Module-3 MCQs

1. Which of the following command is used to create a new branch and switch to it?

a. git checkout <branch_name>
b. git checkout -b <branch_name>
c. git change <branch_name>
d. git traverse <branch_name>

Ans: b

Explanation:

"git checkout -b <branch_name>" command creates a branch and then switch to it.

---

2. When would Git employ the “fast-forward” merge algorithm?

a. At all times
b. Only when merge master commits to branches
c. Only when a branch has branched off the latest commit, you want to merge to
d. Only when the branch has branched off from an older commit on the master branch

Ans: c

Explanation:

Fast-forward is employed when you merge into a branch, the latest commit of which is your parent.

---

3. How can you view all the branches that exist in the repository that you cloned?

a. You can only view the master branch of the cloned repository
b. You can only view the active branch of the cloned repository
c. You can view them using the command 'git branch –a'
d. You can view them using the command 'git branch –show'

Ans: c

Explanation:

The 'git branch –a' command will show all the branches.

---

4. How rebase is different from the merge functionality?

a. In the merge, all commits from the point of divergence are brought into a new commit, whereas in rebase, the branch commits are moved sequentially after the latest master commit
b. Merge and rebase are the same
c. Merge and rebase are the same except that for rebase to work, the latest commit in the master branch must be the parent of the feature branch

d. In the merge, all commits from the point of divergence are merged into the latest master commit, whereas in rebase, the branch commits are moved sequentially after the latest master commit

Ans: a

Explanation:

In the merge, all commits from the divergent point are brought into a new commit, whereas in rebase, the branch commits are moved sequentially after the latest master commit. Git rebase moves a feature branch into a master. Whereas git merge adds a new commit, preserving the history.

---

5. How to delete a local branch in git forcefully?

a. git branch -d <local-branch>
b. git branch -D <local-branch>
c. git branch -f -d <local-branch>
d. git branch -fd <local-branch>

Ans: b

Explanation:

'git branch -D <local-branch>' is used to delete local branches forcefully. This will force deletion of the branch, even if it contains unmerged/unpushed commits.

---

6. How to delete a remote branch in git?

a. git push origin --delete <remote-branch-name>
b. git push origin --remove <remote-branch-name>
c. git push origin --flush <remote-branch-name>
d. git push origin --erase <remote-branch-name>

Ans: a

Explanation:

To delete a *remote* branch, you need to use the "git push origin --delete" command.

---

7. What is the main reason for the merge conflicts?

a. Changes done in different files in different branches
b. Changes done in same file in different branches
c. Changes done in same file in the same branch
d. Changes done in different files in the same branch

Ans: b

Explanation:

Generally, merge conflicts occur due to changes made in the same file in different branches.

---

8. Which command is used to compare two specified branches?

a. git diff
b. git merge
c. git blame
d. git check

Ans: a

Explanation:

git diff command compares specified branches.

---

9. Which command is used to rename the current branch?

a. git remote rm
b. git branch -m
c. git branch -D
d. git rebase

Ans: b

Explanation:

git branch -m command is used to rename the current branch.

---

10. Which command is used to list the git tags?

a. git tag
b. git tag --list
c. git list tags
d. git tags -l

Ans: a, b, d

Explanation:

git tag, git tag --list and git tags -l commands are used to list all the git tags.

. Git tag is used to capture a point in history used for a marked version release like v1. 0.1
