

Subject: Project Management with Git

Course Code: BCS358C

Name: SNEHAGANGA N S

USN: 4GW24CI048

Semester: 3rd Semester

Program 1: Setting Up and Basic Commands

1. Configure Git

```
git config --global user.name "SnehagangaNS"
```

```
git config --global user.email "snehaganganadumane@gmail.com"
```

2. Initialize repository

```
git init
```

3. Create a file

```
touch test.txt
```

4. Check status

```
git status
```

5. Add files to staging

```
git add .
```

6. Commit changes

```
git commit -m "Initial commit: added file1.txt"
```

7. Add GitHub remote (REPLACE with your repo URL)

```
git remote add origin https://github.com/username/4GW24CI048GIT.git
```

8. Verify remote

```
git remote -v
```

9. Rename branch to main

```
git branch -M main
```

10. Push to GitHub

```
git push -u origin main
```

```
MINGW64/e/4GW24CIO48NEWGIT
user@swasthi MINGW64 /e/4GW24CIO48NEWGIT (master)
$ git config --global user.name "SnehagangaNS"
user@swasthi MINGW64 /e/4GW24CIO48NEWGIT (master)
$ git config --global user.email "snehaganganadumane@gmail.com"
user@swasthi MINGW64 /e/4GW24CIO48NEWGIT (master)
$ git init
Reinitialized existing Git repository in E:/4GW24CIO48NEWGIT/.git/
user@swasthi MINGW64 /e/4GW24CIO48NEWGIT (master)
$ touch test.txt
user@swasthi MINGW64 /e/4GW24CIO48NEWGIT (master)
$ git add .
user@swasthi MINGW64 /e/4GW24CIO48NEWGIT (master)
$ git add test.txt
user@swasthi MINGW64 /e/4GW24CIO48NEWGIT (master)
$ git commit -m "initial commit"
[master aa0d672] initial commit
1 file changed, 1 insertion(+)
create mode 100644 test.txt
user@swasthi MINGW64 /e/4GW24CIO48NEWGIT (master)
$
```

Program 2: Creating and Managing Branches Create a new branch named "feature-branch." Switch to the "master" branch. Merge the "feature-branch" into "master."

1. Global Git configuration

```
git config --global user.name "SnehagangaNS"
```

```
git config --global user.email "snehaganganaumane@gmail.com"
```

2. Initialize a new Git repository

```
git init
```

3. Create a file and make first commit (required before branching)

```
touch test.txt
```

```
git add .
```

```
git commit -m "Initial commit"
```

4. Create a new branch named feature-branch

```
git branch feature-branch
```

5. Switch to feature-branch

```
git checkout feature-branch
```

6. Make changes in feature-branch

```
git add .
```

```
git commit -m "Added changes in feature-branch"
```

7. Switch back to master branch

```
git checkout master
```

8. Merge feature-branch into master

```
git merge feature-branch add a file in the main directory.
```

```
MINGW64:/e/4Gw24CI048NEWGIT
user@swasthi MINGW64 /e/4Gw24CI048NEWGIT (master)
$ git add test.txt

user@swasthi MINGW64 /e/4Gw24CI048NEWGIT (master)
$ git commit -m "initial commit"
[master aa0d672] initial commit
1 file changed, 1 insertion(+)
create mode 100644 test.txt

user@swasthi MINGW64 /e/4Gw24CI048NEWGIT (master)
$ git checkout feature-branch
Switched to branch 'feature-branch'

user@swasthi MINGW64 /e/4Gw24CI048NEWGIT (feature-branch)
$ git checkout -b feature-branch
fatal: a branch named 'feature-branch' already exists

user@swasthi MINGW64 /e/4Gw24CI048NEWGIT (feature-branch)
$ git merge feature-branch
Already up to date.

user@swasthi MINGW64 /e/4Gw24CI048NEWGIT (feature-branch)
$ git checkout feature-branch
Already on 'feature-branch'

user@swasthi MINGW64 /e/4Gw24CI048NEWGIT (feature-branch)
$ git checkout master
Switched to branch 'master'

user@swasthi MINGW64 /e/4Gw24CI048NEWGIT (master)
$ git checkout -b feature-branch
fatal: a branch named 'feature-branch' already exists

user@swasthi MINGW64 /e/4Gw24CI048NEWGIT (master)
$ git merge feature-branch
Already up to date.

user@swasthi MINGW64 /e/4Gw24CI048NEWGIT (master)
$ git checkout feature-branch
Switched to branch 'feature-branch'

user@swasthi MINGW64 /e/4Gw24CI048NEWGIT (feature-branch)
$ git merge feature-branch
Already up to date.

user@swasthi MINGW64 /e/4Gw24CI048NEWGIT (feature-branch)
$
```

