

Announcements

- CATME Survey Due Date 8/31.
- Office Hours have begun – in CCB 267
- Redeem your GCP credits ASAP
 - Always remember to shut down/delete your cluster or buckets after a session of using GCP
- Project Planning Assignment will release on August 31
- Project Team Mentors will be assigned this week
- Code Review Assignment Released today
 - Due October 12th with Project 1

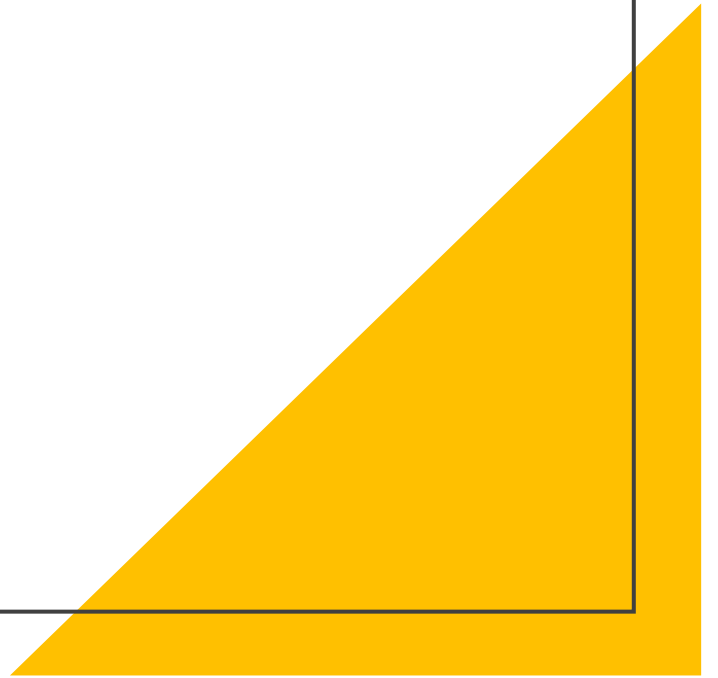
CS3300 Introduction to Software Engineering

