Experiment -1.2

Student Name: Sushant jha **UID: 22BDO10052**
Branch: CSE (Devops) **Section/Group: 22BCD-1\A**
Semester: 4th **Date of Performance: 17/01/24**
Subject Name: Git and Git Hub **Subject Code: 22CSH-293**

1. **Aim/Overview of the practical:**

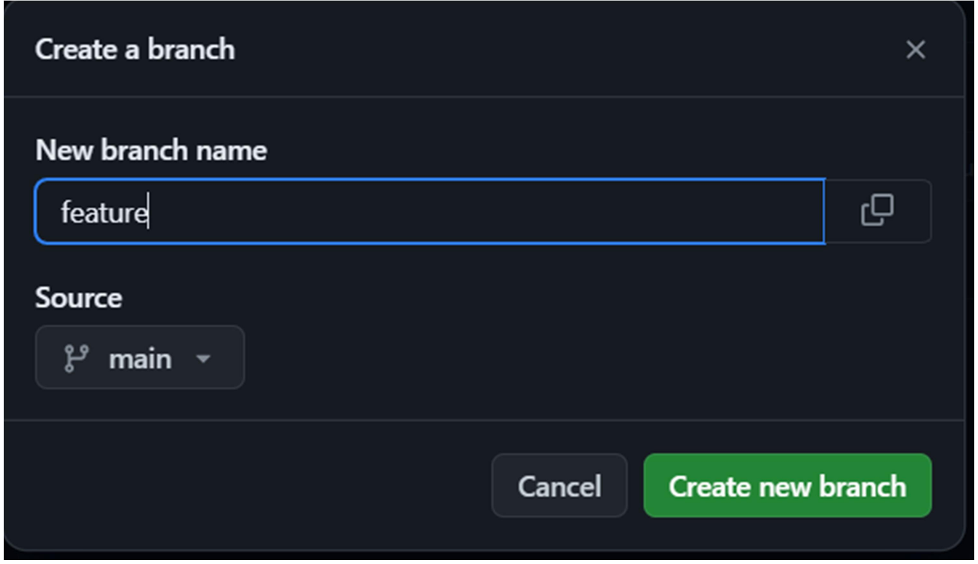
   Creating branches with GitHub and merging with the main branch.

2. **Task to be done:**
   Create branch of a repository and make changes and merge the file into main branch.

3. **Steps for experiment/practical:**

Using git hub:

1. Create a New Branch:
Navigate to your GitHub repository on the web.
Click on the "Branch: main" button near the top left of the page.
In the text box that appears, type a name for your new branch and press Enter.



