A. SETTING UP SSH KEY (for a secure and passwordless git connection between a local system and remote repo):

All are Terminal commands for Mac (use DOS command prompt for Windows).

1. Creating a SSH key:

$ ssh-keygen -t rsa -b 4096 -C "myname@mydomain.com"

2. Adding the SSH:

$ ssh-add ~/.ssh/id_rsa

3. Show and copy the SSH key:

$ cat ~/.ssh/id_rsa (to view and copy the key) or

$ pbcopy < ~/.ssh/id_rsa.pub (to copy the key in the clipboard)

-----BEGIN OPENSSH PRIVATE KEY-----

ssh-rsa
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAACFwAAA2gtcnwrfuvwprtlEjS
. . . .
. . . .
. . . .
VWxoUoX2BtFekTHJYFsumkBVMtkYhYBm85aWg00UT5w6hNJlxjwwj0lC2ccgQxkkAAAAUZW1ydWxoYXNhbkBnbWFpbC5jb20BAgMEBQYH== mymail@domain.com

-----END OPENSSH PRIVATE KEY-----

4. Go to GitHub.com and sign in. Navigate to Settings > SSH and GPG keys > New SSH key. Copy the key everything from *ssh-rsa* to *mymail@domain.com* and paste in GitHub > Settings > SSH and GPG keys > New SSH key. Title: description for the new key, e.g. *My MacBook Air*. **Key: Paste the key into the Key field. Click** Add SSH key**.** Enter your GitHub password.

B. COMMTTING CODE FROM TERMINAL / COMMAND PROMPT:

1. Navigate to local repo folder (myrepo).

$ git init (to initialize local empty Git repo folder (myrepo))

2. Configure user name and email – as used in GitHub account:

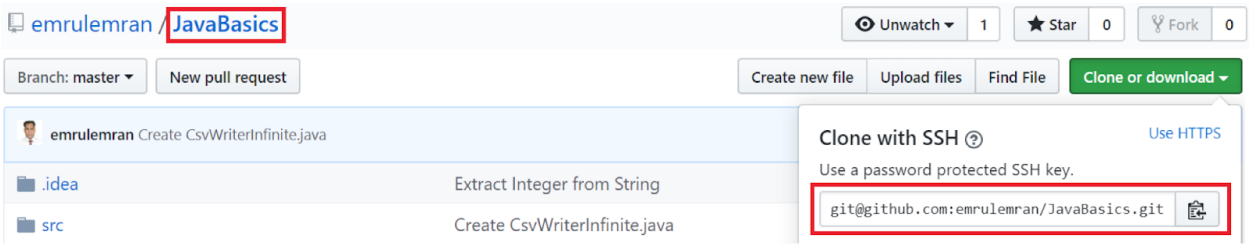
git config --global user.email "you@example.com"
git config --global user.name "Your Name"

3. Check if user name and email has been configured properly:

$ git config --global user.email
$ git config --global user.name

4. Cloning with SSH: Copy GitHub (remote) repo to local. *JavBasics* will be copied to local if we use the highlighted link below.

$ git clone git@github.com:emrulemran/***JavaBasics***.git



5. Navigate to JavaBasics folder inside the local repo folder (myrepo):

$ cd JavaBasics/

6. Check status:

$ git status

7. Edit, make changes to any file and save inside the local repo folder (myrepo). Besides using IDEs, we can use **vi editor to modify source codes**:

$ vi StringToCharArray
i to enter Insert mode
Esc
:w to enter Write mode and press Enter
:q to quit the vi editor

8. Add file modified file:

$ git add . (. or * or specific filename with extension)

9. Check status again:

$ git status

10. To push changed code:

$ git push
$ git push -u origin master

11. Commit the modified file:

$ git commit -m "checking git"

Other Git commands:

$ git log (to check log of changes made to Git repo)
$ git branch (to check branch status)
$ git checkout -b *testbranch* (create) (to create a new branch named *testbranch*)
$ git branch -d *tempbranch* (to delete a branch name *tempbranch*)
$ git push --set-upstream origin *testbranch* (to push code to remote repo named *testbranch*)