Rachit Bodhare

1. Set your username and email in git config

2. Create a new branch named "feature-branch" and switch to it.

3. List all branches in the repository.

4. Delete the branch “feature-branch”

5. How do you undo the last commit

a) git reset --soft HEAD~1 This will move the HEAD pointer back to the previous commit and keep your changes in the staging area.

b) git reset --hard HEAD~1 This will move the HEAD pointer back to the previous commit and discard all changes.

6. Create a new branch names “conflict-branch”

7. Create a another branch named “feature1”

8. Make some changes in to feature1 branch

9. Merge “feature1” branch into main branch

10. Make changes in “conflict-branch”, in the same file and line that you had made changes

in feature1

11. Merge master into conflict-branch [Attach screenshot of terminal & file]

12. Resolve merge conflicts

13. Add a remote named "origin" pointing to a GitHub repository.

14. Fork a repository on GitHub and clone it to your local machine.

15. Create a new branch on your fork, make changes, and open a pull request to the original

repository.

16. Comment on a PR and suggest improvements

17. Create a Git alias for the command `git log --oneline` named `gitlol`.

18. Create a pre-commit hook

19. You have made local changes in your branch, but you need to switch to another branch

urgently without committing. How would you handle this situation?

20. You accidentally deleted a file in your local repository. How do you restore it using Git?

21. You have committed changes to your branch but forgot to include a file. How do you add

the file to the last commit without creating a new commit?

22. You want to discard all changes in your working directory and revert to the last commit.

What Git command would you use?

23. You need to view a specific commit's changes. What Git command can be used to show

the changes introduced by a particular commit?

24. You want to change a commit message, after you have already committed, how do you

do so?

25. Your colleague has made changes in their branch, and you want to incorporate those

changes into your branch without merging. How do you achieve this?

26. You've made several commits on a branch, but you want to club them into a single

commit before pushing to the remote repository. How would you do that?

27. You accidentally staged a file that you don't want to commit. How do you unstage it?

28. You don’t want to commit files that have .yml in the end, and also files inside folder

config. How do you do that?

29. You want to see a list of all the files changed in the last commit. What Git command

would you use?

30. You realize that your local branch is outdated, and you want to fetch the latest changes

from the remote repository. How do you do this without merging?

31. You accidentally deleted a branch. How do you recover it?

32. You want to remove untracked files and directories from your working directory. What Git

command would you use?

33. You have a commit from a feature branch that you want to apply to the main branch

without merging the entire feature branch.

34. You mistakenly committed a change to the wrong branch and need to apply that commit

to the correct branch.

35. There is a series of commits on a feature branch, but you only want to cherry-pick a

specific range of commits.

36. You want to clone a GitHub repository onto your local machine, but you only need a

specific branch. How can you achieve this?

37. You've made changes to your local repository and want to push them to your fork on

GitHub. What Git commands would you use?

38. You want to create a new branch both locally and on GitHub to work on a new feature.

What commands would you use?

39. You want to see the commit history of a GitHub repository. How can you do this using Git

commands?

40. You've accidentally committed sensitive information and want to remove the commit from

both your local and remote repositories on GitHub. What commands would you use?

41. You want to delete a remote branch on GitHub. What Git command would you use?

42. Create a git repository for all your assignments and upload them in it. Ask your peers to

code review it, and you need to code review your peers assignments

43. Create a pull request on any open source library on github, attach the pull request link to

the readme file of this project’s repository

19. You have made local changes in your branch, but you need to switch to another branch urgently without committing. How would you handle this situation?

git stash # Temporarily stores your local changes

git checkout other-branch # Switches to another branch

20. You accidentally deleted a file in your local repository. How do you restore it using Git?

git checkout HEAD -- path/to/file # Retrieves the deleted file from the last commit

21. You have committed changes to your branch but forgot to include a file. How do you add the file to the last commit without creating a new commit?

git add forgotten-file

git commit --amend # Adds the file to the last commit

22. You want to discard all changes in your working directory and revert to the last commit. What Git command would you use?

git reset --hard HEAD # Resets your working directory to the last commit

23. You need to view a specific commit's changes. What Git command can be used to show the changes introduced by a particular commit?

git show commit-hash # Displays the changes made by a specific commit

24. You want to change a commit message, after you have already committed, how do you do so?

git commit --amend # Allows you to edit the last commit message

25. Your colleague has made changes in their branch, and you want to incorporate those changes into your branch without merging. How do you achieve this?

git cherry-pick colleague-branch-commit # Applies a specific commit from another branch to yours

26. You've made several commits on a branch, but you want to club them into a single commit before pushing to the remote repository. How would you do that?

git rebase -i HEAD~number-of-commits # Squashes multiple commits into one during interactive rebase

27. You accidentally staged a file that you don't want to commit. How do you unstage it?

git reset HEAD file # Unstages the file while keeping the changes in your working directory

28. You don’t want to commit files that have .yml in the end, and also files inside folder config. How do you do that?

- Add to `.gitignore`:

\*.yml

/config/

- This prevents Git from tracking files matching these patterns.