Create a branch

New branch name: feature

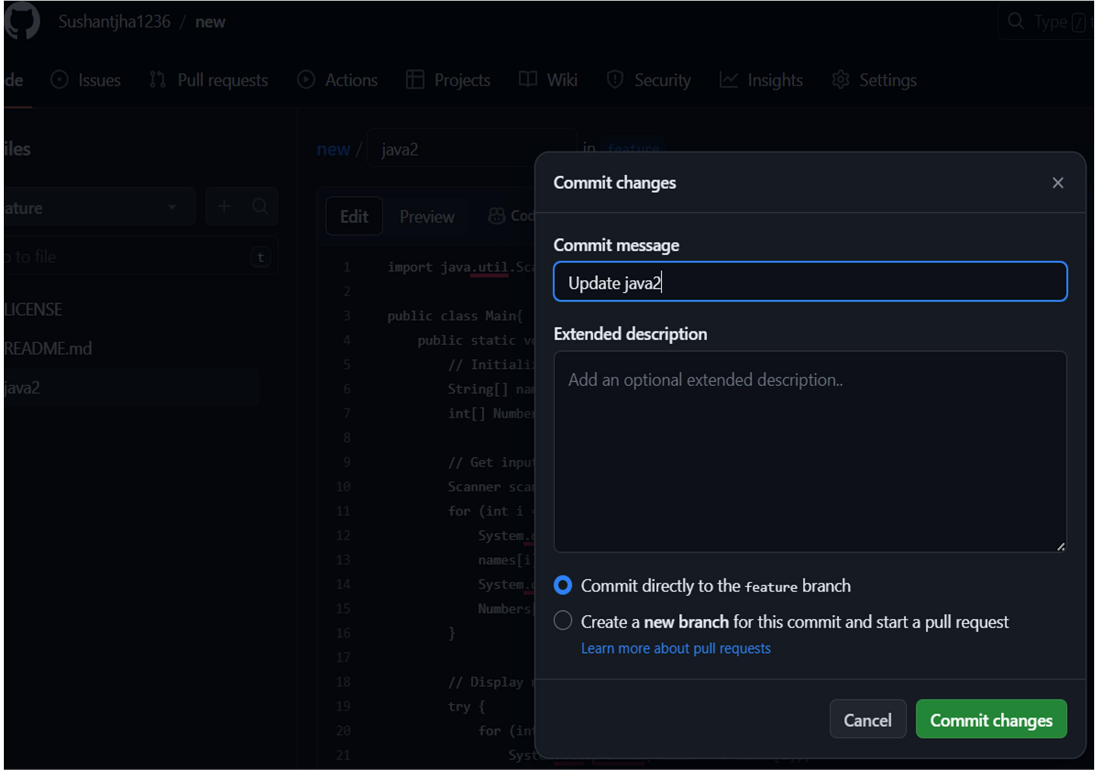
Source: main

2. Make Changes:

Navigate to the newly created branch by clicking on the branch name dropdown and selecting your branch**.**

3. Edit and Commit Changes:

Make changes to your code or add new files.
Scroll down to the "Changes" section.
Enter a commit message in the "Commit changes" section.
Click on the "Commit changes" button.



```
import java.util.Sc
 
public class Main{
    public static v
        // Initiali
        String[] na
        int[] Number
 
        // Get inpu
        Scanner sca
        for (int i
            System.
            names[i
            System.
            Numbers
        }
 
        // Display
        try {
            for (in
                Syst
```

Commit changes ×

Commit message

Update java2

Extended description

Add an optional extended description..

● Commit directly to the `feature` branch

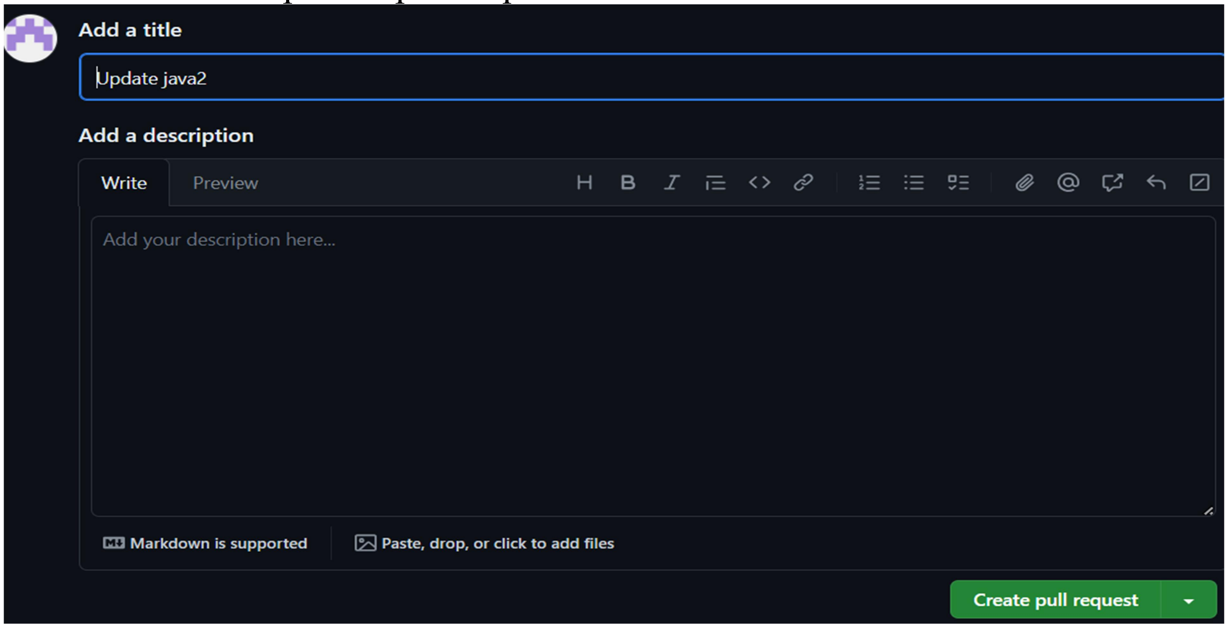
○ Create a **new branch** for this commit and start a pull request

