

## Homework: III

Team Name: Peaky Blinders

Roll Numbers:11940150, 11940370, 11940380

**Solution of problem 4.** The answers are provided below:

a The following commands should be used in part 1 of the question.

```
#Started making changes in main branch for traversal
#Urgency for new func comes up
git stash push #Stash the unfinished changes
git checkout -b NewFunc #Create and checkout the NewFunc Branch
#Write the NewFunc code and commit it
git add BinaryTree.cpp
git commit -m "new func added"
git checkout main #Return to master branch
git stash pop stash@{0} #Apply the unfinished work and also pop it from the stash
#Complete the work in main branch and commit
git add BinaryTree.cpp
git commit -m "Traversal code updated"
git merge NewFunc #merge the NewFunc branch into main
git push origin main "Push the changes to the remote repository"
```

The stash function is useful in situations as described in the question. You just stash the modified files and move to the other branch. Then whenever you want to resume the work pop from the stash, complete it and commit. The stash command saves all the modified and staged changes into and also removes them from the working directory. A separate commit is made for this purpose but the it is done outside the current branch and thus the HEAD does not move to this commit

b For situation in part 2 use the following commands:

```
git add f1.h f1.cpp f2.cpp f3.cpp #stage unfinished changes before changing branch
git stash #Stash the unfinished changes
git checkout -b NewFunc2 #create the new branch and checkout in it
#add the new function to the code
git add BinaryTree.cpp #stage the new func added
git commit -m "new func 2 added"
git checkout main #return to main(master) branch
git stash list #view the stashed changes
git stash pop stash@{0} #apply and drop the relevant changes from the stash
#We only want to keep 2 of 3 functionalities being developed
#All have been already staged so unstage the function to be removed
#Here f1 is removed
git restore --staged f1.cpp #unstage f1.cpp and f1.h
```

```
git restore --staged f1.h
#complete the remaining 2 functions
#f2.cpp and f3.cpp were already staged now also stage f2.h and f3.h
#Also make necessary changes in main code
git add f2.h f3.h BinaryTree.cpp #stage the header files for f2 and f3
git commit -m "new functions added in main"
#f1 can be discarded from the wd as now it is unstaged
git merge NewFunc2 #Merge NewFunc2 branch into the main
git push origin main #push to remote repository
```

Here again stashing was used to save unfinished changes. Also initially we were working on 3 functions f1, f2 and f3(.cpp and .h files). Then we moved to the NewFunc2 branch to add another function and after finishing those we came back to the main branch and retrieved the stashed changes. Here f1 was removed(first unstaged using git restore --staged and then deleted from wd) and f2 and f3 were completed(.cpp and .h files) and then committed to the main branch. Finally the NewFunc2 was merged with main.(NewFunc2 branch was locally created)

The repository used for this is

[https://github.com/Dhruv88/CS200\\_hw3\\_q4](https://github.com/Dhruv88/CS200_hw3_q4)

Some of the C++ codes have been taken from geeksforgeeks and tutorialspoint.