Lecture 02: Project Description; Tools of the Trade #1

Version Control Systems, GIT, Code Review

Dr. Nimisha Roy ▶ nroy9@gatech.edu

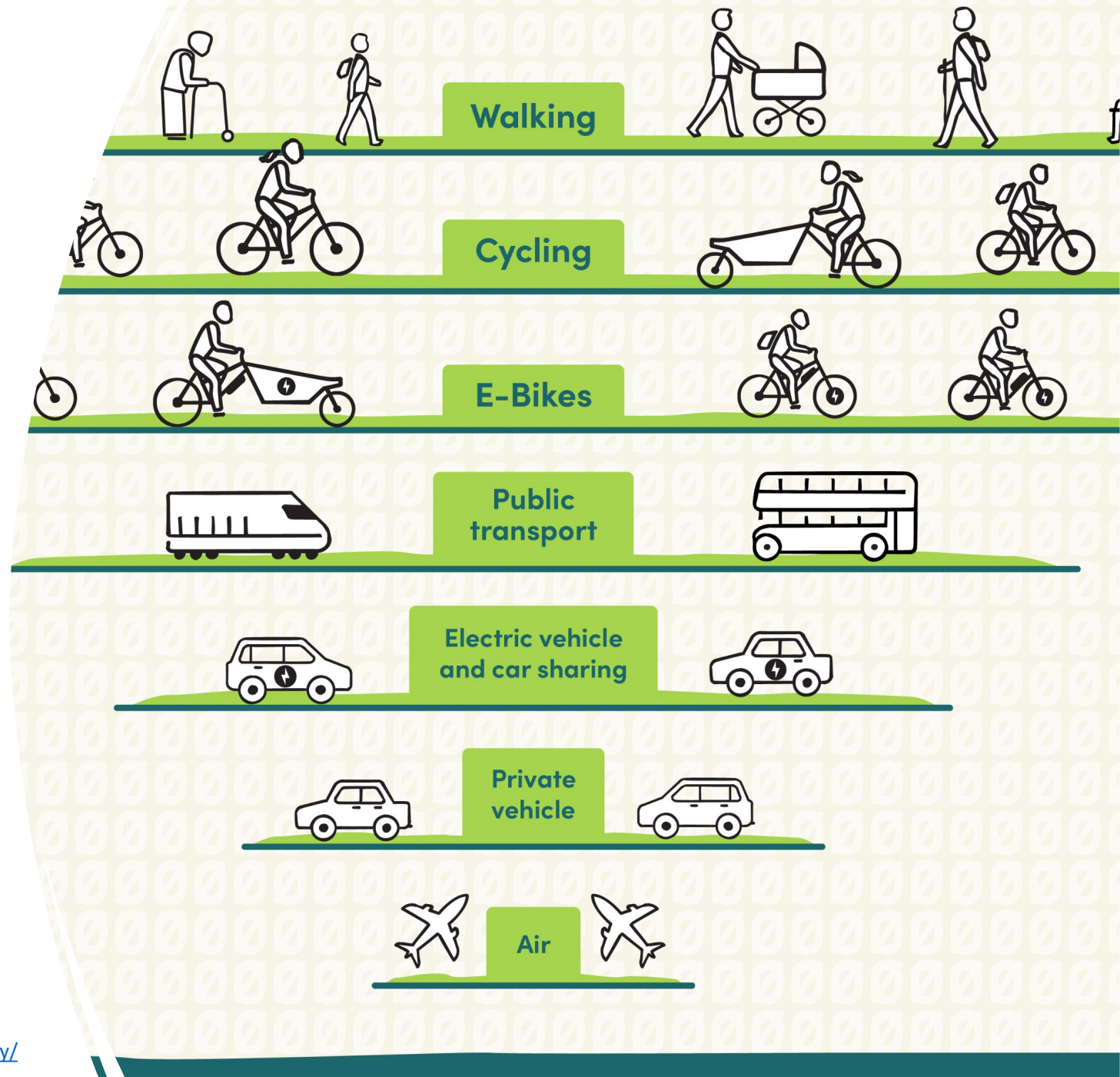
Project Description



Sustainable Commute Finder Web Application using SpringBoot and Google Cloud Platform

Problem Statement

Objective: Help users determine the most sustainable commuting option by considering carbon footprint of various transport modes



Problem Statement

You may have more features
for some extra credit

Compulsory Features:

1. Landing Page (Sign up + Login)

2. Route Selection:

- Allow users to input starting and ending locations.
- Offer multiple commute options (walking, cycling, public transport, car, carpool, electric vehicle, etc.)
- Should have a map component

3. Carbon Footprint Estimation:

- For each route, display the estimated carbon footprint.
- Allow users to adjust variables (e.g., car fuel efficiency, number of passengers in carpool, type of public transport).
- Show comparative analysis – e.g., "Choosing to cycle saves X kg of CO₂ compared to driving alone."

4. Interactive Environmental Impact:

- Highlight the positive environmental impact of choosing a sustainable commute option with visual representations, e.g., "This is equivalent to planting X trees."

Carbon Footprint Estimation

- **Static Values:**

- Pre-defined values based on research for average emissions.
 - Walking, Cycling: Essentially zero emissions.
 - Gasoline Car: Roughly 2.31 kg of CO₂ per liter of gasoline.
 - Electric Car: Depends on the source of electricity but can use average values.
 - Bus, Trains, Trams: Can use average values per passenger-km.

- **Dynamic Adjustments:**

- Allow users to input specific values:
 - Car fuel efficiency.
 - Electricity source for electric vehicles (solar, wind, coal, etc.).
 - Number of passengers (for carpools or if a user is driving others).

