

Time: 15 mins
Maximum Marks: 10

1. Which Git command is used to unstage changes from the staging area for a specific file?
a) git reset
b) git stash
c) git checkout -- file-name
d) git clean -f
2. For things like major releases and big merges, is there any way to mark these commits with something more permanent than a branch?

-
- The diagram illustrates the difference between a branch and the HEAD pointer in Git. It is divided into two parts: 'Branch' (top) and 'HEAD' (bottom).
- Branch:** Shows a sequence of commits: 'first commit' (e137e9b..), 'C2' (7920cd1..), 'C3' (666697a..), and 'C4' (9a56bf2..). The 'master' branch is highlighted in orange, and the 'HEAD' pointer is shown in a green box pointing to commit C3. A dashed circle represents a commit that is not part of the current branch's history.
- HEAD:** Shows a sequence of commits: 'first commit' (e137e9b..), 'C1' (a75b3df..), 'C2' (0b92626..), 'C3' (7ba84ee..), and 'C4' (a21ca88..). The 'HEAD' pointer is shown in a green box pointing to commit C3, and the 'master' branch is highlighted in orange.

4. What does "md" stand for in readme.md?_____

5. What is equivalent to fetch and merge?

a) push

b) pull

c) fetch

d) synchronize

6. Recall the experiment you did in class. Now carefully see the following git-flow. [4 Marks]

```
git init

echo ``Alice`` >> alice.md
git add alice.md; git commit -m ``C1-Alice-1-file``

echo ``Bob`` >> bob.md
git add bob.md; git commit -m ``C2-Bob-2-files``

echo ``Eve`` >> eve.md
echo ``Alice Updated`` >> alice.md
git add eve.md alice.md
git commit -m ``C3-Eve-Alice-3-files``

rm bob.md eve.md
git add .
git commit -m "C4-all deleted-0-files"

echo ``Bob`` >> bob.md
git add bob.md; git commit -m ``C5-Bob-2-files``
```

git cherry-pick <C4-Commit-Hash>

What are the files in the directory after the above cherry-pick? What is the status of the above cherry-pick? Justify your answer.

Time: 15 mins
Maximum Marks: 10

-
- The diagram illustrates a Git revert operation. It is divided into two parts: the initial state (top) and the state after the revert (bottom).
- Initial State (Top):**
- A sequence of commits is shown as circles connected by arrows pointing left, indicating the commit history.
 - first commit** (hash: e137e9b...)
 - C2** (hash: 7920cd1...)
 - C3** (hash: 666697a...): This is the current **HEAD** and the **master** branch.
 - C4** (hash: 9a56bf2...): A dashed circle, indicating it is not part of the current branch's history.
- State After Revert (Bottom):**
- The sequence of commits is updated:
 - first commit** (hash: e137e9b...)
 - C1** (hash: cd6a584...)
 - C2** (hash: 0e64049...)
 - C3** (hash: f9f98e4...)
 - C4** (hash: 7c987c1...)
 - 6b75dc1...** (hash: 6b75dc1...): This is the new **HEAD** and **master** branch. It is labeled **Revert f9f98e4**, indicating it is a revert commit that undoes the changes of C3.

5. Recall the experiment you did in class. Now carefully see the following git-flow. [4 Marks]

```
git init

echo ``Alice`` >> alice.md
git add alice.md; git commit -m ``C1-Alice-1-file``

echo ``Bob`` >> bob.md
git add bob.md; git commit -m ``C2-Bob-2-files``

echo ``Eve`` >> eve.md
echo ``Alice Updated`` >> alice.md
git add eve.md alice.md
git commit -m ``C3-Eve-Alice-3-files``

rm bob.md eve.md
git add .
git commit -m "C4-all deleted-0-files"

echo ``Eve`` >> eve.md
git add eve.md; git commit -m ``C5-Eve-2-files``
```

git cherry-pick <C4-Commit-Hash>

What is the status of the above cherry-pick? What are the files in the directory before and after the above cherry-pick? Justify your answer.

6. What does "md" stand for in readme.md? _____