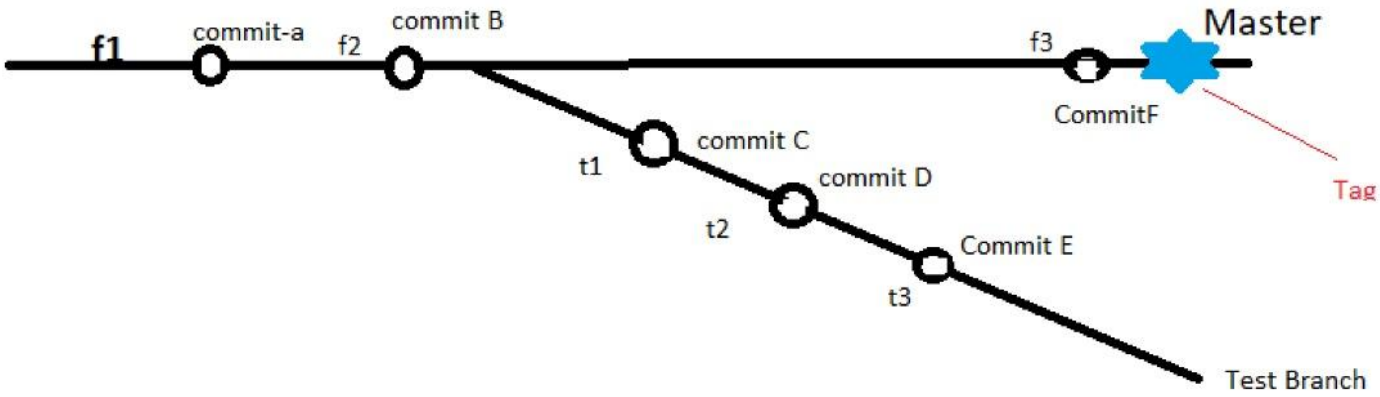
BRANCHING




```
MyPC@Bharath MINGW64 ~/Desktop/Project1
$ git init
Initialized empty Git repository in C:/Users/Desktop
/Project1/.git/

MyPC@Bharath MINGW64 ~/Desktop/Project1 (master)
$ touch f1

MyPC@Bharath MINGW64 ~/Desktop/Project1 (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be
committed)
        f1

nothing added to commit but untracked files present
(use "git add" to track)

MyPC@Bharath MINGW64 ~/Desktop/Project1 (master)
$ git add .
```

```
MyPC@Bharath MINGW64 ~/Desktop/Project1 (master)
$ git commit -m "Commit -- A"
[master (root-commit) dd3c51a] Commit -- A
 Committer: Bharath Gowda R
<bharathgowda.r@outlook.com>

 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 f1

MyPC@Bharath MINGW64 ~/Desktop/Project1 (master)
$ touch f2

MyPC@Bharath MINGW64 ~/Desktop/Project1 (master)
$ git add f2

MyPC@Bharath MINGW64 ~/Desktop/Project1 (master)
$ git commit -m "Commit -- B"
[master 89e0985] Commit -- B
 Committer: Bharath Gowda R
<bharathgowda.r@outlook.com>
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 f2

MyPC@Bharath MINGW64 ~/Desktop/Project1 (master)
$
```