- **Calculations:**

- **Distance:** Use the chosen route's distance.
- **Emission Factor:** The amount of CO₂ emitted per km or mile for the chosen transport.
- **Carbon Footprint** = Distance x Emission Factor

Resources

- **Online Carbon Footprint Calculator:** [Link](#)
 - **Features:**
 - Calculates individual, business, and event carbon footprints.
 - Breakdown by home, flight, car, motorbike, bus & rail, and secondary footprints.
 - Allows users to offset their carbon footprint by contributing to environmental projects.
 - **Usage:**
 - Understand the breakdown of emissions sources.
 - Explore how different variables (e.g., car type, fuel, distance) impact carbon emissions.
- **Carbon Interface API:** [Carbon Interface](#)
 - **Features:**
 - Allows estimation of carbon footprints for various activities including transportation.
 - Returns carbon footprint in CO2 equivalent units.
 - **Usage:**
 - Directly integrate this API into the application to get CO2 emission data for various transport modes.

Resources

- **OpenStreetMap and GraphHopper:**
- **GraphHopper:** [Link](#)
- **Features:**
 - Uses OpenStreetMap data for routing.
 - Provides distance and route details for various transport modes.
- **Usage:**
 - To extract route data and distances.
 - Once distance is obtained, combine with carbon emission factors (possibly from Carbon Interface) to estimate emissions.

Technology Suggestions

GOOGLE CLOUD PLATFORM*

Offers services for compute, storage, networking, big data, machine learning and the internet of things (IoT), as well as cloud management, security and developer tools.

BACKEND SERVER ARCHITECTURE

- **SpringBoot*** – Framework for developing the backend server architecture and to run the server application easily
- **Maven*** - Used for dependency injection and management as a Java package manager. Provided as a part of central repository in Spring
- Mockito – Test stub method for unit testing SpringBoot application
- Eclipse/IntelliJ - IDE

DATABASE

- GCP DataStore
- MySQL
- Google Storage

*** Must use**

Technology Suggestions

API

- Google Places API - To implement an interactive, searchable map
- Google Distance Matrix API/ OpenStreetMap API
- Sustainability related APIs

FRONTEND

- Thymeleaf - Java template engine for both web and standalone environments. Set of Spring integrations.
- Leaflet - Open-source JavaScript library for creating interactive maps
- jQuery - JS Library for implementing HTML document traversal and manipulation, event handling, animation, and Ajax.
- Bootstrap – CSS framework to create modern websites and web apps
- React.JS - open-source JavaScript library used for building user interfaces for single-page applications

VERSION CONTROL

GitHub*

*** Must use**

Some Considerations/Functionalities

Password Security

- Salting & Hashing Passwords
- JWT – Validity of Tokens (optional)

Database

- Google DataStore is recommended.
- Google storage has some problem: 2 people simultaneous logging in may cause overwrite

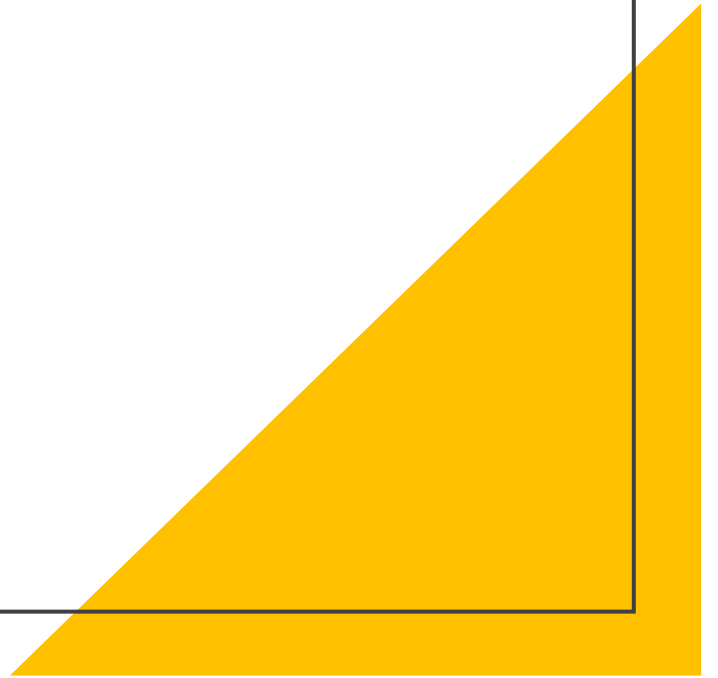
Webjars

Use webjars - client-side web libraries packaged into JAR (Java Archive) files to easily manage web dependencies