Learn more about pull requests

4. Create a Pull Request:

Once you've committed your changes to the new branch, GitHub will display a message with a "Compare & pull request" button.

Click on the "Compare & pull request" button.



Add a title

Update java2

Add a description

Write | Preview

Add your description here...

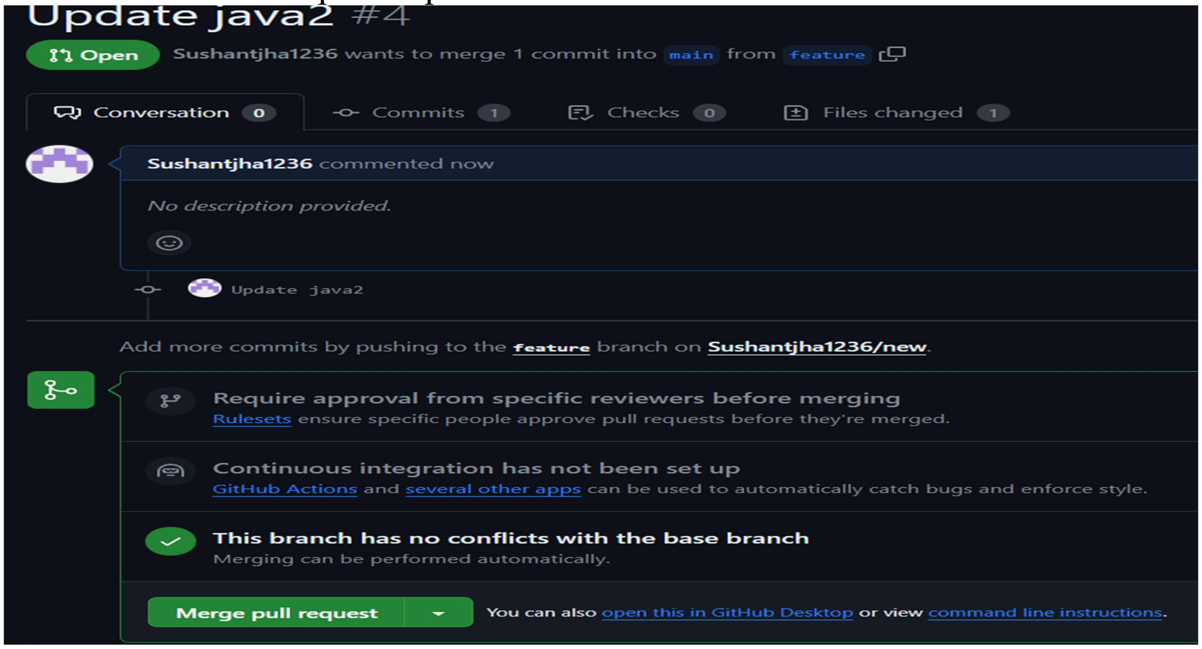
Markdown is supported | Paste, drop, or click to add files

Create pull request

5. Open a Pull Request:

You'll be taken to a page where you can review your changes. Add any additional comments if needed.

Click on the "Create pull request" button.



Update java2 #4

Open Sushantjha1236 wants to merge 1 commit into main from feature

Conversation 0 | Commits 1 | Checks 0 | Files changed 1

Sushantjha1236 commented now

No description provided.

Update java2

Add more commits by pushing to the feature branch on Sushantjha1236/new.

Require approval from specific reviewers before merging
Rulesets ensure specific people approve pull requests before they're merged.

Continuous integration has not been set up
GitHub Actions and several other apps can be used to automatically catch bugs and enforce style.

This branch has no conflicts with the base branch
Merging can be performed automatically.

