

COMP 4081  
Exam 1 – Git Graph Retry  
Spring 2019

Name: Solutions, \_\_\_\_\_  
Last name First name

**Rules:**

- No potty breaks.
- Turn off cell phones/devices.
- Closed book, closed note, closed neighbor.
- WEIRD! Do not write on the backs of pages. If you need more pages, ask me for some.

**Reminders:**

- Verify that you have all pages.
- Don't forget to write your name.
- Read each question carefully.
- Don't forget to answer every question.

Each of the following problems presents a Git log graph (log messages omitted) of a local repo and a scenario. Update the graph by crossing out and/or adding appropriate text.

- You may or may not need to use the two blank lines above the graph.
- If you need to add a commit, use the hash c1c1c1c.
- If a command would be rejected by GitHub (e.g., because the remote contains work that you do not have locally), write “REJECTED” on the top line.
- Assume that all remote bookmarks depicted are up to date.

I have included an example problem and solution below to help clarify what’s expected.

### Example Problem

Scenario: Developer makes changes to the code, stages the changes, and commits.

```
* 86b8116 (HEAD --> master, iss1)
* dc003f8
* 026c6cf
```

### Example Solution

```
* c1c1c1c (HEAD → master)
* 86b8116 (HEAD --> master, iss1)
* dc003f8
* 026c6cf
```

1. [3%] Scenario: Developer runs `git checkout -b iss12`.

```
* 04ca8c6 (master)
* | 3283dd7 (HEAD → iss11, HEAD → iss12)
|/
* a8da338
* fe2251a
```

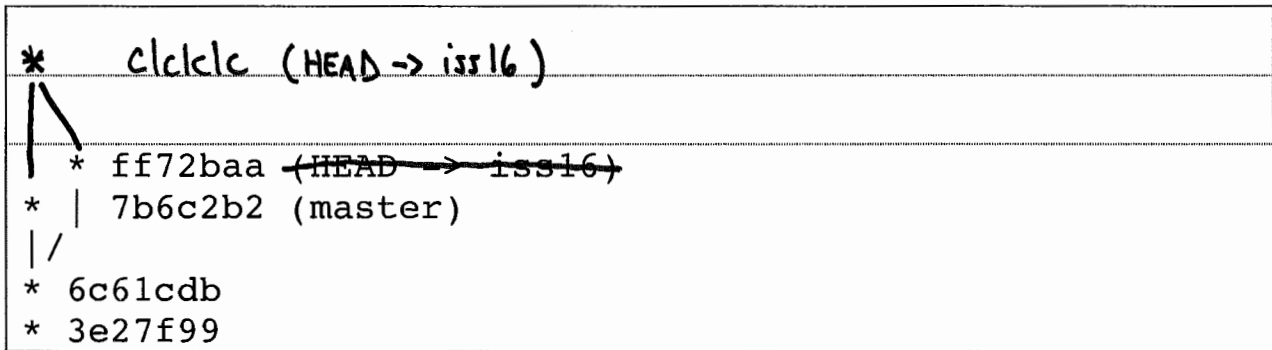
2. [3%] Scenario: Developer makes changes to the code, stages the changes, and commits.

```
* e1c1c1c (HEAD → iss13)
* d3994b3 (HEAD → iss13)
* | 0152ac4 (origin/master, master)
|/
* 77a9025 (origin/iss13)
```

3. [3%] Scenario: Developer runs `git checkout master`.

```
* d197593 (HEAD → iss14)
* | 0f50aa4 (iss15)
|/
* 75a7005 (master) (HEAD → master)
* cbff2e7
```

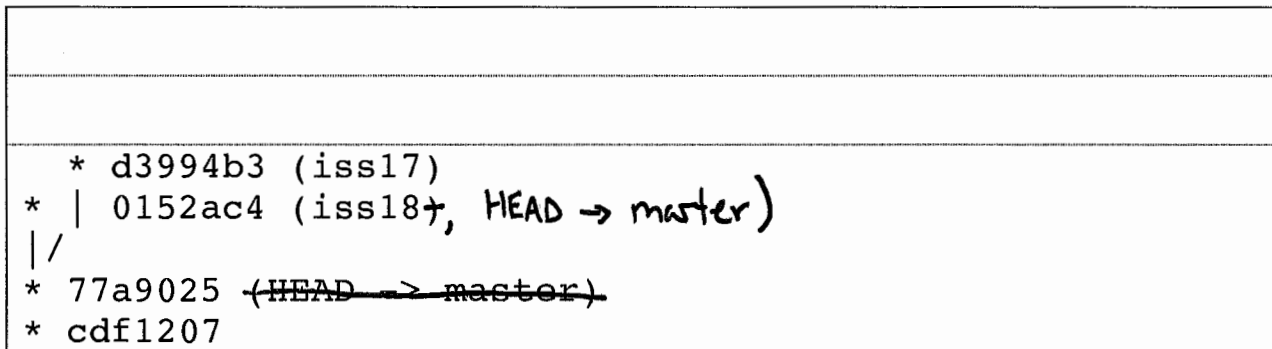
4. [3%] Scenario: Developer runs `git merge master`. Assume that auto-merge, if used, would complete successfully with no merge conflicts.



5. [1%] Would this be a fast-forward merge?

- a. Yes
- b.  No

6. [3%] Scenario: Developer runs `git merge iss18`. Assume that auto-merge, if used, would complete successfully with no merge conflicts.



7. [1%] Would this be a fast-forward merge?

- a.  Yes
- b. No

8. [3%] Scenario: Developer runs `git pull origin master`. Assume that auto-merge, if used, would complete successfully with no merge conflicts.

```
* c1c1c1c (HEAD -> iss19)
| \
| * d3994b3 (HEAD -> iss19, origin/iss19)
* | 0152ac4 (origin/master, master)
| /
* 77a9025
```

9. [3%] Scenario: Developer runs `git push`. Assume that auto-merge, if used, would complete successfully with no merge conflicts. Assume that all issue branches are tracking with their corresponding branches on the remote.

```
REJECTED

* 79bb885 (HEAD -> iss20)
* | 33a99ac (origin/iss20, master)
| /
* d23531d (origin/master)
```

*← correction*

10. [3%] Scenario: Developer runs `git push origin master`. Assume that auto-merge, if used, would complete successfully with no merge conflicts. Assume that all issue branches are tracking with their corresponding branches on the remote.

```
* e28a3c2 (HEAD -> iss21, origin/iss21)
* f40dd3b (origin/iss21)
* | 1061bb5 (master)
| /
* 86b8116 (origin/master)
```