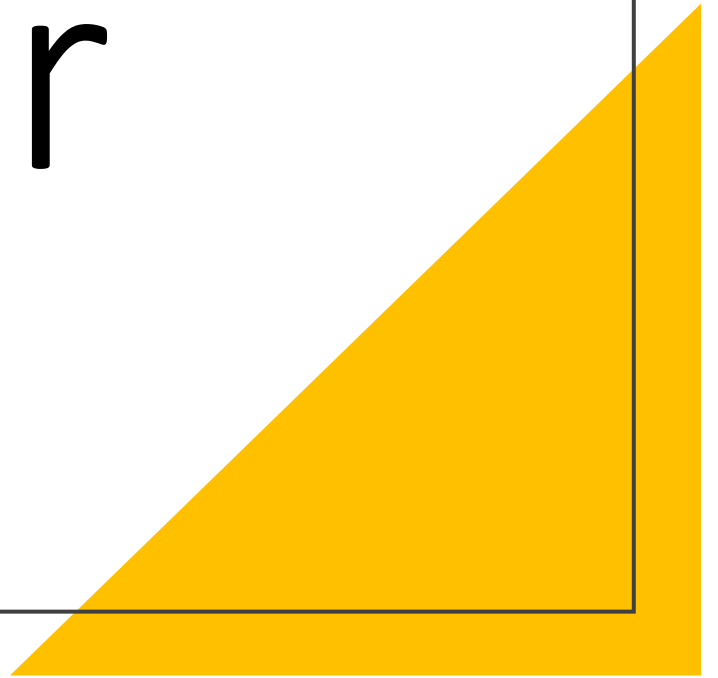
Testing

Important. Modules Included in Google App Engine.

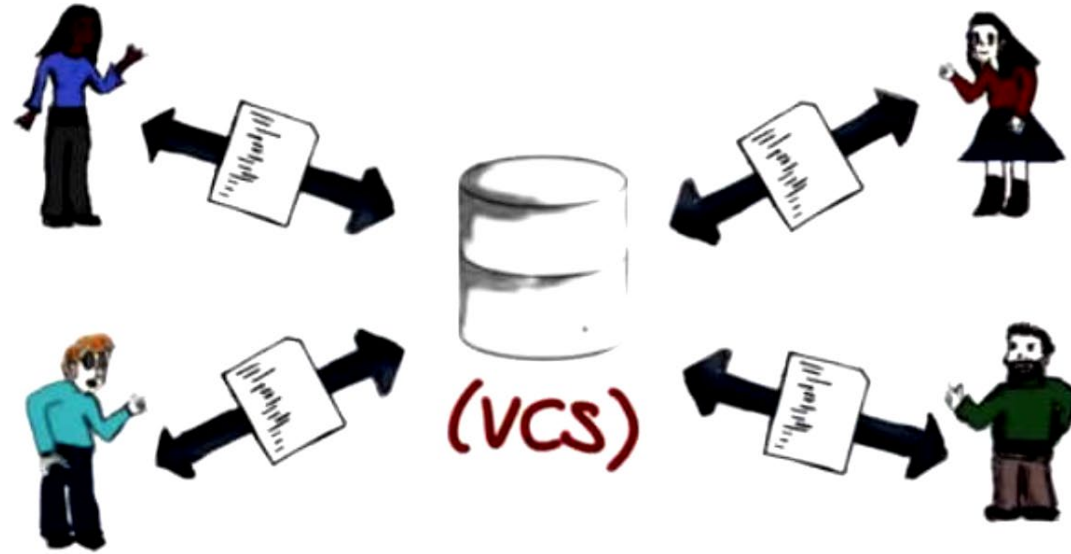
Project Planning Assignment



Git Refresher



What are Version Control Systems?



