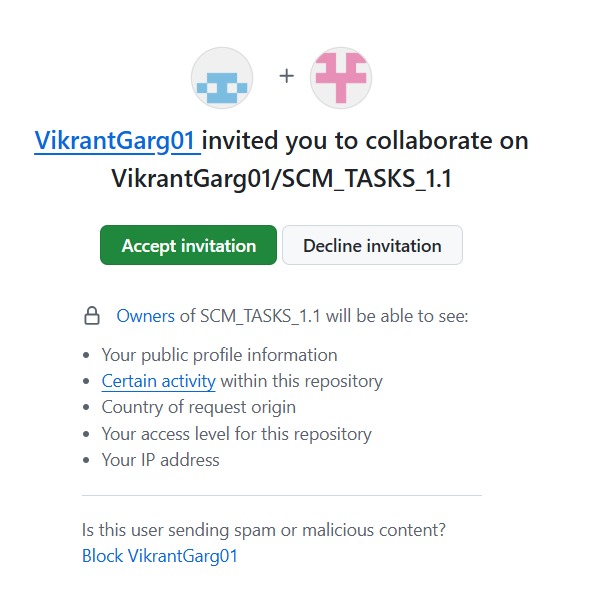
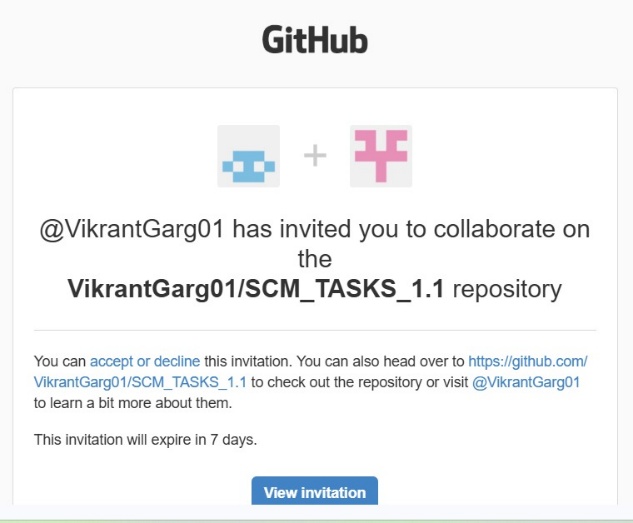


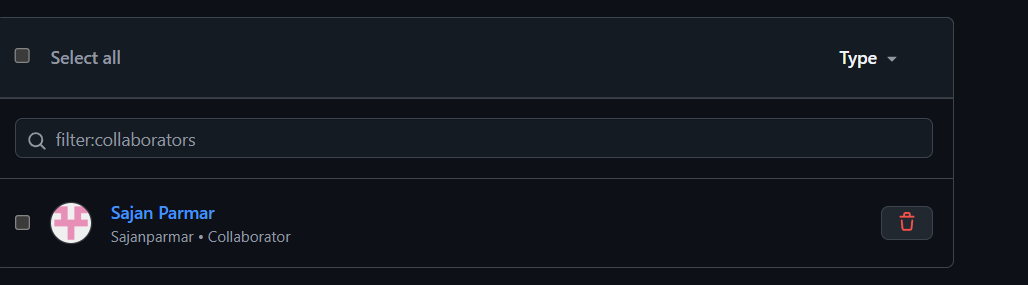
1. Navigate to Settings > Manage Access in the repository.

1. Add collaborators by their GitHub usernames.



1. Collaborators will receive an invitation email, which they must accept.





# Practical No.: 3

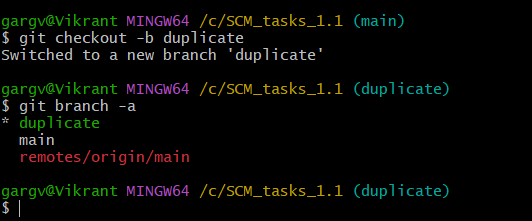
**Aim:** To merge two branches within a Git repository.

