CSC263 - Software Engineering
Assignment 3 – Version Control

What is Git?

1. Git is a popular version control system. It was created by Linus Torvalds in 2005. It is used for:
   - Tracking code changes
   - Tracking who made changes
   - Coding collaboration
2. What does Git do?
   - Manage projects with **Repositories**
   - **Clone** a project to work on a local copy
   - Control and track changes with **Staging** and **Committing**
   - **Branch** and **Merge** to allow for work on different parts and versions of a project
   - **Pull** the latest version of the project to a local copy
   - **Push** local updates to the main project

Download and Install Git

1. Download Git here
2. Click on file to install
3. Chose location (C:\Program Files\Git is the default)
4. Hit Next on all subsequent screens until Finish
5. Open up Command shell (cmd)
6. C:\pathname\>cd\ (this will put you on root C:\>)
7. C:\>git –version (to check if Git is properly installed)

Configure Git

1. C:\>git config –global user.name "Type a user name here"
2. C:\>git config –global user.email "Type an email here"
3. C:\>mkdir myProject (this will create a folder "myProject" for your work)
4. C:\>cd myProject (this will set the folder "myProject" as the working directory)
5. C:\myProject>git init (to create Git repository)

CHOSE TO DO THE FOLLOWING IN GIT OR GITHUB

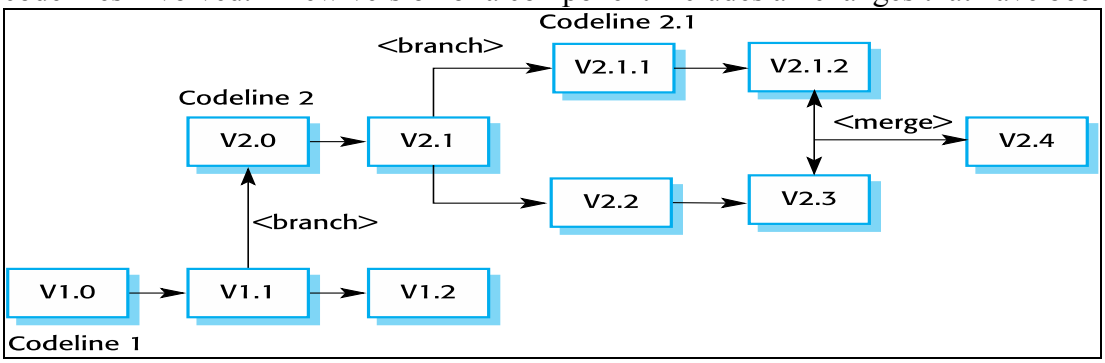
Manage the repository – Using Git

1. Add any file to myProject using any text editor, or you can copy paste an image …
2. Example:
   - C:\myProject>Notepad index.html (write any html script)
   - C:\myProject>dir (to list the files and folders that are inside "myProject" folder)
   - Index.html is called Untracked file (it is in "myProject" folder but not added to the repository.
3. C:\myProject>git add index.html (index.html is now added to the repository, we say **staged**, **tracked** which means it is ready to be **committed** to the repository)

4. C:\myProject>git restore –staged name-of-the-file(to cancel the added file before commit, in this case the file index.html. If you try this, you have to add it again before commit)
5. C:\myProject>git commit –m “any message” (all files “added” are committed)

Branching and Merging

A codeline is a set of versions of a software component and other items on which that component depends.
Branching is the creation of a new codeline from a version in an existing codeline. The new codeline and the existing codeline may then develop independently by different independent developers.
Merging is the creation of a new version of a software component by merging separate versions in different codelines. These codelines may have been created by a previous branch of one of the codelines involved. A new version of a component includes all changes that have been made



Git Branch (new/separate version of the main repository)

1. C:\myProject>git branch name-of-the-branch (Like; git branch new-images)
2. C:\myProject>git checkout name-of-the branch (Like; git checkout new-images, move current workspace from current branch “master” to the new branch “new-images”)
3. Add new files and/or make changes to existing files and add them to the repository by:
4. C:\myProject>git add –all (all files are staged, now commit them to the new branch)
5. C:\myProject>git commit –m “text message”