- A tool that software developers use for keeping track of revisions of their project
 - snapshots of your project over time.
 - Documents, source files etc.
- Most obvious benefits:
 - Option to go back and revisit
 - Collaborate with multiple people

Importance

- **Enforce Discipline:** Manages process by which control of items passes from one person to another
- **Archive versions:** store subsequent versions of source-controlled items
- **Maintain Historical Information:** Author of a specific version; date and time of a specific version; etc. Retrieve and compare.
- **Enables Collaboration:** share data, files and documents
- **Recover from accidental edits/revisions**
- **Conserve Disk Space:** centralizing the management of the version.
 - Instead of having many copies spread around, one or a few central points where these copies are stored
 - efficient algorithms to store changes, so keep many versions without taking up too much space.

Don'ts in VCS

- Adding Derived Files
 - E.g., executable file derived from compiling a set of source codes
 - No reason to add it
- Adding bulky binary files
 - Try to keep them local
- Creating a local copy of files/tree of files
 - Don't do this!!
 - Useless, risky, confusing
 - Trust the version control system

Types of VCS

- Centralized VCS
 - Single centralized repository on which you are committing files
- Decentralized/ Distributive VCS
 - Sort of local repository on which you can commit changes
 - Commit without other users of VCS being able to see the changes
 - When you are happy with the version, push to central repository. Now available to other users of VCS
 - Advantages:
 - Having a local version
 - Can use multiple remote repositories

GIT

- One good representative of distributed version control systems, is GIT.
- Designed and developed by Linus Torvalds, creator of the Linux operating system.
- Linus was unhappy with the existing version control systems and wanted a different one for maintaining the Linux kernel, with some key characteristics.
 - distributed.
 - fast.
 - simple design
 - strong support for parallel branches
 - able to handle large projects.
- GIT workflow



Workspace
(Working Directory)



Index
(Stage)



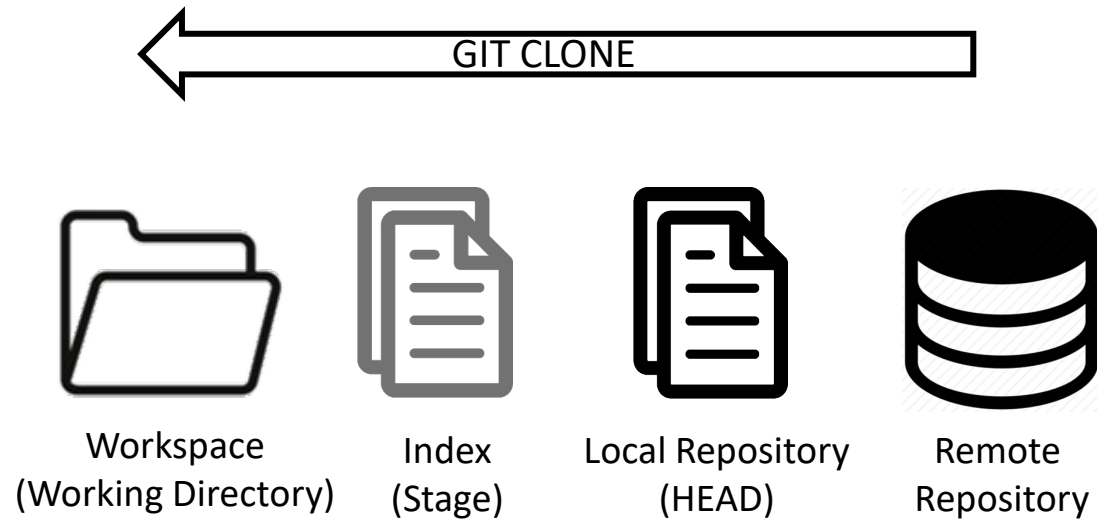
Local Repository
(HEAD)