Note:
Assume we have got new functionality to work on, to start working on that, lets create a new branch (name: testbranch) & start working on that branch parallelly

```
MyPC@Bharath MINGW64 ~/Desktop/Project1 (master)
$ git branch testbranch

MyPC@Bharath MINGW64 ~/Desktop/Project1 (master)
$ git branch
* master
  testbranch

MyPC@Bharath MINGW64 ~/Desktop/Project1 (master)
```

```
$ git checkout testbranch
Switched to branch 'testbranch'

MyPC@Bharath MINGW64 ~/Desktop/Project1 (testbranch)
$ ls
f1  f2

MyPC@Bharath MINGW64 ~/Desktop/Project1 (testbranch)
$ touch t1

MyPC@Bharath MINGW64 ~/Desktop/Project1 (testbranch)
$ git add t1

MyPC@Bharath MINGW64 ~/Desktop/Project1 (testbranch)
$ git commit -m "Commit -- C"
[testbranch a4b2246] Commit -- C
 Committer: Bharath Gowda R
<bharathgowda.r@outlook.com>

    git commit --amend --reset-author

 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 t1


MyPC@Bharath MINGW64 ~/Desktop/Project1 (testbranch)
$ touch t2

MyPC@Bharath MINGW64 ~/Desktop/Project1 (testbranch)
$ git add t2

MyPC@Bharath MINGW64 ~/Desktop/Project1 (testbranch)
$ git commit -m "Commit -- D"
[testbranch 367926c] Commit -- D
 Committer: Bharath Gowda R
<bharathgowda.r@outlook.com>

    git commit --amend --reset-author

 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 t2

MyPC@Bharath MINGW64 ~/Desktop/Project1 (testbranch)
$ touch t3

MyPC@Bharath MINGW64 ~/Desktop/Project1 (testbranch)
```

```
$ git add t3

MyPC@Bharath MINGW64 ~/Desktop/Project1 (testbranch)
$ git commit -m "Commit -- E"
[testbranch 2b9adf7] Commit -- E
 Committer: Bharath Gowda R
<bharathgowda.r@outlook.com>

    git commit --amend --reset-author

 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 t3

MyPC@Bharath MINGW64 ~/Desktop/Project1 (testbranch)
$
```

Note:
Now If our work in test branch is completed we can switch back to **master** branch & proceed working there

```
MyPC@Bharath MINGW64 ~/Desktop/Project1 (testbranch)
$ git branch
  master
* testbranch

MyPC@Bharath MINGW64 ~/Desktop/Project1 (testbranch)
$ git checkout master
Switched to branch 'master'

MyPC@Bharath MINGW64 ~/Desktop/Project1 (master)
$ git branch
* master
  testbranch

MyPC@Bharath MINGW64 ~/Desktop/Project1 (master)
$ ls
f1  f2
```

Note:
files (t1 t2 t3) & commits created on testbranch will not be visible in master branch until we merge testbranch with master

```
MyPC@Bharath MINGW64 ~/Desktop/Project1 (master)
$ git log --oneline
89e0985 (HEAD -> master) Commit -- B
dd3c51a Commit -- A

MyPC@Bharath MINGW64 ~/Desktop/Project1 (master)
$ touch f3

MyPC@Bharath MINGW64 ~/Desktop/Project1 (master)
$ git add f3

MyPC@Bharath MINGW64 ~/Desktop/Project1 (master)
$ git commit -m "Commit -- F"
[master 5c4c68e] Commit -- F
 Committer: Bharath Gowda R
<bharathgowda.r@outlook.com>

    git commit --amend --reset-author

 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 f3

MyPC@Bharath MINGW64 ~/Desktop/Project1 (master)
$ git log --oneline
5c4c68e (HEAD -> master) Commit -- F
89e0985 Commit -- B
dd3c51a Commit -- A

MyPC@Bharath MINGW64 ~/Desktop/Project1 (master)
$
```

Now since work on testbranch has completed, lets merge testbranch changes to master branch

```
MyPC@Bharath MINGW64 ~/Desktop/Project1 (master)
$ git merge testbranch
Merge made by the 'ort' strategy.
 t1 | 0
 t2 | 0
```

```
 t3 | 0
 3 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 t1
 create mode 100644 t2
 create mode 100644 t3

MyPC@Bharath MINGW64 ~/Desktop/Project1 (master)
$ git log --oneline
1aaa669 (HEAD -> master) This is coming from my merge
commit Merge branch 'testbranch'
5c4c68e Commit -- F
2b9adf7 (testbranch) Commit -- E
367926c Commit -- D
a4b2246 Commit -- C
89e0985 Commit -- B
dd3c51a Commit -- A

MyPC@Bharath MINGW64 ~/Desktop/Project1 (master)
$
```

Observations:

Commits in master

i. **Before merging testbranch:**
   Commit A , Commit c, Commit F

ii. **After merging testbranch:**
   Commit A, Commit B Commit C Commit D,Commit E, Commit F