[day-3] Git command

* Branch
  ↳ Bug fix
  ↳ Add New feature.

- Copy of a same source code.
  ↳ main copy
  New Feature.↵ ↳ Bugfix

* HEAD → master
* git branch - all branch & active
  * master
* git branch bugfix
  git branch
* git checkout bugfix
* git branch
* master Bugfix
  A A
  B
* git checkout branch (switch)
* [bugfix ___] B

- - - -

Bugfix branch

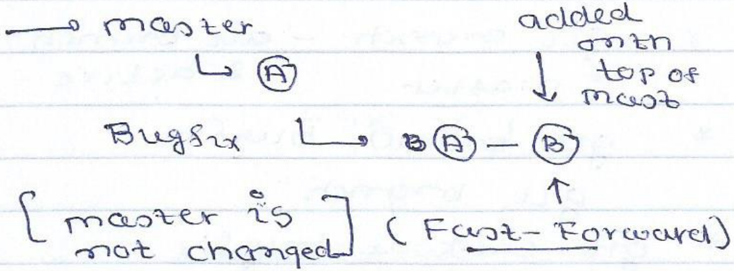
> git merge bugfix ↳ master

Fast-Forward

↳ git log

Commit.... (HEAD → master, bugfix)

* Fast Forward

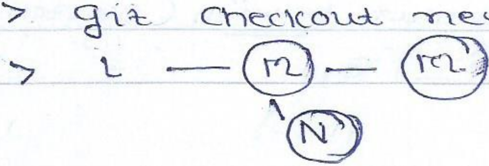


> delete git branch -d bugfix

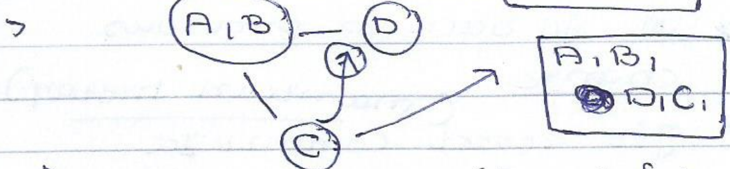
Deleted branch bugfix (_)

> git branch newfeature1

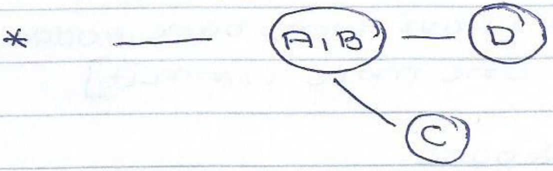
> git checkout newfeature1.



A,B ~~B~~ C,D , A,B,D,C



> One more commit id for merge.



* git branch exp

git checkout exp.

master - 5 (File)

↳ exp - 5 (File)

↳ conflict

↳ Automatic merge failed ---

* Rollback

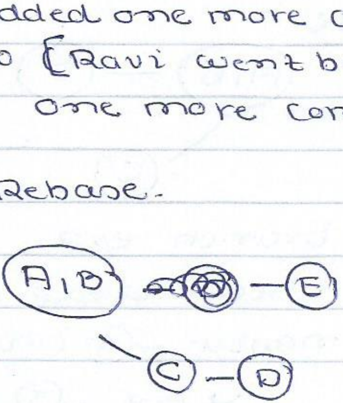
git reset --hard commitId

> [Head is now at 46bcfg] 46bcfa

* go back to previous change.
(maintain history)

* git revert commitId
(add message.
↳ added one more commit to [Ravi went back, adds one more commit]

* git Rebase.



git branch new

> git Rebase new

> git log --oneline

> git checkout -b cube

```
> git Log --oneline
    E
  D
  C
  B
  A
```

```
> git cherry-pic commitId
```

→ only specific commit will be part of master branch.

```
> git diff --summary
          ae475  5c69476
```

create

* Continuous Integration *

↳ Executable ≈ Artifact.

Source code → Artifact.

$\boxed{\text{Github}} \xrightarrow{\text{Artifact}}$ .jar
.war
.ear
.exe

Process → Build

Source → Artifact called Build

1. compile (high Level to machine understand)

(Sonar, pmd, check style) 2. Code Review (coding Standard) → heap memory (unnecessory code.)

3. Unit-test (Not Functional code)

~~4~~

4. Code Coverage

5. Package (generating artifacts)

* Build tools
- Ant
- maven
- Gradle
- msbuild

→ how maven knows every commit

* ~~Buil~~ jenkins - helps to run the code

(sense the change)

↳ jenkins $\xrightarrow[\text{watching}]{\text{keep}}$ $\boxed{\text{Github}}$

(Not a Build tool but automation tool) →(help to Run all steps)

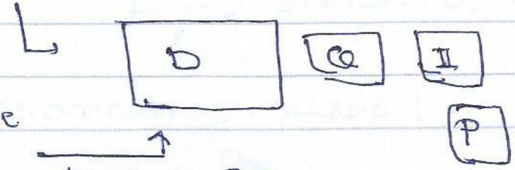
↳[Bamboo, teamcity] Like jenkins

↳ Continous Integration

jenkins → creates Artifacts

↳ continous Building in each commits

↳

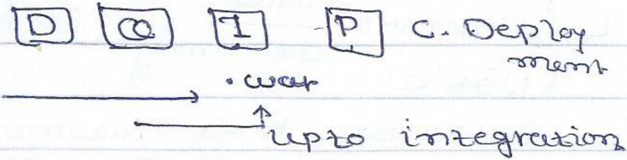


.jar file

deploy on the Dev, QA, Integration production

Continous deployment

But not deploy
C. Delivery



Not to deploy on Production

Delivery | Deploymen
| ↓
Integr | Production

> git
> Java (Java-path)
> maven (maven_path)
> jenkins (...)

- (setup-jenkins.sh)

{Control + R}

* java

* xcode

* git

* Proxy → cntlm

*

```
> git branch -d branch
> git branch -D branch
```

-d → merged branch

-D → non-merged branch