



Merges

TOTAL POINTS 5

1. Which procedure does a better job of preserving a project's history?

1 point

- ☒ **git merge**
- ☐ **git rebase**

2. Why might a merge result in problems that do not show conflicts (select all correct answers)?

1 point

- ☐ The merge may alienate a developer who was opposed to it
- ☒ There may be two different solutions to the same problem which interfere with other
- ☐ It may increase the cost of the product
- ☒ A merge might affect code in a very different part of the product in a non-obvious way and not receive enough testing

3. A **git rebase**

1 point

- ☐ hides changes so noone can trace where they are coming from
- ☒ adapts a branch to incorporate the latest changes in another branch without yet merging this branch into the other branch
- ☐ is not a legal operation in a public repository

4. How would you merge two branches (**br1** and **br2**) into the master branch?

1 point

- ☐ **git checkout br1 && git merge br2 && git checkout master && git merge br1**
- ☐ **git checkout master && git merge br1 br2**
- ☒ **git checkout master && git merge br1 && git merge br2**

5. What do you do when a merge fails?

1 point

- ☒ Evaluate the conflict, see what the correct result should be and then fix
- ☐ Abandon all changes desired and start over with a **git revert**
- ☐ Find out who is to blame and force them to fix their errors

☒ I, **Piyush Sambhi**, understand that submitting work that isn't my own may result in permanent failure of this course or deactivation of my Coursera account.

[Learn more about Coursera's Honor Code](#)



Save

Submit