Program 3: Creating and Managing Branches Write the commands to stash your changes, switch branches, and then apply the stashed changes.

git config --global user.name "SnehagangaNS"

git config --global user.email "snehaganganadumane@gmail.com"

git init

touch test.txt

git add .

git commit -m "Initial commit"

git branch feature-branch

git checkout feature-branch

git rm test.txt

git commit -m "Deleted test.txt in feature-branch"

git checkout master

git stash

git checkout feature-branch

git stash apply

git add test.txt

git commit -m "Resolved stash conflict by restoring test.txt"

```
MINGW64/e/4GW24CI048NEWGIT
$ git merge feature-branch
Already up to date.

user@swasthi MINGW64 /e/4GW24CI048NEWGIT (feature-branch)
$ git checkout master
Switched to branch 'master'

user@swasthi MINGW64 /e/4GW24CI048NEWGIT (master)
$ git status
On branch master
nothing to commit, working tree clean

user@swasthi MINGW64 /e/4GW24CI048NEWGIT (master)
$ git add .
$ git commit -m "initial commit"
[master de68ce5] initial commit
1 file changed, 1 insertion(+), 1 deletion(-)

user@swasthi MINGW64 /e/4GW24CI048NEWGIT (master)
$ git stash
No local changes to save

user@swasthi MINGW64 /e/4GW24CI048NEWGIT (master)
$ git stash
Saved working directory and index state WIP on master: de68ce5 initial commit

user@swasthi MINGW64 /e/4GW24CI048NEWGIT (master)
$ git branch feature-branch

user@swasthi MINGW64 /e/4GW24CI048NEWGIT (master)
$ git branch feature-branch
fatal: a branch named 'feature-branch' already exists

user@swasthi MINGW64 /e/4GW24CI048NEWGIT (master)
$ git checkout feature-branch
Switched to branch 'feature-branch'

user@swasthi MINGW64 /e/4GW24CI048NEWGIT (feature-branch)
$ git stash apply
CONFLICT (modify/delete): test.txt deleted in Updated upstream and modified in s
tashed changes. Version Stashed changes of test.txt left in tree.
On branch feature-branch
Unmerged paths:
  (use "git restore --staged <file>..." to unstage)
  (use "git add/rm <file>..." as appropriate to mark resolution)
        deleted by us:   test.txt
```

```
user@swasthi MINGW64 /e/4GW24CI048NEWGIT (feature-branch)
$ git add test.txt

user@swasthi MINGW64 /e/4GW24CI048NEWGIT (feature-branch)
$ git commit -m "Resolved stash conflict: restored test.txt"
[feature-branch ad36c78] Resolved stash conflict: restored test.txt
1 file changed, 1 insertion(+)
create mode 100644 test.txt
```

Program 4: Collaboration and Remote Repositories Clone a remote Git repository to your local machine.

```
user@swasthi MINGW64 /e/4GW24CI048NEWGIT (feature-branch)
$ git clone https://github.com/varsha-sureshh/SignLingo_TechSprint_Hackathon.git
Cloning into 'SignLingo_TechSprint_Hackathon'...
remote: Enumerating objects: 70, done.
remote: Counting objects: 100% (70/70), done.
remote: Compressing objects: 100% (54/54), done.
remote: Total 70 (delta 20), reused 66 (delta 16), pack-reused 0 (from 0)
Receiving objects: 100% (70/70), 24.14 MiB | 4.39 MiB/s, done.
Resolving deltas: 100% (20/20), done.
```

git config --global user.name "SnehagangaNS"

git config --global user.email "snehaganganadumane@gmail.com"