Remote
Repository

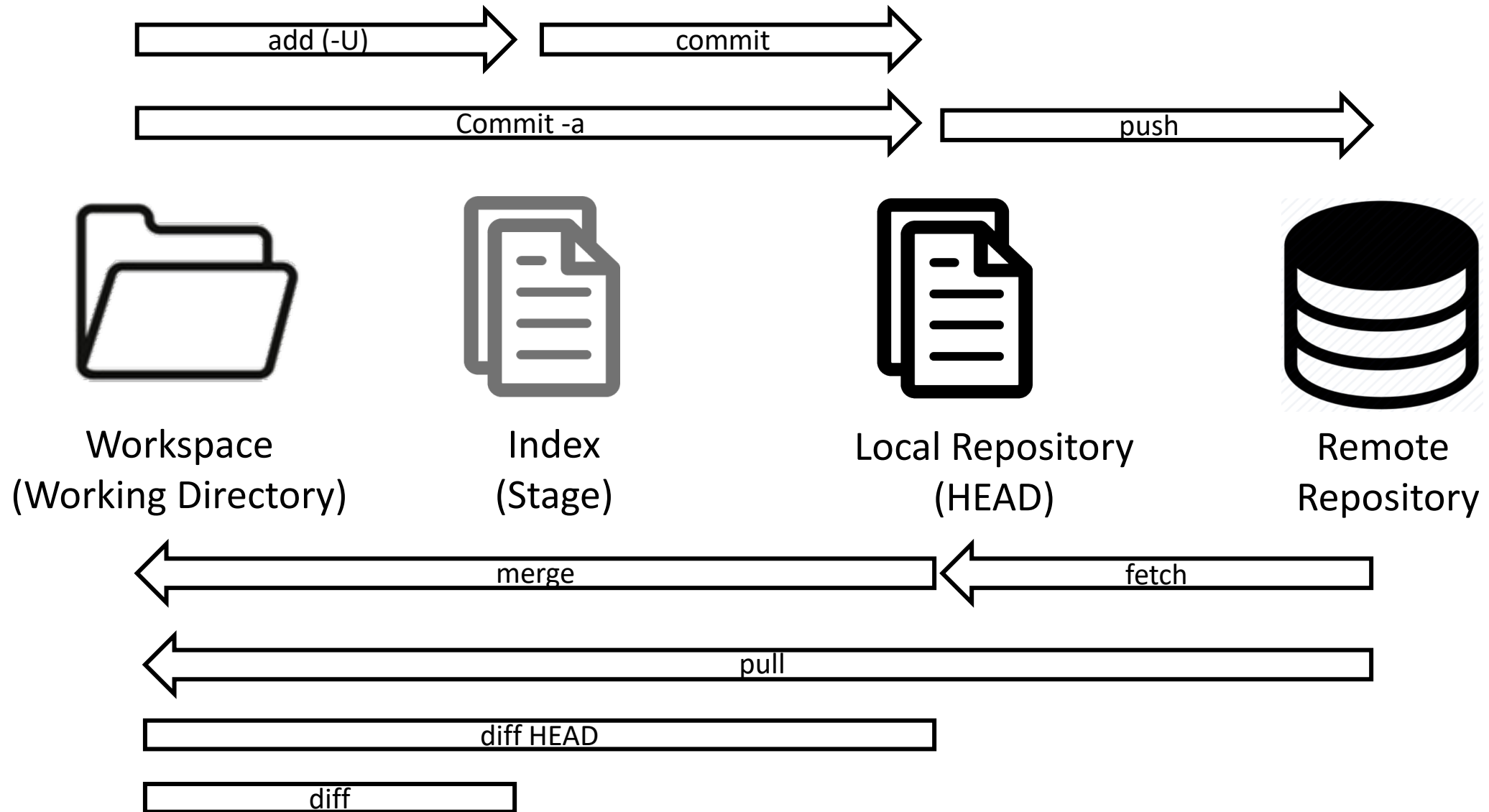
GIT Workflow

At any point, your files can be **modified** (locally) , **staged** or **committed**.



The git clone, followed by the url for that repository, will create a local copy of the repository in your workspace. You don't have to do this step if you're creating the repository yourself.

