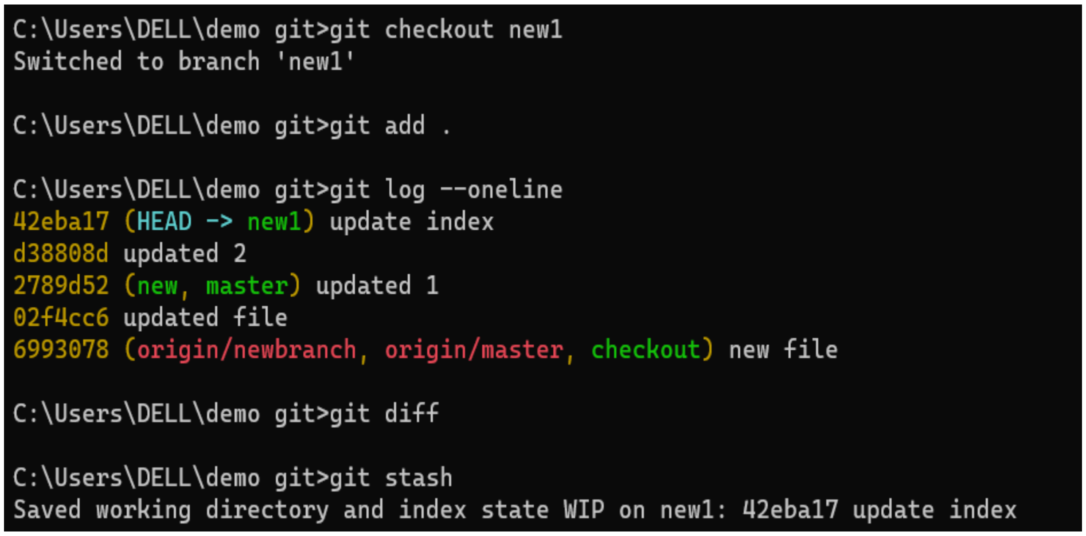
Problem Statement 1: The Advanced Git Guide: Git Stash, Reset, Rebase, and More. developers should have a good understanding of version control, and Git has become the de-facto standard for version control in software development.

Solution:
Git Stash:
The git stash command is used to save changes in your working directory that are not ready to be committed yet. It allows you to switch to a different branch, pull changes, or perform other tasks without committing your current changes. The stashed changes can later be reapplied to your working directory or dropped.

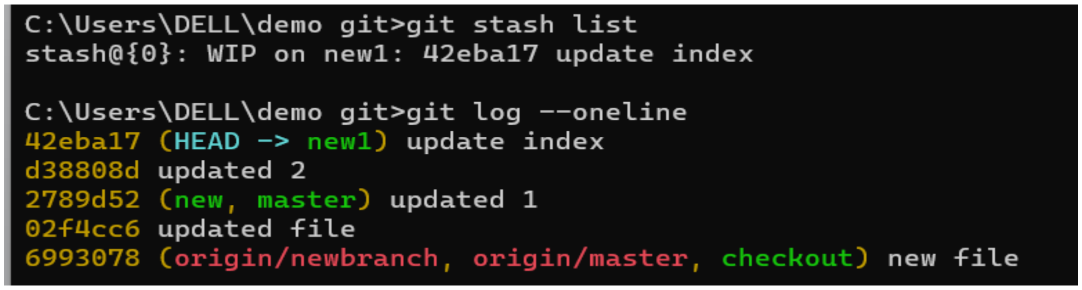
Example:



```
C:\Users\DELL\demo git>git checkout new1
Switched to branch 'new1'

C:\Users\DELL\demo git>git add .

C:\Users\DELL\demo git>git log --oneline
42eba17 (HEAD -> new1) update index
d38808d updated 2
2789d52 (new, master) updated 1
02f4cc6 updated file
6993078 (origin/newbranch, origin/master, checkout) new file

C:\Users\DELL\demo git>git diff

C:\Users\DELL\demo git>git stash
Saved working directory and index state WIP on new1: 42eba17 update index
```

Git stash list:
The git stash list command is used to display a list of stashes that you have created in your Git repository

