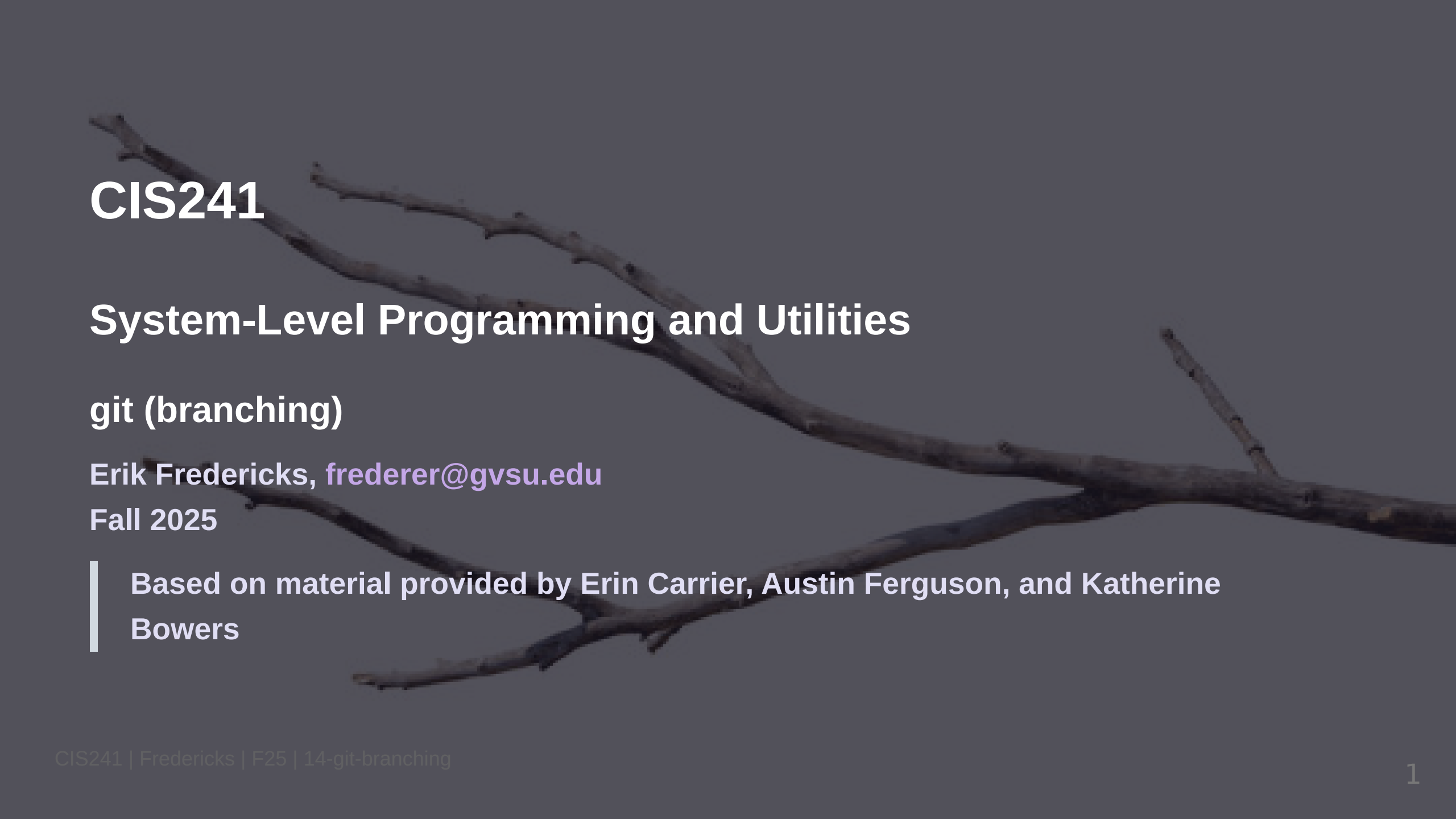




CIS241

System-Level Programming and Utilities

git (branching)

