Day 11 Task:

Advance Git & GitHub for DevOps Engineers:Part-2

Task-01

- Create a new branch and make some changes to it.

- Use git stash to save the changes without committing them.

- Switch to a different branch, make some changes and commit them.

- Use git stash pop to bring the changes back and apply them on top of the new commits.

➔ Created branch called **features** and edit **version01.txt** . After that use git stash to saves the changes just made without committing them. You can see your stash by using command **git stash show.** Then switch to **dev** branch and edit the same file i.e **version01.txt** and commit changes for dev branch.

Finally, to bring the changes you stashed earlier back and apply them on top of the new commits used **git stash pop** command. this applies the changes you stashed earlier and removes the stash from the stash list. And commit this after stash pop.

After that, the changes from the **features** branch will be applied on top of the new commits from the **dev** branch, and git will prompt a merge conflict for the **version01.txt** file were changed in both branches. Then I resolve the merge conflicts manually by choosing which changes to keep or discard. And below im attaching some screenshot. Take a look on it.

```
G15@DESKTOP-0O26P56 MINGW64 ~/Desktop/git/task/devops/Git (dev)
$ git checkout features
Switched to branch 'features'

G15@DESKTOP-0O26P56 MINGW64 ~/Desktop/git/task/devops/Git (features)
$ vim version01.txt

G15@DESKTOP-0O26P56 MINGW64 ~/Desktop/git/task/devops/Git (features)
$ vim version01.txt git stash
3 files to edit

G15@DESKTOP-0O26P56 MINGW64 ~/Desktop/git/task/devops/Git (features)
$ git stash
Saved working directory and index state WIP on features: a5c33da Merge branch 'main' into features

G15@DESKTOP-0O26P56 MINGW64 ~/Desktop/git/task/devops/Git (features)
$ git stash show
 Git/version01.txt | 1 +
 1 file changed, 1 insertion(+)

G15@DESKTOP-0O26P56 MINGW64 ~/Desktop/git/task/devops/Git (features)
$ git checkout dev
Switched to branch 'dev'

G15@DESKTOP-0O26P56 MINGW64 ~/Desktop/git/task/devops/Git (dev)
$ vim version01.txt

G15@DESKTOP-0O26P56 MINGW64 ~/Desktop/git/task/devops/Git (dev)
$ git add .

G15@DESKTOP-0O26P56 MINGW64 ~/Desktop/git/task/devops/Git (dev)
$ git commit -m "modified version.txt before stash pop"
```

```
MINGW64:/c/Users/G15/Desktop/git/task/devops/Git

G15@DESKTOP-OO26P56 MINGW64 ~/Desktop/git/task/devops/Git (dev)
$ git add .

G15@DESKTOP-OO26P56 MINGW64 ~/Desktop/git/task/devops/Git (dev)
$ git commit -m "modified version.txt before stash pop"
[dev bded2b3] modified version.txt before stash pop
 1 file changed, 1 insertion(+), 1 deletion(-)

G15@DESKTOP-OO26P56 MINGW64 ~/Desktop/git/task/devops/Git (dev)
$ stash pop
bash: stash: command not found

G15@DESKTOP-OO26P56 MINGW64 ~/Desktop/git/task/devops/Git (dev)
$ git stash pop
Auto-merging Git/version01.txt
CONFLICT (content): Merge conflict in Git/version01.txt
On branch dev
Unmerged paths:
  (use "git restore --staged <file>..." to unstage)
  (use "git add <file>..." to mark resolution)
        both modified:   version01.txt

no changes added to commit (use "git add" and/or "git commit -a")
The stash entry is kept in case you need it again.

G15@DESKTOP-OO26P56 MINGW64 ~/Desktop/git/task/devops/Git (dev)
$ git add version01.txt

G15@DESKTOP-OO26P56 MINGW64 ~/Desktop/git/task/devops/Git (dev)
$ git commit -m " modified version.txt after stash pop"
[dev 012ad00]  modified version.txt after stash pop
 1 file changed, 4 insertions(+)

G15@DESKTOP-OO26P56 MINGW64 ~/Desktop/git/task/devops/Git (dev)
$ cat version01.txt
This is first feature of our application
<<<<<<< Updated upstream
This is the bug fix in development branch

this is from feature branch practicing stash from dev
```

Task-02

- In version01.txt of development branch add below lines after "This is the bug fix in development branch" that you added in Day10 and reverted to this commit.

- Line2>> After bug fixing, this is the new feature with minor alteration"

Commit this with message " Added feature2.1 in development branch"

- Line3>> This is the advancement of previous feature

Commit this with message " Added feature2.2 in development branch"

- Line4>> Feature 2 is completed and ready for release

Commit this with message " Feature2 completed"

- All these commits messages should be reflected in Production branch too which will come out from Master branch (Hint: try rebase).

➜ Added this commits one by one from **dev** branch. Then check log of it and also switch to **prod** branch and check its log. There is commit differnce as I didn't commit anything from **prod** branch.

To bring all these commits into **prod** branch, used *git rebase* command to replay commits from **dev** branch on top of the **prod** branch. Below I attach screenshot for log take a look on it.