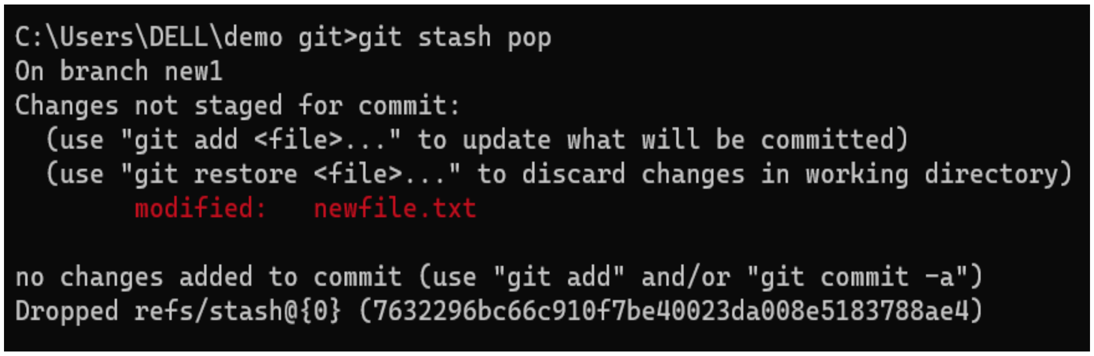
Example:



```
C:\Users\DELL\demo git>git stash list
stash@{0}: WIP on new1: 42eba17 update index

C:\Users\DELL\demo git>git log --oneline
42eba17 (HEAD -> new1) update index
d38808d updated 2
2789d52 (new, master) updated 1
02f4cc6 updated file
6993078 (origin/newbranch, origin/master, checkout) new file
```

Git stash pop:
The git stash pop command is a combination of two operations: applying the changes from the latest stash and then dropping that stash.

Example:

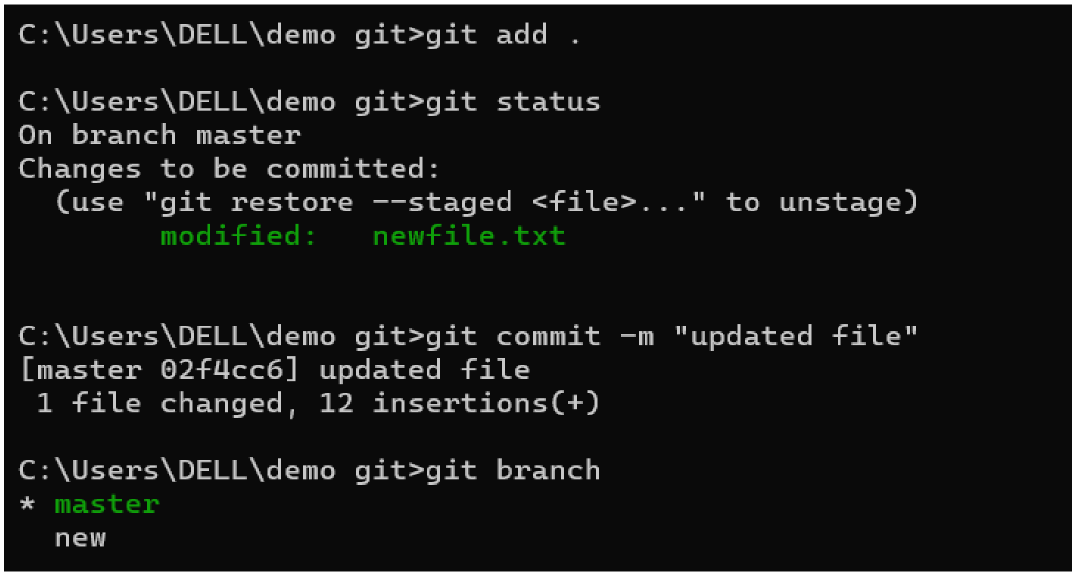


```
C:\Users\DELL\demo git>git stash pop
On branch new1
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   newfile.txt

no changes added to commit (use "git add" and/or "git commit -a")
Dropped refs/stash@{0} (7632296bc66c910f7be40023da008e5183788ae4)
```

Git Rebase:

