#### Practice Quiz: Pull Requests

**TOTAL POINTS 5**

1.

Question 1

What is the difference between using squash and fixup when rebasing?

**1 / 1 point**



Squash deletes previous commits.



Squash combines the commit messages into one. Fixup discards the new commit message.



Squash only works on Apple operating systems.



Fixup combines the commit messages into one. Squash discards the commit message.

**Correct**

Awesome! The fixup operation will keep the original message and discard the message from the fixup commit, while squash combines them.

2.

Question 2

What is a pull request?

**1 / 1 point**



The owner of the target repository requesting you to add your changes.



A request sent to the owner and collaborators of the target repository to pull your recent changes.



A request to delete previous changes.



A request for a specific feature in the next version.

**Correct**

Right on! You send a pull request to the owner of the repository in order for them to incorporate it into their tree.

3.

Question 3

Under what circumstances is a new fork created?

**1 / 1 point**



When you want to experiment with changes without affecting the main repository.



When you clone a remote repository to your local machine.



During a merge conflict.



When there are too many branches.

**Correct**

Nice work! For instance, when you want to propose changes to someone else's project, or base your own project off of theirs.

4.

Question 4

What combination of command and flags will force Git to push the current snapshot to the repo as it is, possibly resulting in permanent data loss?

**1 / 1 point**



git push -f



git log --graph --oneline --all



git status



git rebase -i

**Correct**

Awesome! git push with the -f flag forcibly replaces the old commits with the new one and forces Git to push the current snapshot to the repo as it is. This can be dangerous as it can lead to remote changes being permanently lost and is not recommended unless you're pushing fixes to your own fork (nobody else is using it) such as in the case after doing interactive rebasing to squash multiple commits into one as demonstrated.

5.

Question 5

When using interactive rebase, which option is the default, and takes the commits and rebases them against the branch we selected?

**1 / 1 point**



squash



edit



reword



pick

**Correct**

Great job! The pick keyword takes the commits and rebases them against the branch we have chosen.

Git Fork and Pull Request Cheat Sheet

Check out the following link for more information:

* <https://help.github.com/en/articles/about-pull-request-merges>

More Information on Code Reviews

Check out the following links for more information:

* <http://google.github.io/styleguide/>
* <https://help.github.com/en/articles/about-pull-request-reviews>
* <https://medium.com/osedea/the-perfect-code-review-process-845e6ba5c31>
* <https://smartbear.com/learn/code-review/what-is-code-review/>

#### Practice Quiz: Code Reviews

**TOTAL POINTS 5**

1.

Question 1

When should we respond to comments from collaborators and reviewers?

**1 / 1 point**



When their comments address software-breaking bugs



No need, just resolve the concerns and be done with it



Always



Only when a code correction is necessary

**Correct**

Excellent! It is good manners and proper conduct to respond, even when it's simply an acknowledgement.

2.

Question 2

What is a nit?

**1 / 1 point**



A trivial comment or suggestion



A couple lines of code



A repository that is no longer maintained



An orphaned branch

**Correct**

Good work! In git jargon (and elsewhere in the tech world), a nit is a minor “nitpick” about a piece of code.

3.

Question 3

Select common code issues that might be addressed in a code review. (Check all that apply)

**1 / 1 point**



Using unclear names

**Correct**

Excellent! Unclear names can make our code hard to understand.



Following PEP8 guidelines



Forgetting to handle a specific condition

**Correct**

Alright! If there is a specific condition that could cause a problem and we don't address it, the result could be catastrophic.



Forgetting to add tests

**Correct**

Woohoo! Tests are an important addition to our code to ensure it runs smoothly.

4.

Question 4

If we've pushed a new version since we've made a recent change, what might our comment be flagged as?

**1 / 1 point**



Accepted



Resolved



Outdated



Merged

**Correct**

Nice job! If we push a new version after making a change, old comments are marked with the "Outdated" flag.

5.

Question 5

What are the goals of code review? (Check all that apply)

**1 / 1 point**



Make sure that the contents are easy to understand

**Correct**

Right on! By reviewing our code, we can identify where we can make our code more clear and easy to understand.



Ensure consistent style

**Correct**

Awesome! By comparing our code to style guidelines, we can keep our style consistent and readable.



Build perfect code



Ensure we don't forget any important cases

**Correct**

Good job. Code review can reveal cases or conditions we need to handle in our code.

Additional Tools

Check out the following links for more information:

* <https://arp242.net/diy.html>
* <https://help.github.com/en/articles/closing-issues-using-keywords>
* <https://help.github.com/en/articles/setting-guidelines-for-repository-contributors>
* <https://www.infoworld.com/article/3271126/what-is-cicd-continuous-integration-and-continuous-delivery-explained.html>
* <https://stackify.com/what-is-cicd-whats-important-and-how-to-get-it-right/>
* <https://docs.travis-ci.com/user/tutorial/>
* <https://docs.travis-ci.com/user/build-stages/>

#### Practice Quiz: Managing Collaboration

**TOTAL POINTS 5**

1.

Question 1

How do we reference issues in our commits with automatic links?

**1 / 1 point**



By using one of the keywords followed by a hashtag and the issue number.



By using an asterisk (\*) after the issue number.



By typing the issue number inside braces ({}).



By using a special keyword.

**Correct**

Right on! Keywords such as ***closes*** or ***resolves*** followed by a hashtag and the issue number will tell Git to autolink to the issue with the provided ID number.

2.

Question 2

What is an artifact in terms of continuous integration/continuous delivery (CI/CD) pipelines?

**1 / 1 point**



An old and obsolete piece of code or library.



Any file generated as part of the CI/CD pipeline.



An unintended minor glitch in a computer program



An automated series of tests that run each time there is a new commit or pull request.

**Correct**

Awesome! Artifacts can include compiled code, test results, logs, or any type of file generated as part of the pipeline.

3.

Question 3

Which of the following statements are good advice for project maintainers? (Check all that apply)

**1 / 1 point**



Coordinate solely via email



Reply promptly to pull-requests

**Correct**

Woohoo!  The more time that passes until a pull-request gets reviewed, the more likely it is that there's a new commit that causes a conflict when you try to merge in the change.



Understand any changes you accept

**Correct**

Nice job! Not only do we not know whether the original coder is going to be around to maintain those functions, we also want to avoid feature creep and unmanageable code.



Use an issue tracker

**Correct**

Excellent! The larger our project grows, the more useful an issue tracker can be for collaborating.

4.

Question 4

Which statement best represents what a Continuous Integration system will do?

**1 / 1 point**



Run tests automatically



Update with incremental rollouts



Assign issues and track who's doing what



Specify the steps that need to run to get the result you want

**Correct**

Nice job! A continuous integration system will build and test our code every time there's a change.

5.

Question 5

Which statement best represents what a Continuous Delivery (CD) system will do?

**1 / 1 point**



Run tests automatically



Update with incremental rollouts



Assign issues and track who's doing what



Specify the steps that need to run to get the result you want

**Correct**

Right on! Continuous Delivery means new code is often deployed with the goal of avoiding rollouts with lots of changes between two versions.