Note that when you rebase, you'll be moving the entire branch to a new starting point, while preserving all commits' message and change, hence this operation is not a safe one, it will change the commits hash and

it can cause conflict if other people also made changes to the **prod** branch.

```
MINGW64:/c/Users/G15/Desktop/git/task/devops/Git
 1 file changed, 1 insertion(+)

G15@DESKTOP-OO26P56 MINGW64 ~/Desktop/git/task/devops/Git (dev)
$ git log --oneline
7d979b2 (HEAD -> dev)  Feature2 completed
f8fba8b Added feature2.2 in development branch
0b6f926 Added feature2.1 in development branch
978087c version.txt back again in its position after stash practice done
012ad00  modified version.txt after stash pop
bded2b3 modified version.txt before stash pop
73f916f version01.txt modified after stash pop
3f58c35 modified Day01.txt for stash ops
e6fa064 (origin/main, origin/dev, prod, main) Restore the file to a previous version
4e7f3bb Added feature2 in development branch
3e9c372 Added new feature
1ac93b6 added Day-01.txt
370a0a2 added day01 file

G15@DESKTOP-OO26P56 MINGW64 ~/Desktop/git/task/devops/Git (dev)
$ git checkout prod
Switched to branch 'prod'

G15@DESKTOP-OO26P56 MINGW64 ~/Desktop/git/task/devops/Git (prod)
$ git log --oneline
e6fa064 (HEAD -> prod, origin/main, origin/dev, main) Restore the file to a previous version
4e7f3bb Added feature2 in development branch
3e9c372 Added new feature
1ac93b6 added Day-01.txt
370a0a2 added day01 file

G15@DESKTOP-OO26P56 MINGW64 ~/Desktop/git/task/devops/Git (prod)
$ git rebase dev
Successfully rebased and updated refs/heads/prod.

G15@DESKTOP-OO26P56 MINGW64 ~/Desktop/git/task/devops/Git (prod)
$ git log --oneline
7d979b2 (HEAD -> prod, dev)  Feature2 completed
f8fba8b Added feature2.2 in development branch
0b6f926 Added feature2.1 in development branch
978087c version.txt back again in its position after stash practice done
012ad00  modified version.txt after stash pop
bded2b3 modified version.txt before stash pop
73f916f version01.txt modified after stash pop
3f58c35 modified Day01.txt for stash ops
e6fa064 (origin/main, origin/dev, main) Restore the file to a previous version
4e7f3bb Added feature2 in development branch
3e9c372 Added new feature
1ac93b6 added Day-01.txt
370a0a2 added day01 file

G15@DESKTOP-OO26P56 MINGW64 ~/Desktop/git/task/devops/Git (prod)
$
```

Task-03

- In Production branch Cherry pick Commit "Added feature2.2 in development branch" and added below lines in it:

- Line to be added after Line3>> This is the advancement of previous feature

- Line4>>Added few more changes to make it more optimized.

- Commit: Optimized the feature

➔ Cherry-pick the commit you want to bring over from development branch used *git cherry-pick f8fba8b*

f8fba8b This is commit id from dev branch commit i.e. "Added feature2.2 in development branch". I Gt conflict there so I resolve it and add "Line4>>Added few more changes to make it more optimized" & save and stage thye changes. For continue cherry-pick I used *git cherry-pick –continue.* And write commite massge and saved it. Also take a look on some screenshot

```
This is first feature of our application
After bug fixing, this is the new feature with minor alteration
This is the advancement of previous feature
<<<<<<< HEAD
 Feature 2 is completed and ready for release
=======
>>>>>>> f8fba8b (Added feature2.2 in development branch)
~
~
~
```

```
G15@DESKTOP-OO26P56 MINGW64 ~/Desktop/git/task/devops/Git (prod)
$ git cherry-pick f8fba8b
Auto-merging Git/version01.txt
CONFLICT (content): Merge conflict in Git/version01.txt
error: could not apply f8fba8b... Added feature2.2 in development branch
hint: After resolving the conflicts, mark them with
hint: "git add/rm <pathspec>", then run
hint: "git cherry-pick --continue".
hint: You can instead skip this commit with "git cherry-pick --skip".
hint: To abort and get back to the state before "git cherry-pick",
hint: run "git cherry-pick --abort".

G15@DESKTOP-OO26P56 MINGW64 ~/Desktop/git/task/devops/Git (prod|CHERRY-PICKING)
$ vim version01.txt

G15@DESKTOP-OO26P56 MINGW64 ~/Desktop/git/task/devops/Git (prod|CHERRY-PICKING)
$ git add version01.txt

G15@DESKTOP-OO26P56 MINGW64 ~/Desktop/git/task/devops/Git (prod|CHERRY-PICKING)
$ git cherry-pick --continue
[prod d1c9b45] Optimized the feature Added feature2.2 in development branch
 Date: Wed Jan 11 10:21:20 2023 +0530
 1 file changed, 1 insertion(+), 1 deletion(-)

G15@DESKTOP-OO26P56 MINGW64 ~/Desktop/git/task/devops/Git (prod)
$ git log --oneline
d1c9b45 (HEAD -> prod) Optimized the feature Added feature2.2 in development branch
7d979b2 (dev)  Feature2 completed
f8fba8b Added feature2.2 in development branch
0b6f926 Added feature2.1 in development branch
978087c version.txt back again in its position after stash practice done
012ad00  modified version.txt after stash pop
bded2b3 modified version.txt before stash pop
73f916f version01.txt modified after stash pop
3f58c35 modified Day01.txt for stash ops
e6fa064 (origin/main, origin/dev, main) Restore the file to a previous version
4e7f3bb Added feature2 in development branch
3e9c372 Added new feature
1ac93b6 added Day-01.txt
370a0a2 added day01 file

G15@DESKTOP-OO26P56 MINGW64 ~/Desktop/git/task/devops/Git (prod)
$
```