

Git and L^AT_EX Worksheet

Justin Li

justinnhli@oxy.edu

Occidental College

1 Instructions

This worksheet is due March 1, 2025 at midnight, to be submitted as a GitHub repository URL to Canvas. The repository should contain all files required to compile this worksheet with your answers. You should only change this `document.tex` file and the `references.bib` file; do not change any other file in this starting repository. You should not use any additional packages, and are not allowed to use the `\usepackage{}` command. Additionally, the output should be formatted correctly: your answers should be appropriately nested under the questions, command-line commands should be in monospace, and images should be positioned appropriately.

2 Git Questions

2.1 General questions

1. What is a version control system? Why are they useful?

It helps developers collaborate, as it tracks changes made to files over time, where you can go back to previous versions and manage different iterations of code. It makes it easy to manage mistakes as you can recover past versions of the code.

2. What is the difference between git and GitHub?

Git is a VCS that can track and manage changes locally on your computer. It works offline and is a command-line tool. GitHub hosts Git repositories as a cloud-based platform. It has a graphical interface like push pull requests, issues, and code reviews. Competing services are GitLab and Bitbucket.

3. What is a repository?

It's a storage location for your project's files and their revision history. It contains all the files, folders, and metadata (like commit history) for a project. They can be local or remote (like on GitHub).

4. What is a commit?

A commit is a specific point in your project at a specific time. It records changes to files in your repository, along with a message describing the changes. Each commit has a unique identifier (hash) and is part of the repository's history.

5. What is the commit graph?

The commit graph is a visual representation of the history of commits in a repository. It shows the relationships between commits, branches, and merges. Each commit is a node, and the lines between nodes represent the flow of changes. This graph helps you understand how the project has evolved over time.

6. What is your preferred local git client (eg., command line, GitHub Desktop, GitKraken, etc.)?

I'd say GitHub Desktop but for storage the command line.

2.2 Local Usage

1. What is the difference between adding a file to the staging area and committing a file?

Adding a file to the staging area (`git add file`), prepares the file to be included in the next commit. The staging area is a "waiting room" to review and organize changes before saving them permanently.

Committing a File: When you commit a file it creating a permanent snapshot of the changes in the staging area. This snapshot is saved to the repo's history with a hash and a commit message.

2. What is a commit message, and why is it important for them to be meaningful?

It describes the changes made in a commit. It helps you and others understand what and why the changes were made. It helps pinpoint specific changes when debugging and can make it easier to maintain and refactor the code over time.

3. Starting with an empty repository, what sequence of commands/actions would result in the following commit graph? You may give a sequence of `git` commands, or describe (with screenshots) how you would

do this in your preferred graphical git interface.

A---B---C---D

```
git init echo "Initial commit" & file.txt git status git add
. git commit -m "A" echo "Second commit" & file.txt
git add file.txt git commit -m "B" echo "Third commit"
& file.txt git add file.txt git commit -m "C" echo "Forth
commit" & file.txt git add file.txt git commit -m "D" git
log
```

4. If you are currently at commit D above, how would you recover code from commit B? What sequence of commands/actions would let you do so? You may give a sequence of `git` command-line commands, or describe (with screenshots) how you would do this in your preferred graphical git interface. Assume the commit hashes are AAAAAA..., BBBBBB..., etc. `git checkout BBBBBB git commit -m "Revert to code from commit B"`
5. Imagine you created a git repository for your project, but only commit your changes once a week on Sundays. You got your code working on Tuesday, but then broke your code on Friday. What can you do to get the working version of your code back? Click "Stash All Changes" to save your current changes, go to history and right click on the Tuesday commit and create a branch from it. Click stashed changes to apply the changes to the new branch.

2.3 Branching and Merging

1. What is a branch? Why are they useful? It captures change, and its helpful to debug and experiment with new ideas and features.
2. Starting with an empty repository, what sequence of commands/actions would result in the following commit graph? You may give a sequence of `git` command-line commands, or describe (with screenshots) how you would do this in your preferred graphical git interface.

A---B---C---D
 \
 E---F

```
git checkout [hash] -b new-branch echo "Commit E"
&& file.txt git add file.txt git commit -m "E" echo
"Commit F" && file.txt git add file.txt git commit -m
"F"
```

3. Why is a merge? Why are they useful? Merge is when you combine multiple commits into the original/default branch.
4. Imagine you are currently at commit D above. What sequence of commands/actions would result in the following commit graph? You may give a sequence of

`git` commands, or describe (with screenshots) how you would do this in your preferred graphical git interface.

A---B---C---D---G
 \ /
 E---F---/

```
git merge new-branch -m "Merge branch 'new-branch'
into main" add file.txt git commit -m "Resolved merge
conflict and merged 'new-branch' into main" git log
--oneline --graph --all
```

5. What is a merge conflict? When do they occur?
 A merge conflict happens when Git is unable to automatically combine changes from two different branches because they have conflicting edits to the same file or lines of code. Merging the two branches needs manual editing.
6. Starting with an empty repository, despite a sequence of commands/actions that would result in a merge conflict. Include the exact edits and `git` commands or screenshots of the graphical git interface. Include the output or screenshot that shows the resulting merge conflict.

2.4 Remotes

1. What is a remote? A remote is a repository that is stored on a server (like GitHub itself) and acts as a central location where multiple developers can share their code by pushing and pulling changes, essentially allowing them to collaborate on a project.
2. What does pushing and pulling do? Push sends commits and asks them to update their branch, so it has to be correct on their end. Pull grabs any changes from the GitHub repository and merges them into your local repository.
3. Imagine you created a git repository for your project on your laptop and commit regularly, but only push your code to GitHub once a week on Sundays. Your laptop caught on fire on Friday. What can you do to get your code back?
 You can check Github and see the code you pushed last Sunday, but it won't have the progress you made from Sunday to Friday. You can check the stash list to apply any stashes or use `reflog` to find any recent commits.

3 L^AT_EX

Find a source of each of the following types and add it to `references.bib`, with the appropriate data. Your sources do not have to relate to your project. Looking at Overleaf (2025) [1] and Wikipedia (2025) [2] may be helpful,

- a journal article
- a conference article
- a PhD or Master's thesis
- an article in an edited popular media venue (newspaper, magazine, etc.)
- a book
- a chapter of a book
- a YouTube video
- a piece of technical documentation (e.g., a programming language reference, and API documentation, etc.)

Additionally, in your own words, explain the difference between `\cite{}` and `\textcite{}`. When should they each be used? Demonstrate your answers by using one of them with each of your references from above. The first generates an inline citation in parenthesis, usually after your statement. The second one shows the author's name in the text.

References

- [1] Overleaf. *Bibliography Management with bibtex*. accessed 2025-02-11. 2025. URL: https://www.overleaf.com/learn/latex/Bibliography_management_with_bibtex.
- [2] Wikipedia. *BibTeX*. accessed 2025-02-11. 2025. URL: <https://en.wikipedia.org/wiki/BibTeX>.