**Theory:**

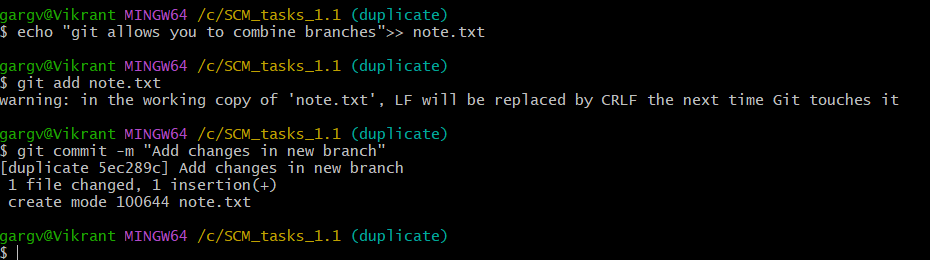
Merging branches in Git allows you to combine changes from one branch into another. It is a fundamental process in collaborative workflows, ensuring all contributions are integrated into a single codebase.

**Procedure:**

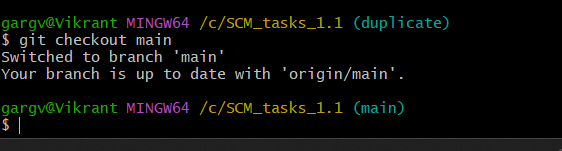
1. Create a new branch and switch to it: • git checkout -b new-branch



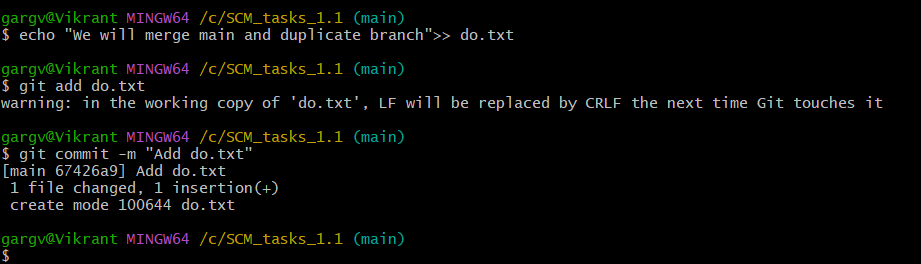
1. Make changes to a file in the new branch and commit them:
   * echo "New content" > file.txt
   * git add file.txt
   * git commit -m "Add changes in new branch"



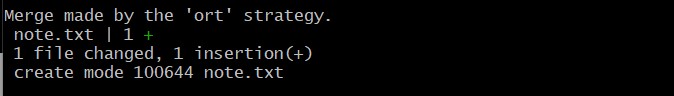
1. Switch back to the main branch:
   * git checkout main



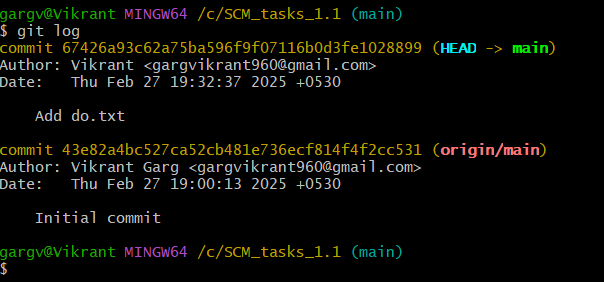
1. Modify another file in the main branch and commit the changes:
   * echo "Main branch changes" > another-file.txt
   * git add another-file.txt
   * git commit -m "Modify file in main branch"



1. Merge the new branch into the main branch:
   * git merge new-branch



**Git log :**



# Practical No.: 4

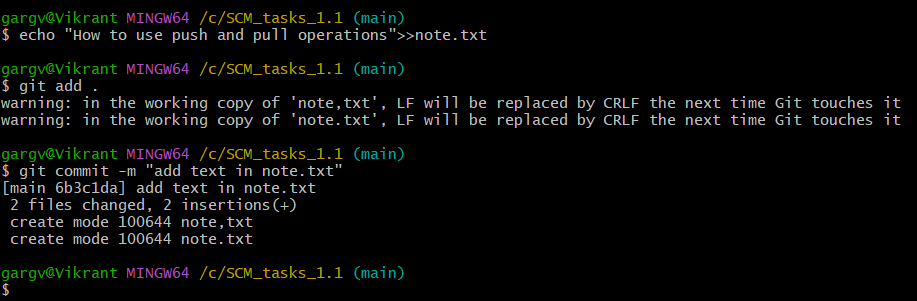
**Aim:** To demonstrate push and pull operations in Git.

**Theory:**

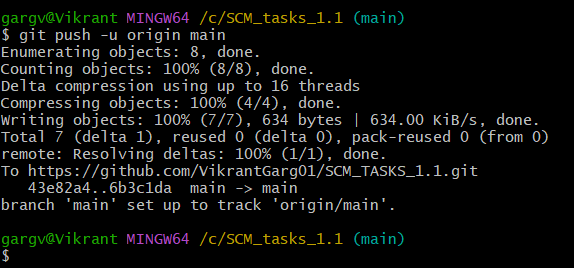
Push transfers committed changes from the local repository to the remote repository, while pull retrieves updates from the remote repository.