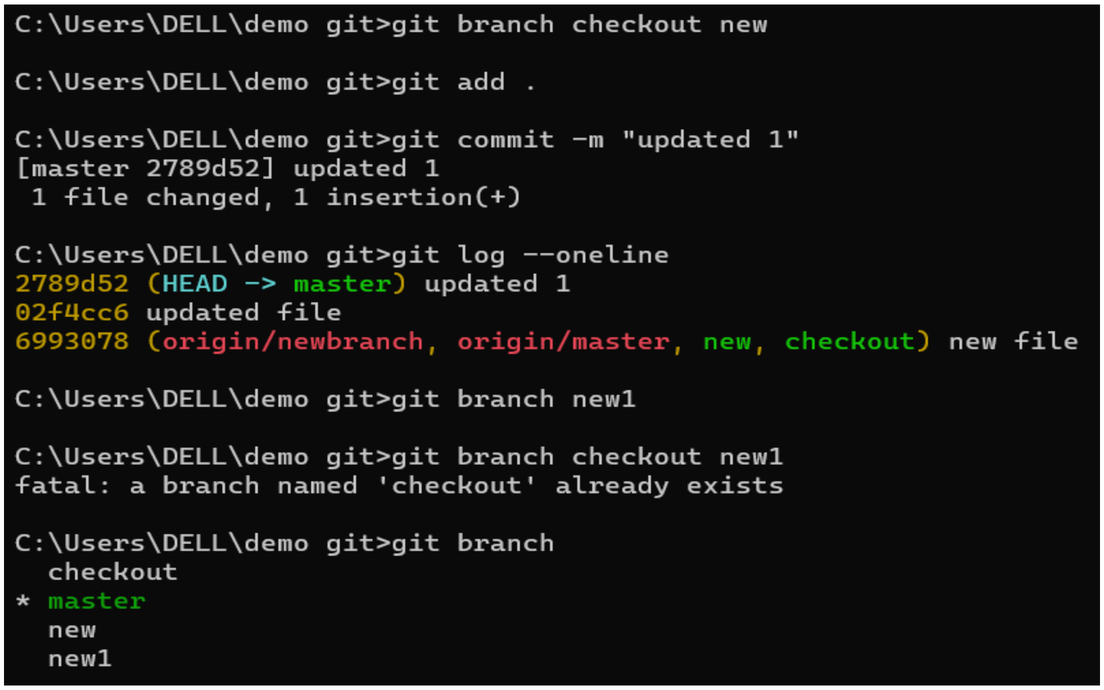
The git rebase command is used to reapply a series of commits onto a different base commit. It's a powerful tool for rewriting commit history, cleaning up your branch, and integrating changes from one branch into another.

