30538 Problem Set 1: git

"This submission is my work alone and complies with the 30538 integrity policy." Add your initials to indicate your agreement: JK

Before you begin this problem set, please rename `ps1_template.qmd` as `ps1.qmd`.

Name: Jonathan Kacvinsky UChicago ID (`jkacvinsky@uchicago.edu`): 12430121

SOLO

Learn git branching (15 points)

Go to https://learngitbranching.js.org. This is the best visual git explainer we know of.

1. Complete all the levels of main "Introduction Sequence". Report the commands needed to complete "Git rebase" with one line per command.

```
git checkout -b bugFix
git commit
git checkout main
git commit
git checkout bugFix
git rebase main
```

2. Complete all the levels of main "Ramping up". Report the commands needed to complete "Reversing changes in git" with one line per command.

```
git checkout C4

git checkout bugFix^

git branch -f main C6
git checkout HEAD~1
git branch -f bugFix HEAD~1

git reset HEAD~1
git checkout pushed
git revert pushed
```

3. Complete all the levels of remote "Push & Pull – Git Remotes!". Report the commands needed to complete "Locked Main" with one line per command.

```
git clone

git checkout main
git commit
git checkout o/main
git commit

git fetch

git pull

git clone
git fakeTeamwork 2
git commit
git pull

git commit
git commit
git push

git clone
git fakeTeamwork 1
git commit
git pull --rebase
git push

git branch -f main o/main
```

```
git checkout -b feature C2
git push origin feature
```

Exercises

- Basic Staging and Branching (10-15)

1. Exercise. For your pset submission, tell us only the answer to the last question (22).

```
On branch master
nothing to commit, working tree clean
```

2. Exercise. For your pset submission, tell us only the output to the last question (18).

```
diff --git a/file1.txt b/file1.txt
deleted file mode 100644
index 785d005..0000000
Binary files a/file1.txt and /dev/null differ
```

- Merging

1. Exercise. After completing all the steps (1 through 12), run git log –oneline –graph –all and report the output.

```
* 6739855 (HEAD -> master) greeting uppercased
* 688b92c Add content to greeting.txt
* f5f9cbb Add file greeting.txt
```

2. Exercise. Report the answer to step 11.

```
*   5f4dd73 (HEAD -> master) Altered Message
|\
| * 492b14c (greeting) New Greeting
* | 13e0241 README File
|/
* 994687b Add content to greeting.txt
* 14a476c Add file greeting.txt
```

3. Identify the type of merge used in Q1 and Q2 of this exercise. In words, explain the difference between the two merge types, and describe scenarios where each type would be most appropriate.

In questions 1 and 2 we merged branches, rather than rebasing. We merge commits in scenarios like this where we want to keep all of our information and the parents of each branch. We rebase when we want to just have a single, sequential line of commmits and history to make it more straightforward.

- Undo, Clean, and Ignore

1. Exercise. Report the answer to step 13.

```
commit 5956c8b27accde186dc02310836e64b4b2069275 (HEAD -> master)
Author: git-katas trainer bot <git-katas@example.com>
Date:   Fri Jan 9 17:55:58 2026 -0600

    Revert "Add credentials to repository"

    This reverts commit 09e77ca26f57caea8d6b49a731fac38a61905e2c.

diff --git a/credentials.txt b/credentials.txt
deleted file mode 100644
index 8995708..0000000
--- a/credentials.txt
+++ /dev/null
@@ -1 +0,0 @@
-supersecretpassword
```

2. Exercise. Look up `git clean` since we haven't seen this before. For context, this example is about cleaning up compiled C code, but the same set of issues apply to random files generated by knitting a document or by compiling in Python. Report the terminal output from step 7.

```
unknown option: -f
usage: git [-v | --version] [-h | --help] [-C <path>] [-c <name>=<value>]
           [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
           [-p | --paginate | -P | --no-pager] [--no-replace-objects]
           [--no-lazy-fetch]
           [--no-optional-locks] [--no-advice] [--bare] [--git-dir=<path>]
           [--work-tree=<path>] [--namespace=<name>]
           [--config-env=<name>=<envvar>]
           <command> [<args>]
```

3. Exercise. Report the answer to 15 ("What does git status say?")

```
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
```

```
        deleted:    file1.txt
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   .gitignore
Tracked files:
  (use "git add <file>..." to include in what will be committed)
        file3.txt


A use for keeping track of files that the extension ignored could be that it
has valuable information that you don't want published but still want to
refer to.'
```