Module-4 MCQs

1. **What are the various kinds of Workflows in Git?**
    a. Centralized Workflow
    b. Feature Branch Workflow
    c. Gitflow Workflow
    d. Forking Workflow
    e. All the above

Ans: e

Explanation:

Below are the various forms of Workflows in Git:

- Centralized Workflow
- Feature Branch Workflow
- Gitflow Workflow
- Forking Workflow

---

2. **What are the key benefits of Git workflow?**
    a. Parallel development
    b. Collaboration
    c. Release staging area
    d. Support for emergency fixes
    e. All the above

Ans: e

Explanation:

Some of the key benefits of working with Git workflow are:

- Parallel Development
- Collaboration
- Release Staging Area
- Support for Emergency fixes

---

3. **What are the key considerations while assessing a Git workflow?**
    a. Is it easy to revert mistakes and errors with this workflow?

b. Does this workflow move up with team size?
c. Does this workflow force any new unnecessary overhead on the team?
d. All the above

Ans: d

Explanation:

All these are important things to consider when assessing a Git workflow.
.

---

4. How to initialize git flow?

a. git flow init
b. git flow initialize
c. git init flow
d. git-flow init

Ans: a

Explanation:

'git flow init' command is employed to initialize a git workflow.

---

5. How to create a feature branch using git flow?

a. git branch develop
b. git flow feature start feature_branch
c. git checkout -b feature_branch
d. git branch

Ans: b

Explanation:

'git flow feature start feature_branch' will be used to create a new feature branch using the git workflow.

---

6. **Deleting old local branches may be a good practice?**
    a. True
    b. False

Ans: a

Explanation:

It is always better to delete the old stale branches if you do not need them anymore. It is considered one of the Git best practices.

---

7. **How to wipe all your staged and uncommitted changes permanently?**
    a. git reset -all
    b. git reset -delete
    c. git reset -soft
    d. git reset -hard

Ans: d

Explanation:

The git reset -hard command removes all your staged and uncommitted changes.

---

8. **How to push a feature branch to a remote repository?**
    a. git clone -u origin demo-feature
    b. git push -u demo-feature origin
    c. git push -u origin demo-feature
    d. git commit -u origin demo-feature

Ans: c

Explanation:

'git push -u origin demo-feature' command pushes demo-feature to the central repository (origin), and the -u flag adds it as a remote-tracking branch.

---

9. **How to create a new branch?**
    a. git checkout -b demo-feature
    b. git clone -b demo-feature
    c. git fork -b demo-feature
    d. git commit -b demo-feature
    e.

Ans: a

Explanation:

'git checkout -b demo-feature' command checks out a branch called demo-feature based on master, and the -b flag tells Git to make the branch if it does not exist already.

---

10. **What are the steps to commit a file to a repo?**
    a. git commit -m “message”
       git add -A
    b. git push <some-file>
       git status
       git commit
    c. git status
       git add -A
       git commit
    d. git add -A
       git status
       git commit -m “commit message”

Ans: d

Explanation:
a. Not correct because without adding the changes to the staging dir, we cannot commit directly
b. Not correct because we cannot push the changes without saving them locally, so first, we need to save the changes and then need to push them to the repository
c. Not correct because a commit message is required to save the changes.
d. Option d is correct because first, you stage the file in the staging area, then check the status and in the last commit with the message.