Example:
Create a file and modify some contents then add to staging environment



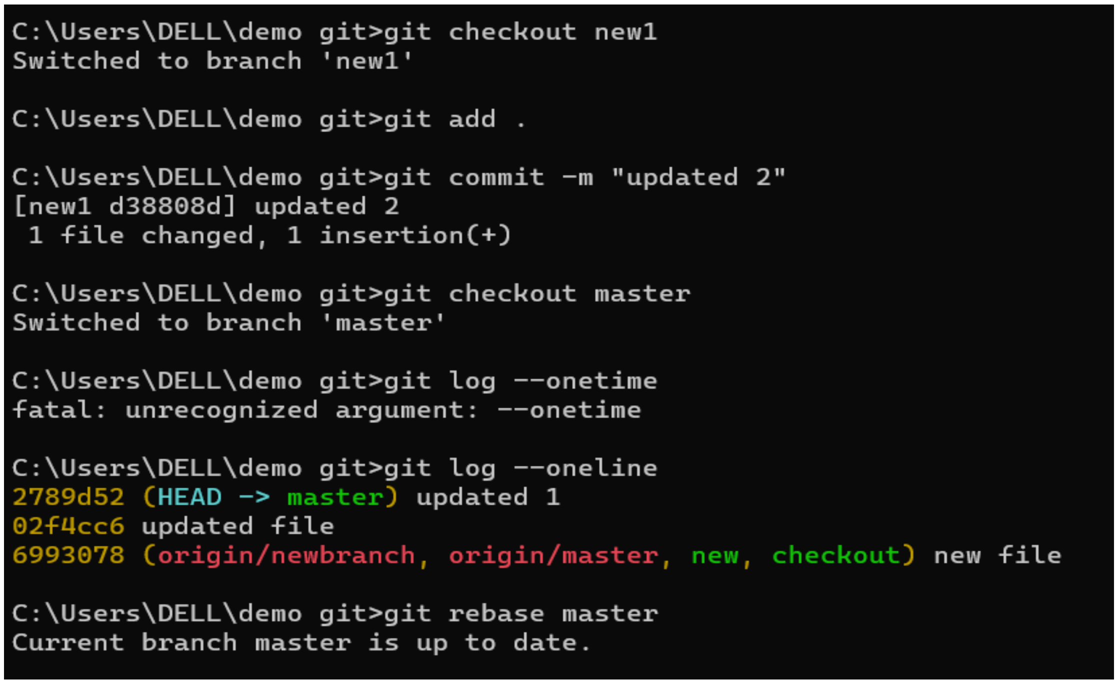
```
C:\Users\DELL\demo git>git add .

C:\Users\DELL\demo git>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   newfile.txt


C:\Users\DELL\demo git>git commit -m "updated file"
[master 02f4cc6] updated file
 1 file changed, 12 insertions(+)

C:\Users\DELL\demo git>git branch
* master
  new
```

Create new branches and make some changes on the file and commit the files

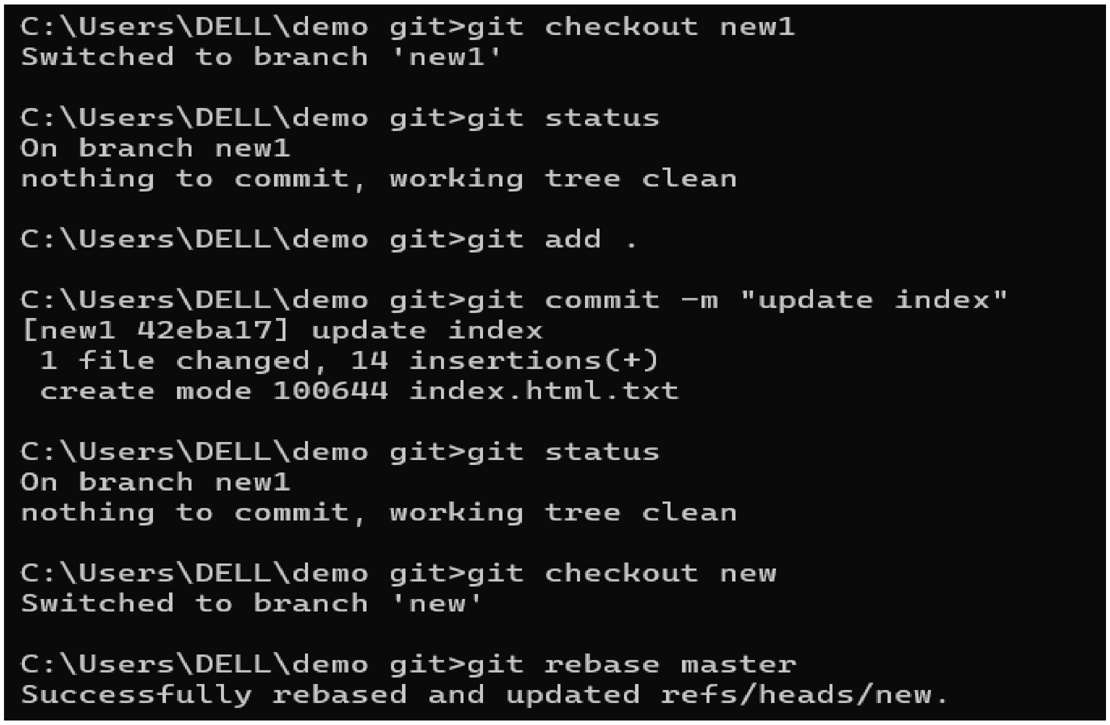


```
C:\Users\DELL\demo git>git branch checkout new

C:\Users\DELL\demo git>git add .

C:\Users\DELL\demo git>git commit -m "updated 1"
[master 2789d52] updated 1
 1 file changed, 1 insertion(+)

C:\Users\DELL\demo git>git log --oneline
2789d52 (HEAD -> master) updated 1
02f4cc6 updated file
6993078 (origin/newbranch, origin/master, new, checkout) new file

C:\Users\DELL\demo git>git branch new1

C:\Users\DELL\demo git>git branch checkout new1
fatal: a branch named 'checkout' already exists

C:\Users\DELL\demo git>git branch
  checkout
* master
  new
  new1
```

