# Branching & Rebasing Operations

1. Create a branch ‘bug’, checkout to bug branch to solve the bug in file1.

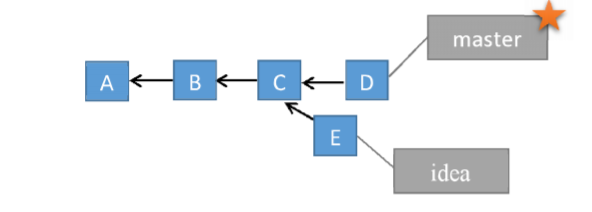
Bug is – Put a “ ” (space) between numerical and letters

E.g.: - Change1 -> Change 1

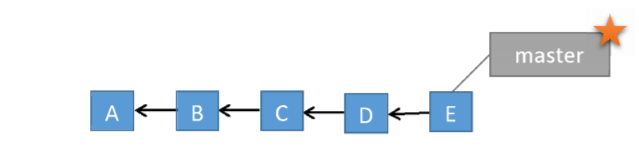
1. Commit this change with message as “Bug Resolved – Space added between numerical and letters”
2. Merge the changes to master assuming bug is resolved
3. Delete the branch ‘bug’
4. Recover the deleted branch ‘bug’ and rename it to ‘bug123’
5. Consider the following scenario and achieve the same to Rebase

* Create a new branch ‘idea’ and make a new commit on ‘idea’ branch
* Now checkout to master and make a new commit on master

Refer the below image: -



1. Now use Rebase to get the linear story line as follows: -



1. Now say bug is resolved and you are good to release this stable version, tag this release as ‘v1.0’ with message as “Stable version 1.0 released”.
2. Consider the following scenario to work with Stashing: -

Checkout to ‘bug123’ branch first and add text as “Change 3” in File1.txt, stage this change and don’t commit the same so that you can get a dirty state of working directory.

Now assume that you have a priority task to be made on master branch, checkout to master and store this dirty state (on ‘bug123’) to stash so that you get back to this state later.

1. Display the list of all stash.
2. Checkout to ‘bug123’ branch, reapply the stash which is stored recently and finally commit this change with message as “Change 3 in File1”