Git Merge

1. We have added files and changes to existing files in “branch” that are not in “master”, we want to merge the “master” with the “branch”
2. C:\myProject>git checkout master (master now is the current working space)
3. C:\myProject>git merge name-of –the-branch (we named it “new-images” in Git Branch)
4. C:\myProject>git branch -d name-of-the-branch (because “master” and “new-images” are the same after merging, you may want to delete the branch)

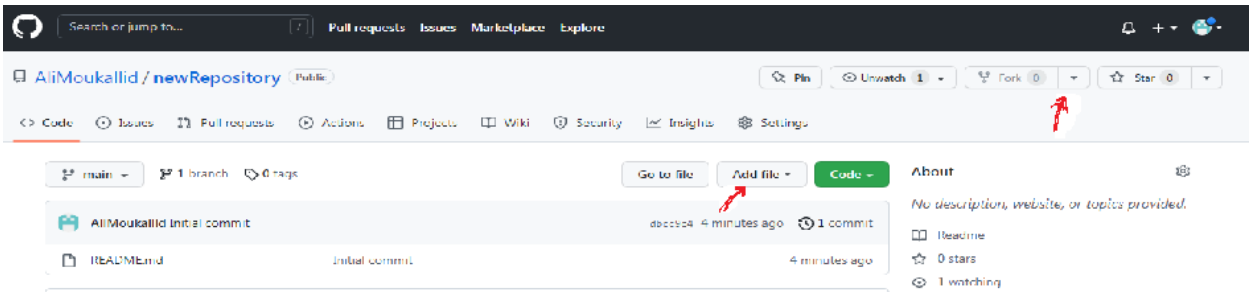
Git Push local file to remote GitHub

1. After commit changes to files in local repository, Push it to GitHub
2. C:\myProject>git remote add origin https://github.com/ ”The rest of your GitHub URL”
3. C:\myProject>git push –u –f origin master

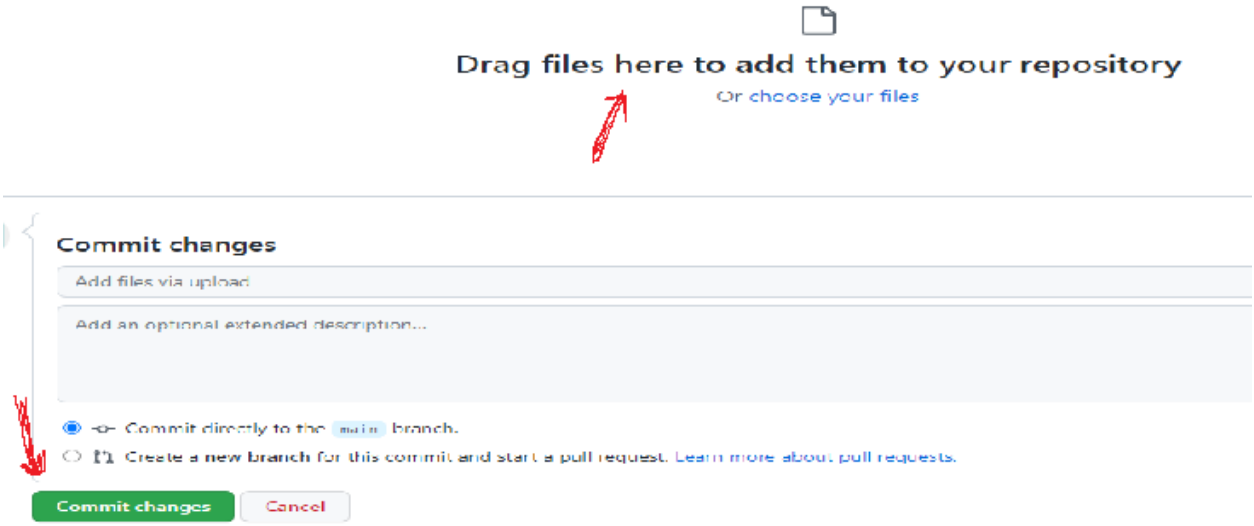
Manage the Repository – Using GitHub

If you only want to keep track of your code locally, you don't need to use GitHub. But if you want to work with a team, you can use GitHub to collaboratively modify the project's code.