git clone https://github.com/username/sample-repo.git

cd sample-repo

git remote -v

Program 5: Collaboration and Remote Repositories Fetch the latest changes from a remote repository and rebase your local branch onto the updated remote branch.

```
user@swasthi MINGW64 /e/4GW24CI048NEWGIT (feature-branch)
$ git branch
* feature-branch
  feature-branh
  master

user@swasthi MINGW64 /e/4GW24CI048NEWGIT (feature-branch)
$ git fetch origin
remote: Enumerating objects: 21, done.
remote: Counting objects: 100% (21/21), done.
remote: Compressing objects: 100% (14/14), done.
remote: Total 21 (delta 2), reused 16 (delta 2), pack-reused 0 (from 0)
Unpacking objects: 100% (21/21), 3.60 KiB | 61.00 KiB/s, done.
From https://github.com/SnehagangaNS/4GW24CI048GIT
 * [new branch]      main      -> origin/main

user@swasthi MINGW64 /e/4GW24CI048NEWGIT (feature-branch)
$ git rebase origin origin/master
Successfully rebased and updated detached HEAD.
```

git checkout feature-branch

git fetch origin

git rebase origin/master

Program 6: Collaboration and Remote Repositories Write the command to merge "feature-branch" into "master" while providing a custom commit message for the merge.

git checkout master

git merge feature-branch --no-ff -m "Merged feature-branch into master"

```
surit@Dell-VarshaS MINGW64 /c/Varsha/GitLabPractice (main)
$ git merge feature-branch -m "mergind the branches"
Already up to date.
```

Program 7: Git Tags and Releases Write the command to create a lightweight Git tag named "v1.0" for a commit in your local repository.

git tag v1.0

```
user@swasthi MINGW64 /e/4GW24CI048NEWGIT (master)
$ git tag v1.0

user@swasthi MINGW64 /e/4GW24CI048NEWGIT (master)
$ git tag
v1.0

user@swasthi MINGW64 /e/4GW24CI048NEWGIT (master)
$
```

Program 8: Advanced Git Operations: Write the command to cherry-pick a range of commits from "source-branch" to the current branch


```
MINGW64:/e/4Gw24CI048NEWGIT
user@swasthi MINGW64 /e/4Gw24CI048NEWGIT (master)
$ git checkout -b source-branch
Switched to a new branch 'source-branch'

user@swasthi MINGW64 /e/4Gw24CI048NEWGIT (source-branch)
$ git add .
warning: adding embedded git repository: SignLingo_TechSprint_Hackathon
hint: You've added another git repository inside your current repository.
hint: Clones of the outer repository will not contain the contents of
hint: the embedded repository and will not know how to obtain it.
hint: If you meant to add a submodule, use:
hint:   git submodule add <url> SignLingo_TechSprint_Hackathon
hint: If you added this path by mistake, you can remove it from the
hint: index with:
hint:   git rm --cached SignLingo_TechSprint_Hackathon
hint: See "git help submodule" for more information.
hint: Disable this message with "git config set advice.addEmbeddedRepo false"

user@swasthi MINGW64 /e/4Gw24CI048NEWGIT (source-branch)
$ git commit -m "new"
[source-branch 8df9087] new
1 file changed, 1 insertion(+)
create mode 160000 SignLingo_TechSprint_Hackathon

user@swasthi MINGW64 /e/4Gw24CI048NEWGIT (source-branch)
$ git checkout master
warning: unable to rmdir 'SignLingo_TechSprint_Hackathon': Directory not empty
Switched to branch 'master'

user@swasthi MINGW64 /e/4Gw24CI048NEWGIT (master)
$ git log source-branch --oneline
8df9087 (source-branch) new
de68ce5 (HEAD -> master, tag: v1.0, feature-branch) initial commit
aa0d672 initial commit
280a1f8 changes done in feature-branch
c15f20f initial commit

user@swasthi MINGW64 /e/4Gw24CI048NEWGIT (master)
$ git cherry-pick 8df9087
[master 83b8167] new
Date: Tue Jan 6 05:57:57 2026 +0530
1 file changed, 1 insertion(+)
create mode 160000 SignLingo_TechSprint_Hackathon
```

Create and switch to source branch

git checkout -b source-branch

Make commits in source-branch (edit file manually before each commit)

git add .

git commit -m "First commit in source-branch"

git add .

git commit -m "Second commit in source-branch"