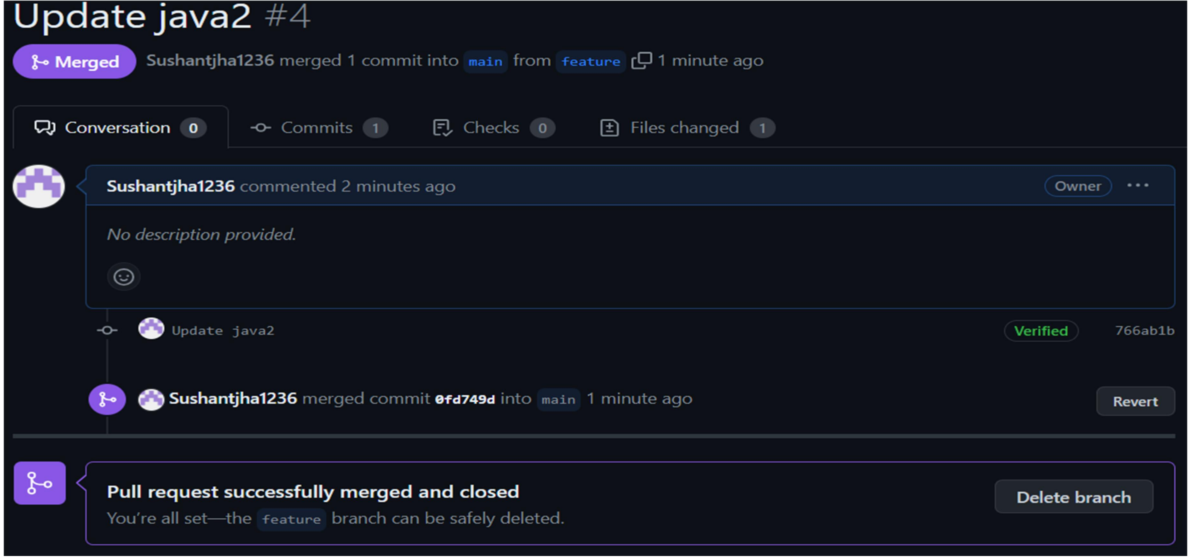
Merge pull request | You can also open this in GitHub Desktop or view command line instructions.

6. Merge the Pull Request:

After the pull request is reviewed and approved, you can merge it.
Click on the "Merge pull request" button.
Optionally, confirm the merge.

7. Confirm the Merge:

Once the pull request is merged, GitHub will prompt you to confirm the merge.
Confirm the merge.



Update java2 #4

Merged Sushantjha1236 merged 1 commit into main from feature 1 minute ago

Sushantjha1236 commented 2 minutes ago

No description provided.

Update java2 Verified 766ab1b

Sushantjha1236 merged commit 0fd749d into main 1 minute ago

Pull request successfully merged and closed

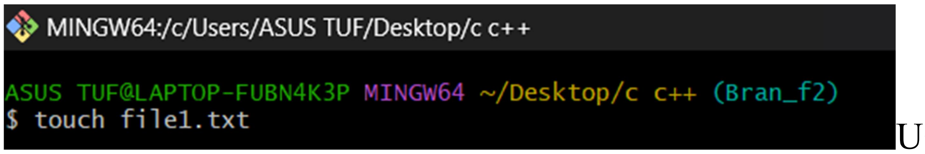
You're all set—the feature branch can be safely deleted.

8. Delete the Branch (optional):

After merging, GitHub will give you the option to delete the branch.
Choose to delete the branch if you no longer need it.

Using git bash:

1. **Create a new file**:

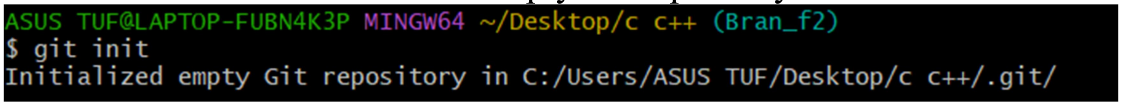


```
MINGW64:/c/Users/ASUS TUF/Desktop/c c++

ASUS TUF@LAPTOP-FUBN4K3P MINGW64 ~/Desktop/c c++ (Bran_f2)
$ touch file1.txt
```

Use a command to create a new file named file1.txt.