1. Download and install GitHub here
2. Github.com (GitHub homepage) -> email -> sign up
3. Start GitHub session
4. Create new repository (click on + sign at top right of page)
5. Click on AddFile and choose create or upload (as needed)

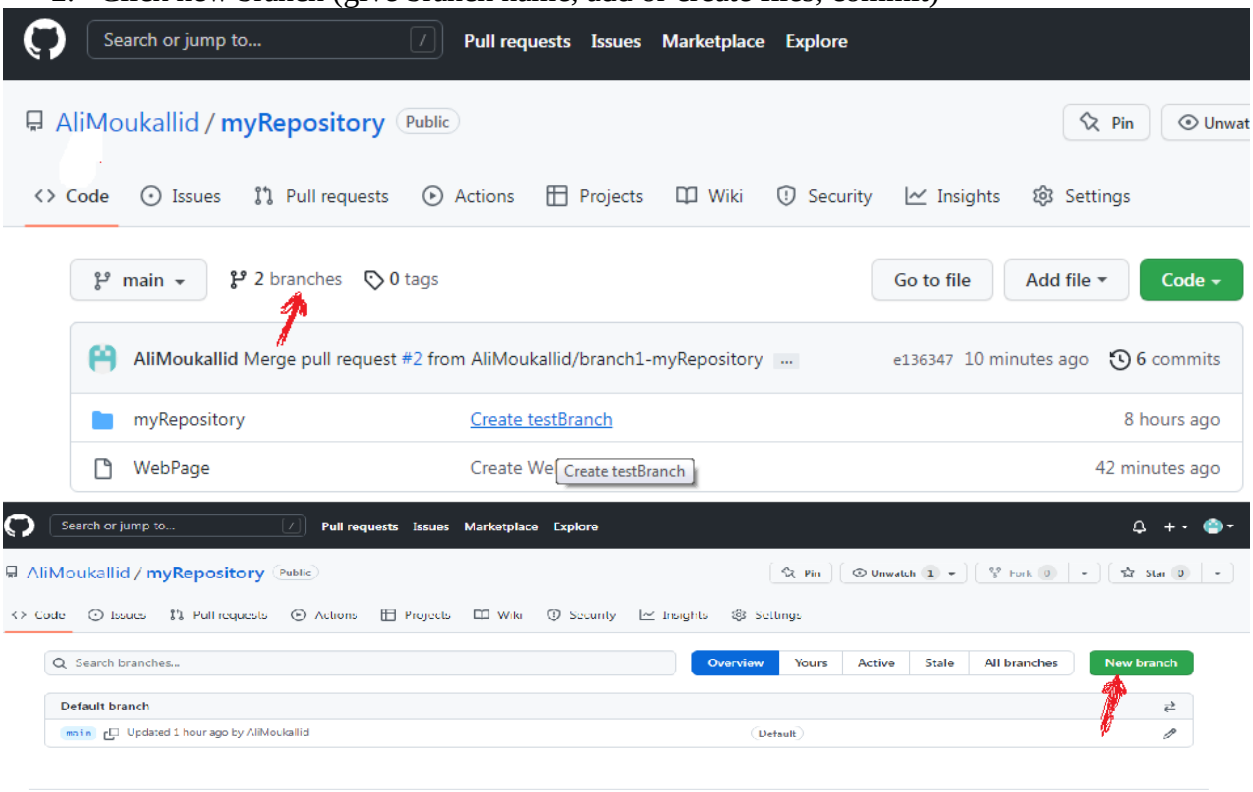


6. Drag and drop your files
7. Commit Changes



Add Branch

1. Click branches
2. Click new branch (give branch name, add or create files, commit)

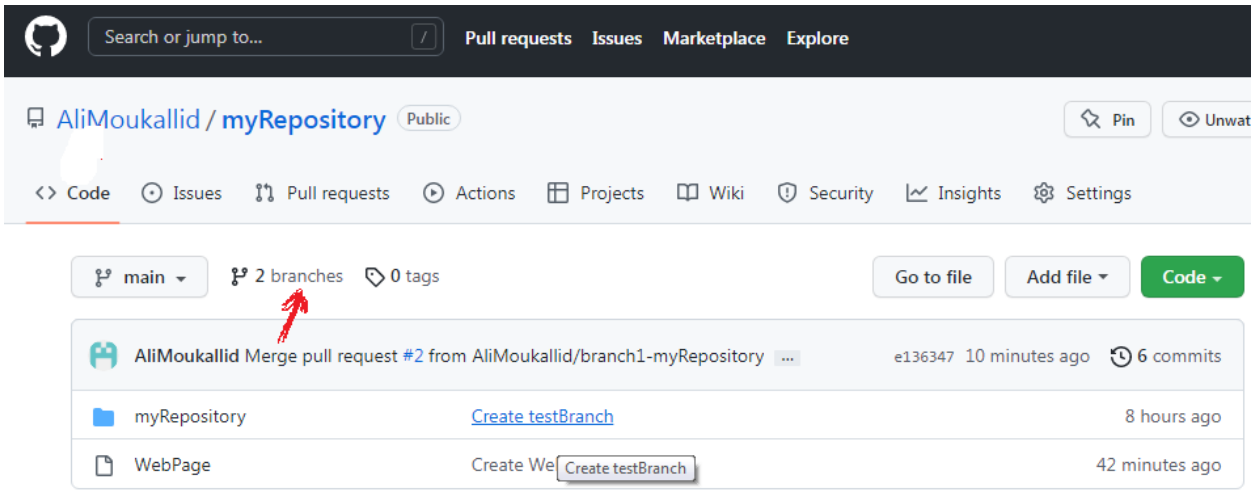


Pull Request