Switch back to target branch

git checkout master

View commits in source-branch

git log source-branch --oneline

Cherry-pick a range of commits from source-branch

git cherry-pick <start-commit>^..<end-commit>

Program 9: Analysing and Changing Git History Given a commit ID, how would you use Git to view the details of that specific commit, including the author, date, and commit message?

```
user@swasthi MINGW64 /e/4GW24C1048NEWGIT (master)
$ git show de68ce5
commit de68ce5e62c8454c752384818ea8660570d2ebc6 (tag: v1.0, feature-branh)
Author: SnehagangaNS <snehaganganadumane@gmail.com>
Date: Tue Jan 6 05:11:46 2026 +0530

    initial commit

diff --git a/test.txt b/test.txt
index fabfbb2..9394436 100644
--- a/test.txt
+++ b/test.txt
@@ -1,1 @@
- hi hello
\ No newline at end of file
+ hi hello namsate
\ No newline at end of file
```

git show <commit-id>

Program 10: Analysing and Changing Git History Write the command to list all commits made by the author "JohnDoe" between "2023-01-01" and "2023-12-31."

```

user@swasthi MINGW64 /e/4GW24CI048NEWGIT (master)
$ git log --author="Swasthi" --since="2025-12-31 00:00" --until="2026-01-04 23:59"

user@swasthi MINGW64 /e/4GW24CI048NEWGIT (master)
$ |

```

git log --author="Swasthi" --since="2025-01-01" --until="2026-12-31"

Program 11: Analysing and Changing Git History Write the command to display the last five commits in the repository's history.

```

user@swasthi MINGW64 /e/4GW24CI048NEWGIT (master)
$ git log -n 5
commit 83b81677768d4093cc193a52574fe36a425e6838 (HEAD -> master)
Author: SnehagangaNS <snehaganganadumane@gmail.com>
Date: Tue Jan 6 05:57:57 2026 +0530

    new

commit de68ce5e62c8454c752384818ea8660570d2ebc6 (tag: v1.0, feature-branh)
Author: SnehagangaNS <snehaganganadumane@gmail.com>
Date: Tue Jan 6 05:11:46 2026 +0530

    initial commit

commit aa0d67202404d45167220234cd46cd6640c1ed6b
Author: SnehagangaNS <snehaganganadumane@gmail.com>
Date: Tue Jan 6 04:43:21 2026 +0530

    initial commit

commit 280a1f887930d43dd50ead0ce94d13fa36bf9552
Author: Your Name <snehaganganadumane@gmail.com>
Date: Mon Jan 5 19:07:44 2026 +0530

    changes done in feature-branch

commit c15f20f2856316529d907598db20894aa31cf95e
Author: Your Name <snehaganganadumane@gmail.com>
Date: Mon Jan 5 19:06:18 2026 +0530

    initial commit

```

git log -n 5

Program 12: Analyzing and Changing Git History Write the command to undo the changes introduced by the commit with the ID "abc123".

git revert abc123

```
MINGW64/e/4GW24C048NEWGIT
fatal: revert failed

user@swasthi MINGW64 /e/4GW24C048NEWGIT (master)
$ git log
commit fcf161aa2ce564b3877f05e8271433057b60b319 (HEAD -> master)
Author: SnehagangaNS <snehaganganadumane@gmail.com>
Date: Tue Jan 6 06:33:43 2026 +0530

    Revert "initial commit"

    This reverts commit c15f20f2856316529d907598db20894aa31cf95e.

commit 3cda4dbb04dbe730d95bda0a271caf8d9626aea8
Author: SnehagangaNS <snehaganganadumane@gmail.com>
Date: Tue Jan 6 06:33:04 2026 +0530

    Revert "initial commit"

    This reverts commit aa0d67202404d45167220234cd46cd6640c1ed6b.

commit d5ee0437975a677f28df467c6b190d84976c6c12 (tag: v2.0)
Author: SnehagangaNS <snehaganganadumane@gmail.com>
Date: Tue Jan 6 06:21:13 2026 +0530

    Revert "changes done in feature-branch"

    This reverts commit 280a1f887930d43dd50ead0ce94d13fa36bf9552.

commit 83b81677768d4093cc193a52574fe36a425e6838
Author: SnehagangaNS <snehaganganadumane@gmail.com>
Date: Tue Jan 6 05:57:57 2026 +0530

    new

commit de68ce5e62c8454c752384818ea8660570d2ebc6 (tag: v1.0, feature-branh)
Author: SnehagangaNS <snehaganganadumane@gmail.com>
Date: Tue Jan 6 05:11:46 2026 +0530

    initial commit

commit aa0d67202404d45167220234cd46cd6640c1ed6b
Author: SnehagangaNS <snehaganganadumane@gmail.com>
Date: Tue Jan 6 04:43:21 2026 +0530

    initial commit

commit 280a1f887930d43dd50ead0ce94d13fa36bf9552
Author: Your Name <snehaganganadumane@gmail.com>
```