29. You want to see a list of all the files changed in the last commit. What Git command would you use?

git diff --name-only HEAD~1 # Lists files changed in the last commit

30. You realize that your local branch is outdated, and you want to fetch the latest changes from the remote repository. How do you do this without merging?

git fetch origin # Fetches changes from the remote repository without merging

31. You accidentally deleted a branch. How do you recover it?

git reflog # Displays the commit history including deleted branches

git checkout -b deleted-branch commit-hash # Recovers the deleted branch from its commit history

32. You want to remove untracked files and directories from your working directory. What Git command would you use?

git clean -fd # Removes untracked files and directories forcefully

33. You have a commit from a feature branch that you want to apply to the main branch without merging the entire feature branch.

git checkout main

git cherry-pick commit-hash # Applies a specific commit from another branch to yours

34. You mistakenly committed a change to the wrong branch and need to apply that commit to the correct branch.

git checkout correct-branch

git cherry-pick commit-hash # Applies the commit to the correct branch

git checkout wrong-branch

git reset --hard HEAD~1 # Removes the commit from the wrong branch

35. There is a series of commits on a feature branch, but you only want to cherry-pick a specific range of commits.

git cherry-pick start-commit..end-commit # Cherry-picks a range of commits to your current branch

36. You want to clone a GitHub repository onto your local machine, but you only need a specific branch. How can you achieve this?

git clone -b branch-name --single-branch repo-url # Clones a specific branch from the remote repository

37. You've made changes to your local repository and want to push them to your fork on GitHub. What Git commands would you use?

git add .

git commit -m "Your commit message"

git push origin branch-name # Pushes changes to your fork on GitHub

38. You want to create a new branch both locally and on GitHub to work on a new feature. What commands would you use?

git checkout -b new-feature-branch # Creates and switches to a new branch locally

git push -u origin new-feature-branch # Pushes the new branch to GitHub

39. You want to see the commit history of a GitHub repository. How can you do this using Git commands?

git log # Displays the commit history of the repository

40. You've accidentally committed sensitive information and want to remove the commit from both your local and remote repositories on GitHub. What commands would you use?

git reset --hard HEAD~1 # Removes the commit locally

git push --force # Forces the removal of the commit from the remote repository

41. You want to delete a remote branch on GitHub. What Git command would you use?

git push origin --delete branch-name # Deletes a branch from the remote repository on GitHub

ganes@LAPTOP-1U5V6JI4 MINGW64 ~

$ ^[[200~git config --global user.name "Your Name"

bash: $'\E[200~git': command not found

ganes@LAPTOP-1U5V6JI4 MINGW64 ~

$ git config --global user.name LunaticSculptor

ganes@LAPTOP-1U5V6JI4 MINGW64 ~

$ git config --global user.email rachitbodhare@gmail.com

ganes@LAPTOP-1U5V6JI4 MINGW64 ~

$ git config --global --list

core.editor="C:\Users\ganes\AppData\Local\Programs\Microsoft VS Code\bin\code" --wait

user.email=rachitbodhare@gmail.com

user.name=LunaticSculptor

ganes@LAPTOP-1U5V6JI4 MINGW64 ~

$ git config --list

diff.astextplain.textconv=astextplain

filter.lfs.clean=git-lfs clean -- %f

filter.lfs.smudge=git-lfs smudge -- %f

filter.lfs.process=git-lfs filter-process

filter.lfs.required=true

http.sslbackend=openssl

http.sslcainfo=C:/Program Files/Git/mingw64/etc/ssl/certs/ca-bundle.crt

core.autocrlf=true

core.fscache=true

core.symlinks=false

core.fsmonitor=true

pull.rebase=false

credential.helper=manager

credential.https://dev.azure.com.usehttppath=true

init.defaultbranch=main

core.editor="C:\Users\ganes\AppData\Local\Programs\Microsoft VS Code\bin\code" --wait

