✓ **Congratulations! You passed!**
TO PASS 80% or higher

Keep Learning

GRADE
100%

Practice Quiz: Branching & Merging

TOTAL POINTS 5

---

1. When we merge two branches, one of two algorithms is used. If the branches have *diverged*, which algorithm is used? `1 / 1 point`

   ⦿ three-way merge

   ○ fast-forward merge

   ○ merge conflict

   ○ orphan-creating merge

   > ✓ **Correct**
   >
   > Excellent! A three-way-mergeoccurs when the two commits have diverged previously, and a new commit is created.

2. The following code snippet represents the result of a merge conflict. Edit the code to fix the conflict and keep the version represented by the current branch. `1 point`

   ```
   1   <<<<<<< HEAD
   2   print("Keep me!")
   3   =======
   4   print("No, keep me instead!")
   5   >>>>>>> brancho-cucamonga
   ```

   Run

   Reset

2. The following code snippet represents the result of a merge conflict. Edit the code to fix the conflict and keep the version represented by the current branch. `1 / 1 point`

   ```
   1   print("Keep me!")
   2
   ```

   Run

   Reset

   ```
   Keep me!
   ```

   > ✓ **Correct**
   >
   > You got it! No more conflicts here!

3. What command would we use to throw away a merge, and start over?   **1 / 1 point**

- ○ git checkout -b <branch>
- ◉ git merge --abort
- ○ git log --graph --oneline
- ○ git branch -D <name>

✓ **Correct**

Right on! If there are merge conflicts, the --abort flag can be used to abort the merge action.

4. How do we display a summarized view of the commit history for a repo, showing one line per commit?   **1 / 1 point**

- ○ git log --format=short
- ○ git branch -D <name>
- ◉ git log --graph --oneline
- ○ git checkout -b <branch>

✓ **Correct**

Awesome! The commandgit log --graph --oneline shows a summarized view of the commit history for a repo.

5. The following script contains the result of a merge conflict. Edit the code to fix the conflict, so that both versions are included.   **1 point**

```
1    def main():
2    <<<<<<< HEAD
3        print("Start of program>>>>>>>")
4    =======
5        print("End of program!")
6    >>>>>>> improvement-to-the-code
7
8    main()
```
Run

Reset

5. The following script contains the result of a merge conflict. Edit the code to fix the conflict, so that both versions are included.   **1 / 1 point**

```
1    def main():
2        print("Start of program>>>>>>>")
3        print("End of program!")
4
5    main()
```
Run

Reset

✓ **Correct**

Great work! Now the code has both versions without any conflicts!