GIT Workflow



GIT Plugins

Install GIT: Follow instructions on <https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>

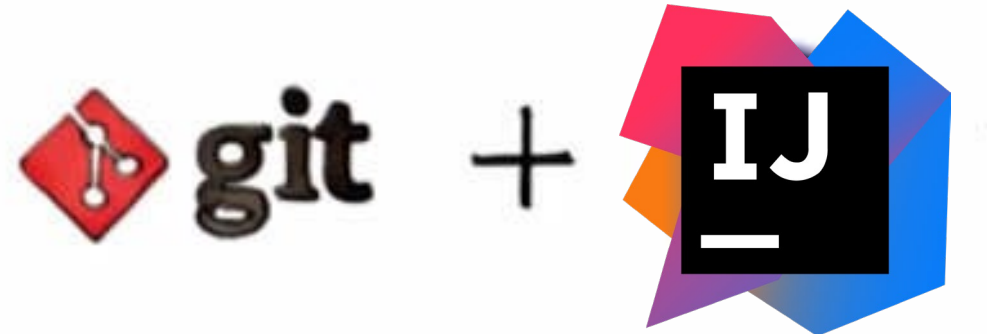
EGit

- GIT Plugin available for Eclipse
- The plugin can be downloaded at www.eclipse.github.com and can be installed in Eclipse



GitToolBox for IntelliJ

- The plugin can be downloaded [here](#).

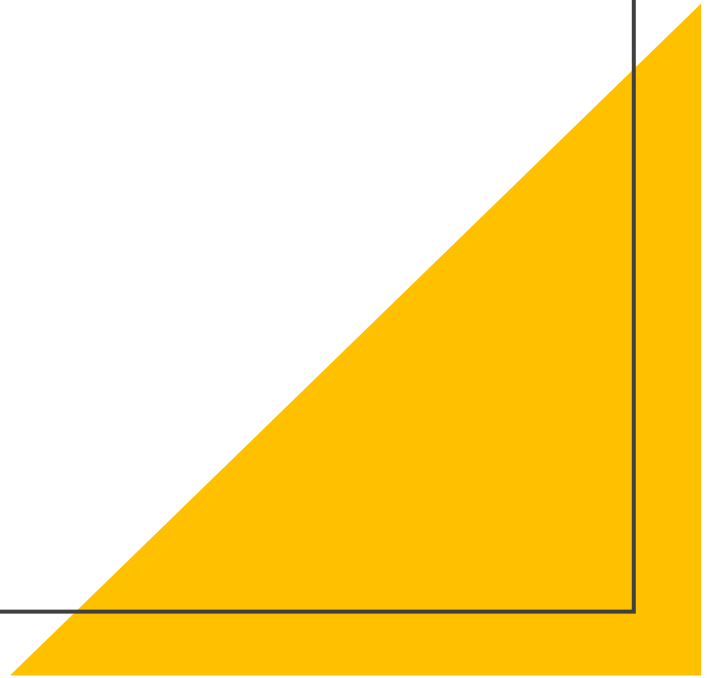


GitHub

- GIT hosting website. Get an account and create your remote repositories
- GitHub repository for your projects
- Provides easy-to-use FREE desktop clients for Mac and Windows
(<https://desktop.github.com>)
- **GitHub Pages:**
 - One click to enable for your GitHub repo.
 - Hosted directly from your GitHub repository.
 - Just edit, push, and your changes are live.
 - This course's website is a GitHub Page.
- ALWAYS SET YOUR GITHUB REPOSITORY TO BE PRIVATE, UNLESS YOU ARE ABSOLUTELY SURE YOU WANT IT PUBLIC !!!