1. While in a branch look for "compare and pull request" or "pull request" buttons
2. Once no conflicts are found, you can merge branch with main
3. Chance to delete the branch or you can delete it later

Delete a Branch

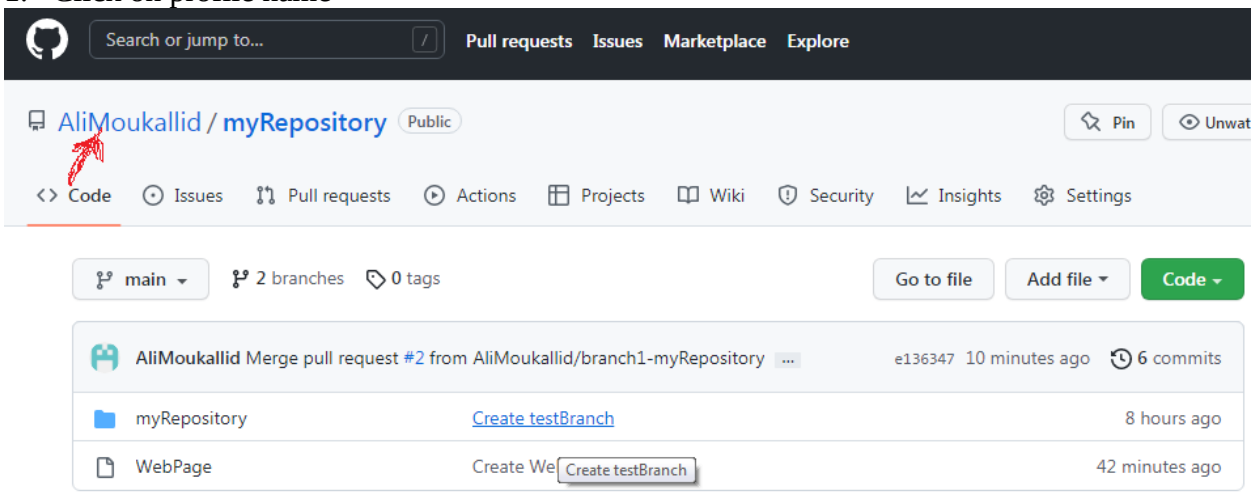
1. Click on Branches



2. Click remove button (to the right of the branch name)

Switch between Repositories

1. Click on profile name



2. Chose repository

Git on Eclipse here (click OK if it displays error message)