Move to new branch and then rebase the branches



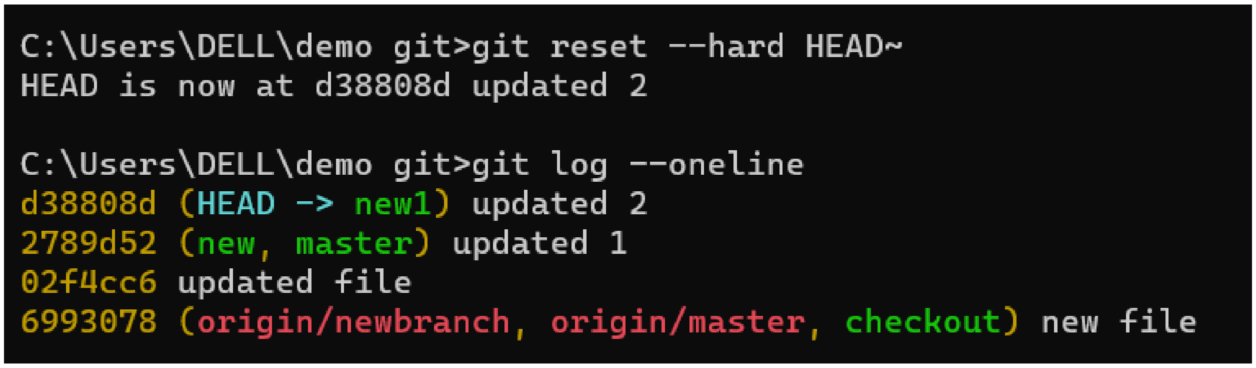
```
C:\Users\DELL\demo git>git checkout new1
Switched to branch 'new1'

C:\Users\DELL\demo git>git add .

C:\Users\DELL\demo git>git commit -m "updated 2"
[new1 d38808d] updated 2
 1 file changed, 1 insertion(+)

C:\Users\DELL\demo git>git checkout master
Switched to branch 'master'

C:\Users\DELL\demo git>git log --onetime
fatal: unrecognized argument: --onetime

C:\Users\DELL\demo git>git log --oneline
2789d52 (HEAD -> master) updated 1
02f4cc6 updated file
6993078 (origin/newbranch, origin/master, new, checkout) new file

C:\Users\DELL\demo git>git rebase master
Current branch master is up to date.
```



```
C:\Users\DELL\demo git>git checkout new1
Switched to branch 'new1'

C:\Users\DELL\demo git>git status
On branch new1
nothing to commit, working tree clean

C:\Users\DELL\demo git>git add .

C:\Users\DELL\demo git>git commit -m "update index"
[new1 42eba17] update index
 1 file changed, 14 insertions(+)
 create mode 100644 index.html.txt

C:\Users\DELL\demo git>git status
On branch new1
nothing to commit, working tree clean

C:\Users\DELL\demo git>git checkout new
Switched to branch 'new'

C:\Users\DELL\demo git>git rebase master
Successfully rebased and updated refs/heads/new.
```

Git reset:

The git reset command is a powerful tool in Git that is used to undo changes in your working directory, staging area, and commit history.



```
C:\Users\DELL\demo git>git reset --hard HEAD~
HEAD is now at d38808d updated 2

C:\Users\DELL\demo git>git log --oneline
d38808d (HEAD -> new1) updated 2
2789d52 (new, master) updated 1
02f4cc6 updated file
6993078 (origin/newbranch, origin/master, checkout) new file
```