2. **Initialize a Git repository**:

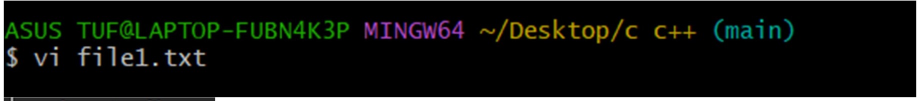
Run a command to initialize an empty Git repository in the current directory.



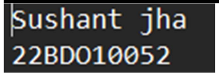
```
ASUS TUF@LAPTOP-FUBN4K3P MINGW64 ~/Desktop/c c++ (Bran_f2)
$ git init
Initialized empty Git repository in C:/Users/ASUS TUF/Desktop/c c++/.git/
```

3. Open and edit a file:

Use a text editor to open and make changes to the file1.txt.



```
ASUS TUF@LAPTOP-FUBN4K3P MINGW64 ~/Desktop/c c++ (main)
$ vi file1.txt
```



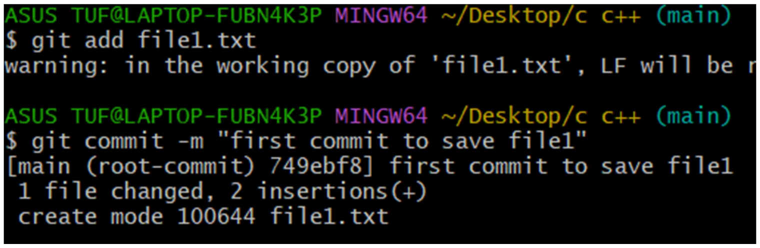
```
Sushant jha
22BD010052
```

4. Add changes to the staging area:

Add the changes made to file1.txt to the staging area.

5. Commit changes to the main branch:

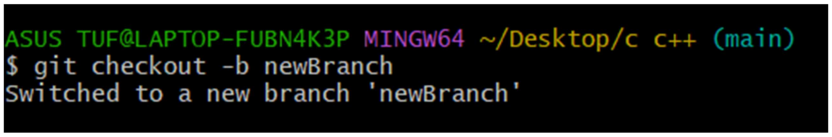
Commit the changes in the staging area with an appropriate commit message to the main branch.



```
ASUS TUF@LAPTOP-FUBN4K3P MINGW64 ~/Desktop/c c++ (main)
$ git add file1.txt
warning: in the working copy of 'file1.txt', LF will be r

ASUS TUF@LAPTOP-FUBN4K3P MINGW64 ~/Desktop/c c++ (main)
$ git commit -m "first commit to save file1"
[main (root-commit) 749ebf8] first commit to save file1
 1 file changed, 2 insertions(+)
 create mode 100644 file1.txt
```

6. Create a new branch:

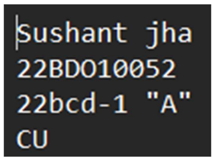
Switch to a new branch named newBranch.



```
ASUS TUF@LAPTOP-FUBN4K3P MINGW64 ~/Desktop/c c++ (main)
$ git checkout -b newBranch
Switched to a new branch 'newBranch'
```

7. Open and edit files in the new branch:

Use a text editor to open and make changes to files in the newBranch.



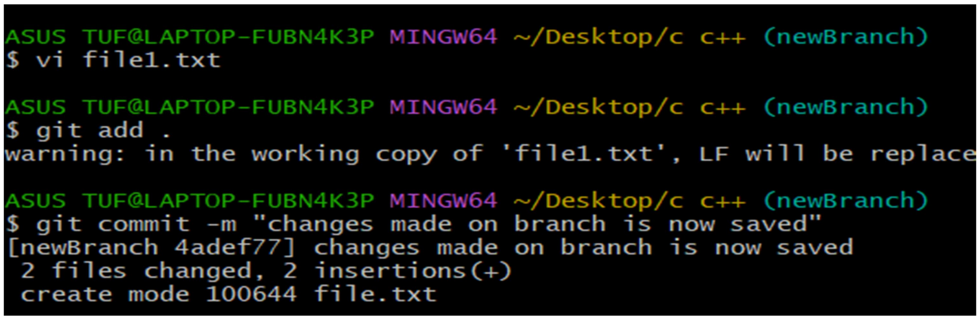
```
Sushant jha
22BD010052
22bcd-1 "A"
CU
```

8. Add changes to the staging area in the new branch:

Add all changes made in the working directory to the staging area.

