GITHUB COLLABORATION LAB: SIMPLE CALCULATOR

Lab Objective

Learn basic GitHub collaboration by building a simple calculator app with your team. Each team member will contribute a different operation.

Team Roles

- **Student 1 (Repository Owner)**: Creates repository, initial project, and merges pull requests
- **Student 2**: Implements addition (+)
- **Student 3**: Implements subtraction (-)
- **Student 4**: Implements multiplication (×)
- **Student 5**: Implements division (÷)

Part 1: Student 1 - Repository Setup

Step 1: Create Repository on GitHub

1. Go to GitHub and sign in
2. Click the + icon (top right) → **New repository**
3. Repository name: simple-calculator
4. Description: A simple calculator built by our team
5. Select **Public**
6. Check **Add a README file**
7. Click **Create repository**

Step 2: Add Collaborators

1. In your repository, click **Settings → Collaborators**
2. Click **Add people**
3. Add all 4 team members by their GitHub usernames
4. They will receive email invitations (they must accept!)

Step 3: Create Initial Project Locally

Create a new folder and files:
mkdir simple-calculator
cd simple-calculator
Create index.html:
Copy the code from Initial Code.txt and paste in HTML

Step 4: Push Initial Code to GitHub

```
# Initialize git repository
git init

# Add all files
git add .

# Commit files
git commit -m "Initial calculator project setup"

# Connect to GitHub repository (replace YOUR_USERNAME)
git remote add origin https://github.com/YOUR_USERNAME/simple-calculator.git

# Rename branch to main (if needed)
git branch -M main

# Push code to GitHub
git push -u origin main
```

Your initial project is now on GitHub!

Part 2: Students 2-5 - Clone and Develop Features

Step 1: Accept Invitation

1. Check your email for GitHub collaboration invite
2. Click **Accept invitation**

Step 2: Clone Repository

```
# Clone the repository (replace OWNER_USERNAME)
git clone https://github.com/OWNER_USERNAME/simple-calculator.git

# Enter the project folder
cd simple-calculator
```

Step 3: Create Your Feature Branch

Each student creates their own branch:

```
# Student 2 (Addition):
git checkout -b feature-addition

# Student 3 (Subtraction):
git checkout -b feature-subtraction

# Student 4 (Multiplication):
```

git checkout -b feature-multiplication

Student 5 (Division):
git checkout -b feature-division

Step 4: Implement Your Operation

Open script.js and replace your assigned function:

Student 2 - Addition Function:

Copy the code from Addition Function.txt

Student 3 - Subtraction Function:

Copy the code from Subtraction Function.txt

Student 4 - Multiplication Function:

Copy the code from Multiplication Function.txt

Student 5 - Division Function:

Copy the code from Division Function.txt

Step 5: Commit and Push Your Changes

Check what files you modified
git status

Add your changes
git add script.js

Commit with a descriptive message
Student 2:
git commit -m "Implement addition operation"

Student 3:
git commit -m "Implement subtraction operation"

Student 4:
git commit -m "Implement multiplication operation"

Student 5:
git commit -m "Implement division operation"

Push your branch to GitHub
git push origin YOUR_BRANCH_NAME

Example:
Student 2:
git push origin feature-addition

Step 6: Create Pull Request

1. Go to the repository on GitHub
2. You'll see a notification: **"Compare & pull request"** button - Click it!
3. Add a title: Add [operation name] functionality
4. Add description: This PR implements the [operation] function
5. Click **Create pull request**
6. Notify Student 1 that your PR is ready for review

Part 3: Student 1 - Merge Pull Requests

Step 1: Review Pull Requests

1. Go to your repository → **Pull requests** tab
2. Click on each pull request
3. Review the changes in the **Files changed** tab
4. Check that the code looks correct

Step 2: Merge Pull Requests

1. If everything looks good, click **Merge pull request**
2. Click **Confirm merge**
3. Click **Delete branch** (optional, to keep repository clean)
4. Repeat for all 4 pull requests

Step 3: Test the Complete Calculator

1. Go to your local repository folder
2. Pull the latest changes:

git checkout main
git pull origin main

3. Open index.html in your browser
4. Test all operations to ensure they work!

Part 4: All Students - Get Final Version

Update Your Local Repository

Switch to main branch
git checkout main

Pull latest changes
git pull origin main

Test the Calculator

1. Open index.html in your browser
2. Test all four operations (+, -, ×, ÷)
3. Verify everything works correctly

Common Git Commands Reference

```
# Check current status
git status

# Check current branch
git branch

# Switch branches
git checkout branch-name

# Create and switch to new branch
git checkout -b new-branch-name

# Add files to staging
git add filename
git add .  # adds all files

# Commit changes
git commit -m "Your message"

# Push to GitHub
git push origin branch-name

# Pull latest changes
git pull origin branch-name

# View commit history
git log --oneline
```

Troubleshooting

Problem: fatal: not a git repository

- **Solution**: Make sure you're inside the project folder

Problem: Permission denied

- **Solution**: Make sure you've accepted the collaboration invitation

Problem: Conflict during merge

- **Solution**: This shouldn't happen if everyone edits different functions. If it does, ask the instructor for help.

Problem: Everything up-to-date when pushing

- **Solution**: Make sure you've committed your changes first

Problem: Can't push changes

- **Solution**: Make sure you're on your feature branch, not main

---

Learning Outcomes

After completing this lab, you should be able to:

- Create a GitHub repository
- Clone a repository
- Create and switch branches
- Make commits with meaningful messages
- Push code to GitHub
- Create pull requests
- Review and merge pull requests
- Pull latest changes from remote repository
- Collaborate with team members using Git

Submission

Take screenshots of:

1. Your final calculator working in the browser
2. The GitHub repository showing all merged pull requests
3. Your terminal showing the git log

Submit these screenshots along with your repository URL.