**Procedure:**

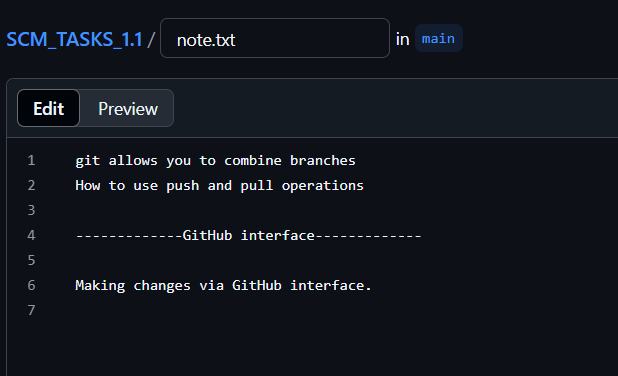
1. Make changes in the local repository and commit them.

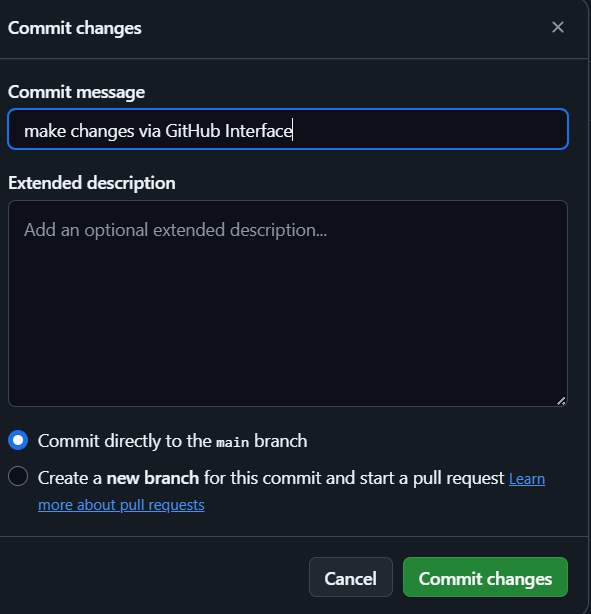


1. Push the changes to the remote repository using git push.

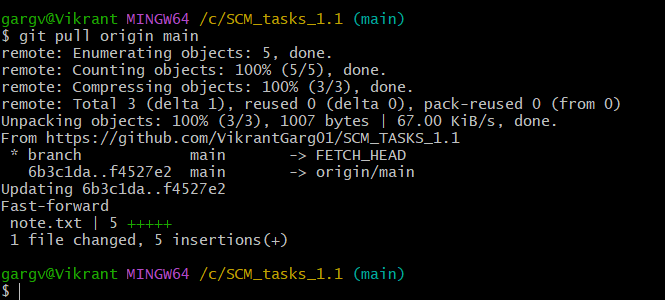


1. Make changes directly on the remote repository (e.g., via GitHub interface).

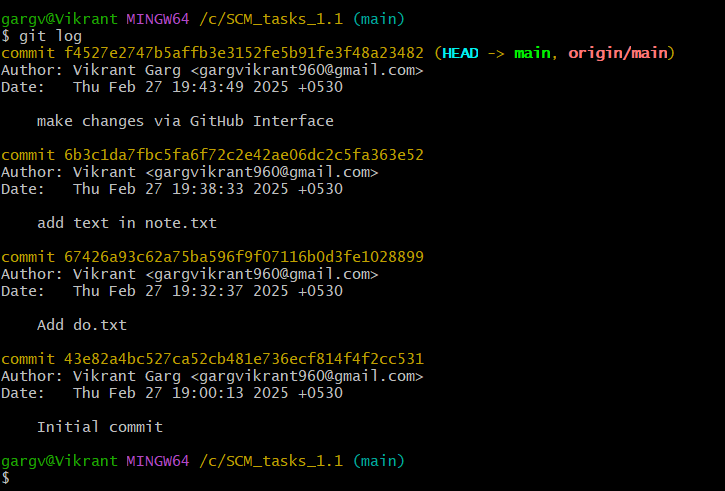




1. Pull the changes to the local repository using git pull.



**Git log :**

****