Exercise 1: Branching

- Create a new folder and initialize a git repository inside it (`git init`)
- Create a file and commit it.
- Create and check out a new branch (`git branch <name>; git checkout <name>`)
- Create another file and commit it.
- Check out the master branch and merge your other branch into it (`git merge <name>`)
- View commit history (`git log --oneline`)

Exercise 2: Merge Conflicts

- Create and check out a new branch called `conflict`.
- Add 4 lines of text to one of your files and commit.
- Check out the `master` branch, edit the same file, add 4 different lines of text and commit.
- Try to merge the `conflict` branch.
- Work out the merge conflict and commit.
- View commit log (`git log --oneline --graph`)
- How many parent commits does each commit have?
- Delete conflict branch and view log again. What changed?

Exercise 3: Remotes

- Create a new repository with a README in GitHub.
- Clone the repository locally.
- In GitHub, edit the README file and save (commit).
- Use `git fetch` to get the new commit. Did your local copy of README change? What is the output of `git log` and `git log --remotes`?
- Merge in the new commit.
- On your computer, edit README again, commit, and push to GitHub.

Exercise 4: Simple Rebase

- In any local repo, create and checkout a new branch called `feature`
- Create a file and commit to the `feature` branch.
- Check out `master`, create another file, and commit.
- Switch back to `feature` and rebase it onto `master` (`git rebase master feature`)
- Check out `master`, and merge the `feature` branch back in. Did git use a merge commit or did it fast-forward?

Exercise 5: Removing Commits

- In any local repo, create a file called “sensitive.txt” and commit.
- Edit a different, non-sensitive file and commit again.
- Use `git rebase –i` to remove the commit that created the file sensitive.txt.
- Run `git log --oneline --graph --all`. What happened to the first commit? Was the other file you edited affected by the rebase?

Exercise 6: GitHub Issues

- In a GitHub repository you own, create an issue and assign it to yourself. Copy the issue number.
- Clone the repository locally (if you haven't already) and create a branch for your issue.
- Create a commit on that branch, and include "Closes #<issue number>" in the commit message. (e.g. "Closes #1")
- Check out `master` and merge in your issue branch, then push.
- Check the issue in GitHub. Did anything change?

Exercise 7: GitHub Forks

- Create a fork of `nuitrcs/github-playground` in your own account and clone the fork to your computer.
- Add the original repository as a remote called “upstream”:

  -
    ```
    git remote add upstream \
    https://github.com/nuitrcs/github-playground.git
    ```
- Wait for the instructor to push to the upstream repo, then pull in the change:
  - `git checkout master`
  - `git pull upstream master`

Exercise 8: Pull Requests

- Pair up with a partner.
- Partner A creates a public repository with a README file. Partner B forks and clones this repository.
- Partner B edits the README and commits, then pushes to their fork.
- Partner B creates a pull request.
- Partner A accepts and merges the pull request.
- Switch roles and do it again! Try commenting on the pull request this time.