Git Demo

Branches, Merge Conflicts, Code Review



GIT Demo – Creating Branches

- By default, when you create your project you will be on main/master
- It is good practice to have different branches for different features, people, etc.
- To see all local branches: `git branch`
- To create a new branch: `git branch [BRANCHNAME]`
- To move to (checkout) a branch: `git checkout [BRANCHNAME]`
- To create a new branch and move to it: `git checkout -b [BRANCHNAME]`

GIT Demo – Merging Branches

- Merging allows you to carry the changes in one branch over to another branch, combining both branches

To merge two branches:

1. `git checkout [NAME_OF_BRANCH_TO_MERGE_INTO]`
2. `git merge [NAME_OF_BRANCH_TO_BRING_IN]`

Example: merging *feature* branch into *master* branch:

1. `git checkout master`
2. `git merge feature`

GIT Recap

LOCAL REPOSITORIES

CREATE

mkdir ProjectName

cd ProjectName

git init

MODIFY

git add foo.txt

git commit -m "message"

INSPECT

git log

git status

git diff

git diff HEAD

git show

REMOTE REPOSITORIES

COPY REPOSITORY

git clone <repo url>

- Creates a completely local copy of repository
- links it to remote repository (origin)

RECEIVE CHANGES

git pull

SEND CHANGES

git push

Code Review



Why Code Reviews?


- Improve Overall Quality of Code
 - Having eyes on source code that you didn't write can help identify issues
- Facilitating Team Collaboration
 - Checking out each other's code better helps you understand how each feature is implemented
- Identifying bugs early in process
- Good for onboarding new developers to establish best practices within the organization
- 22% of your time in your job is code maintenance.


Pull Requests

- Tool to aggregate branch changes and request that the changes be merged into a different branch.
- Done through the GitHub GUI

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

 base: master < compare: dev < > ✓ Able to merge. These branches can be automatically merged.



Merges sprint 6 changes into master

Write

Preview


H B I ≡ <> 🔗 ≡ ≡ ☑ @ ↗ ↶

Sprint 6 changes include:

- ...

- ...

- ...

Attach files by dragging & dropping, selecting or pasting them. 

Create pull request >

Pull Request

Always have a branch protection rule enforced in your main GitHub repository branch

Settings → Branches → Branch Protection Rule → Require a pull request before merging

Note: This is only possible for public repositories (with GitHub free) and private repositories (with GitHub Pro). So, sign up for GitHub Pro. Look at Ed Discussion for details.

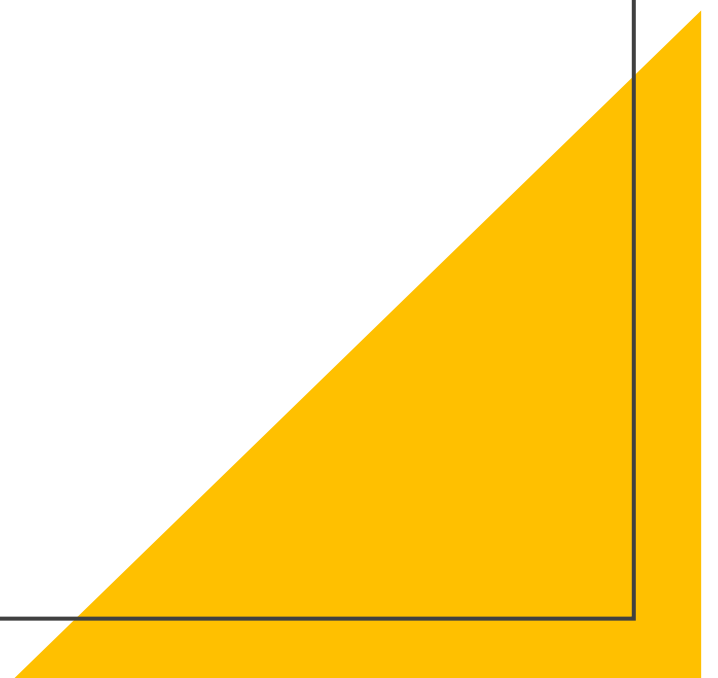
Code Review Assignment: Creating Branches, Pull Requests, Performing Reviews

In order to complete the accompanying assignment, you need do the following things:

1. [Create Separate Branches for every feature](#)
 - a. You might create sub branches of these branches as you implement new portions of each feature
2. [Perform a Pull Request](#)
3. [Reviewing Code and Closing Pull Requests](#)
4. Merging Branches [Do not delete them until the assignment is graded]

Let's do a quick demo of these items.

Git Quiz



Thursday (Work on
your projects in class)