Erik Fredericks, frederer@gvsu.edu

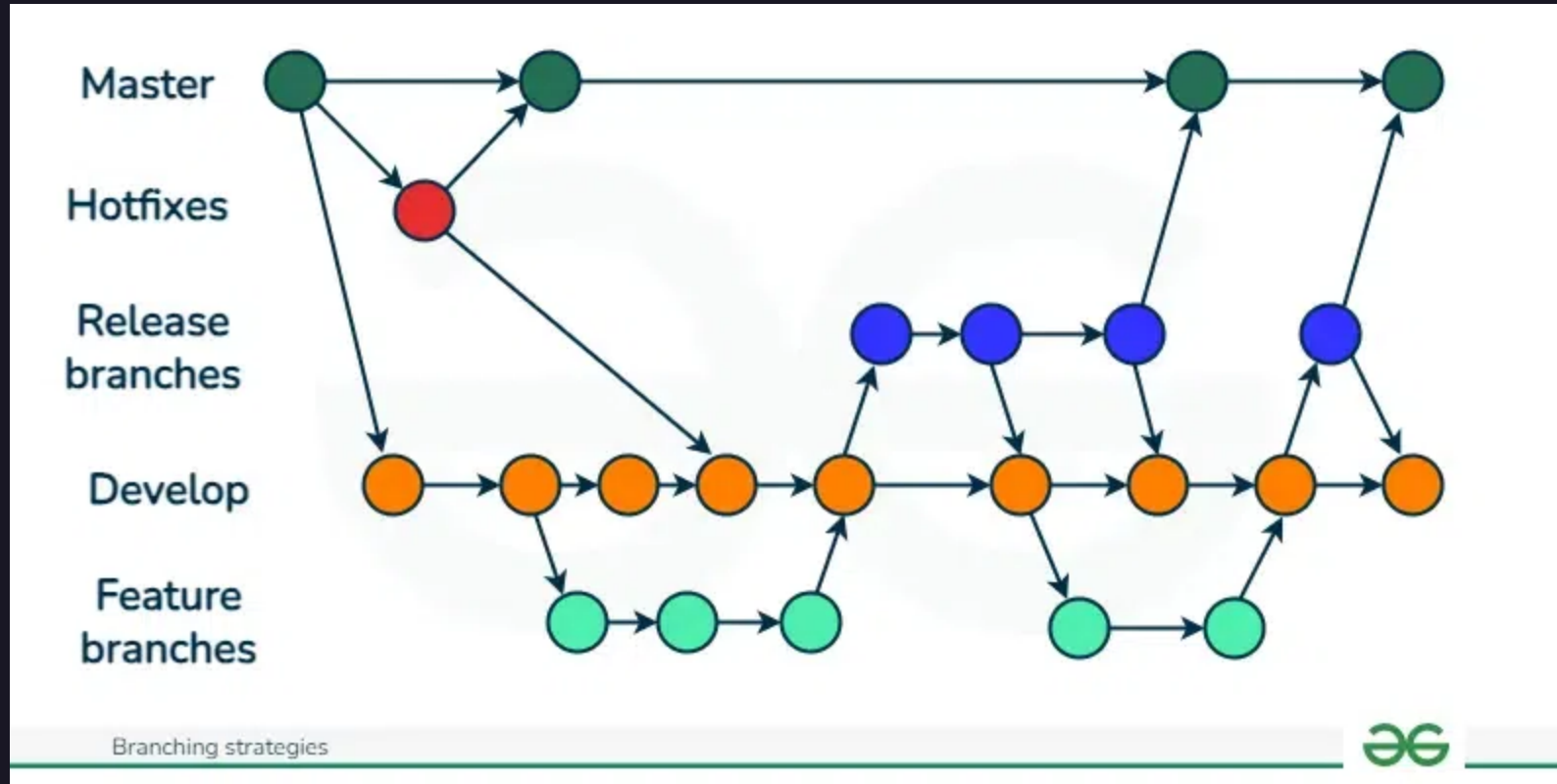
Fall 2025

Based on material provided by Erin Carrier, Austin Ferguson, and Katherine Bowers

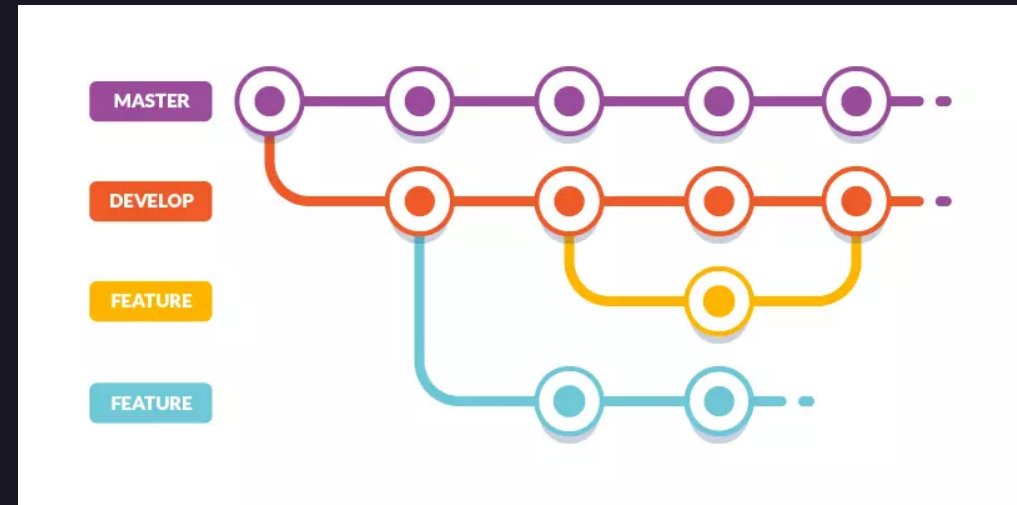
First, what is a branch?

- **Ways for dev teams to explore new features, work independently, etc.**
 - You aren't modifying the base code with your new *thing*
- **Branches have shared history**
 - Somewhere, they split off from the trunk!
- **Changing one branch (including `main`) does not immediately impact the other**

Example!



Default branch and workflow



Default branch:

- `main` - used to be `master`
 - You'll probably see both as you use other GitHub projects

Workflow:

- Typically keep `main` "clean" and "functioning"
- Create branches for bug fixes or feature development
- You can branch off a branch!

Commands

- List branches: `git branch -a`
- Create new branch: `git branch branchname`

More commands!

Switch to existing branch

- `git switch branchname` (newer way)
- `git checkout branchname` (older way)

Create and switch to new branch

- `git switch -c branchname` (newer way)
- `git checkout -b branchname` (older way)

Check where you're at?

- `git status`

OK WE'RE DONE WITH THE BRANCH NOW WHAT

Done and want to incorporate changes (i.e., into `main`)

1. Switch to branch you are merging **into**
 - `git switch` or `git checkout`, as before
2. `git merge branchname`
3. `git branch -d branchname` (delete branch)