

# 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: **A S**

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

Name: **Abraham Sadat**

UChicago ID (xxx@uchicago.edu): **asadat@uchicago.edu**

## 1 SOLO

### 1.1 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 branch bugFix
git checkout 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 reset HEAD~1
git checkout pushed
git revert HEAD
```

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 branch -f main o/main
$ git checkout -b feature c2
$ git push
```

## 1.2 Exercises

- Basic Staging and Branching (10-15)

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

Running `git status` tells us that our branch is still 1 ahead of head/master by one commit, and to use `git push` to publish my local commits. It also tells me there's nothing to commit and that the working tree is 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 3cf2b22..0000000
--- a/file1.txt
+++ /dev/null
@@ -1 +0,0 @@
-Abraham Sadat
diff --git a/file2.txt b/file2.txt
new file mode 100644
index 0000000..e69de29
```

- Merging

1. [Exercise](#). After completing all the steps (1 through 12), run `git log --oneline --graph --all` and report the output.

```
* a6a553f (HEAD -> master) uppercase greeting
* 0d3ce26 Add content to greeting.txt
* 86e42b8 Add file greeting.txt
```

2. [Exercise](#). Report the answer to step 11.

```
* 32ac3fe (HEAD -> master, greeting) tsinga ye
* 30563d8 Add content to greeting.txt
* d8efb7e 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.

The merge used in Q2 was a three-way-merge, whereas question 1 uses a fast forward merge

- Undo, Clean, and Ignore

1. [Exercise](#). Report the answer to step 13.

```
commit 9330f0b3a8c683ebeb61cd2ee390a5bc0b2b1b27 (HEAD -> master)
Author: git-katas trainer bot <git-katas@example.com>
Date: Sat Jan 10 16:18:56 2026 -0600
```

Overwrite greeting.txt

```
diff --git a/greeting.txt b/greeting.txt
index 2bdd920..8a8069f 100644
--- a/greeting.txt
+++ b/greeting.txt
@@ -1,1 @@
-Original file content
+This should have been appended to the original content, rather than overwriting
it.
(base) abrahamsadat@Abrahams-Air exercise % :q!
```

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.

```
(base) abrahamsadat@Abrahams-Air exercise % ls -f -d
```

.

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
```

This method can help with tracking a specific type of file and ignoring irrelevant ones