9. Commit changes to the new branch:

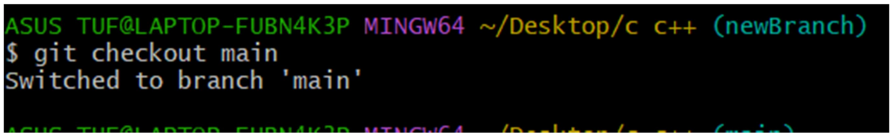
Commit the changes in the staging area with an appropriate commit message to the new Branch.



```
ASUS TUF@LAPTOP-FUBN4K3P MINGW64 ~/Desktop/c c++ (newBranch)
$ vi file1.txt

ASUS TUF@LAPTOP-FUBN4K3P MINGW64 ~/Desktop/c c++ (newBranch)
$ git add .
warning: in the working copy of 'file1.txt', LF will be replace

ASUS TUF@LAPTOP-FUBN4K3P MINGW64 ~/Desktop/c c++ (newBranch)
$ git commit -m "changes made on branch is now saved"
[newBranch 4adef77] changes made on branch is now saved
 2 files changed, 2 insertions(+)
 create mode 100644 file.txt
```

10.Switch back to the main branch:

Switch to the main branch.



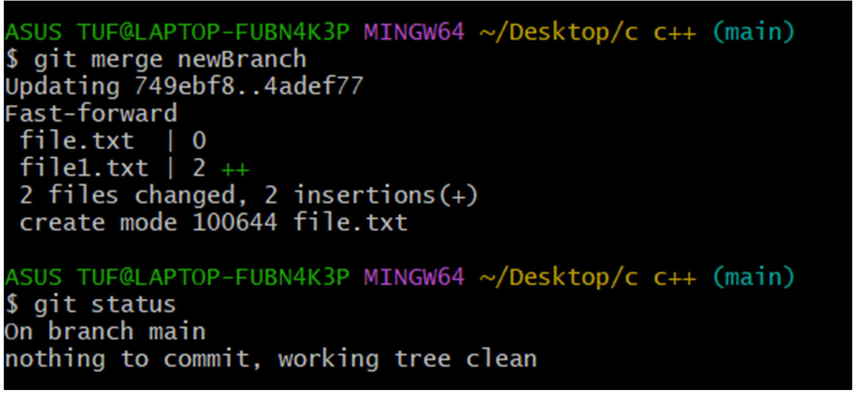
```
ASUS TUF@LAPTOP-FUBN4K3P MINGW64 ~/Desktop/c c++ (newBranch)
$ git checkout main
Switched to branch 'main'

ASUS TUF@LAPTOP-FUBN4K3P MINGW64 ~/Desktop/c c++ (main)
```

11.Merge changes from the new branch to main:

Merge the changes from new Branch into the main branch.

12.Check the status of the repository:

Verify that the working tree is clean and there is nothing to commit in the main branch.



```
ASUS TUF@LAPTOP-FUBN4K3P MINGW64 ~/Desktop/c c++ (main)
$ git merge newBranch
Updating 749ebf8..4adef77
Fast-forward
 file.txt  | 0
 file1.txt | 2 ++
 2 files changed, 2 insertions(+)
 create mode 100644 file.txt

ASUS TUF@LAPTOP-FUBN4K3P MINGW64 ~/Desktop/c c++ (main)
$ git status
On branch main
nothing to commit, working tree clean
```

Learning outcomes (What I have learnt):

1. Understanding Git Workflow

2. Use of Branching and merging.

3. Creating branch on Git bash and git hub.

4. Committing changes.

5. Working with staging area.

Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.			
2.			
3.			