user.email=rachitbodhare@gmail.com

user.name=LunaticSculptor

ganes@LAPTOP-1U5V6JI4 MINGW64 ~

$ cd C:\Users\ganes\Desktop\hellogitworld-master

bash: cd: C:UsersganesDesktophellogitworld-master: No such file or directory

ganes@LAPTOP-1U5V6JI4 MINGW64 ~

$ cd ^[[200~C:/Users/ganes/Desktop/hellogitworld-master

bash: cd: $'\E[200~C:/Users/ganes/Desktop/hellogitworld-master': No such file or directory

ganes@LAPTOP-1U5V6JI4 MINGW64 ~

$ cd C:/Users/ganes/Desktop/hellogitworld-master

ganes@LAPTOP-1U5V6JI4 MINGW64 ~/Desktop/hellogitworld-master

$ git init

Initialized empty Git repository in C:/Users/ganes/Desktop/hellogitworld-master/.git/

ganes@LAPTOP-1U5V6JI4 MINGW64 ~/Desktop/hellogitworld-master (main)

$ git remote add origin https://github.com/LunaticSculptor/test.git

ganes@LAPTOP-1U5V6JI4 MINGW64 ~/Desktop/hellogitworld-master (main)

$ git add .

warning: in the working copy of 'hellogitworld-master/.gitattributes', LF will be replaced by CRLF the next time Git touches it

warning: in the working copy of 'hellogitworld-master/.gitignore', LF will be replaced by CRLF the next time Git touches it

warning: in the working copy of 'hellogitworld-master/.travis.yml', LF will be replaced by CRLF the next time Git touches it

warning: in the working copy of 'hellogitworld-master/README.txt', LF will be replaced by CRLF the next time Git touches it

warning: in the working copy of 'hellogitworld-master/build.gradle', LF will be replaced by CRLF the next time Git touches it

warning: in the working copy of 'hellogitworld-master/pom.xml', LF will be replaced by CRLF the next time Git touches it

warning: in the working copy of 'hellogitworld-master/resources/labels.properties', LF will be replaced by CRLF the next time Git touches it

warning: in the working copy of 'hellogitworld-master/runme.sh', LF will be replaced by CRLF the next time Git touches it

warning: in the working copy of 'hellogitworld-master/src/Division.groovy', LF will be replaced by CRLF the next time Git touches it

warning: in the working copy of 'hellogitworld-master/src/Main.groovy', LF will be replaced by CRLF the next time Git touches it

warning: in the working copy of 'hellogitworld-master/src/Square.groovy', LF will be replaced by CRLF the next time Git touches it

warning: in the working copy of 'hellogitworld-master/src/Subtract.groovy', LF will be replaced by CRLF the next time Git touches it

warning: in the working copy of 'hellogitworld-master/src/Sum.groovy', LF will be replaced by CRLF the next time Git touches it

warning: in the working copy of 'hellogitworld-master/src/main/java/com/github/App.java', LF will be replaced by CRLF the next time Git touches it

warning: in the working copy of 'hellogitworld-master/src/test/java/com/github/AppTest.java', LF will be replaced by CRLF the next time Git touches it

ganes@LAPTOP-1U5V6JI4 MINGW64 ~/Desktop/hellogitworld-master (main)

$ git commit -m "Initial commit"

[main (root-commit) b5d8016] Initial commit

16 files changed, 164 insertions(+)

create mode 100644 hellogitworld-master/.gitattributes

create mode 100644 hellogitworld-master/.gitignore

create mode 100644 hellogitworld-master/.travis.yml

create mode 100644 hellogitworld-master/README.txt

create mode 100644 hellogitworld-master/build.gradle

create mode 100644 hellogitworld-master/fix.txt

create mode 100644 hellogitworld-master/pom.xml

create mode 100644 hellogitworld-master/resources/labels.properties

create mode 100644 hellogitworld-master/runme.sh

create mode 100644 hellogitworld-master/src/Division.groovy

create mode 100644 hellogitworld-master/src/Main.groovy

create mode 100644 hellogitworld-master/src/Square.groovy

create mode 100644 hellogitworld-master/src/Subtract.groovy

create mode 100644 hellogitworld-master/src/Sum.groovy

create mode 100644 hellogitworld-master/src/main/java/com/github/App.java

create mode 100644 hellogitworld-master/src/test/java/com/github/AppTest.java

ganes@LAPTOP-1U5V6JI4 MINGW64 ~/Desktop/hellogitworld-master (main)

$ git push -u origin master

error: src refspec master does not match any

error: failed to push some refs to 'https://github.com/LunaticSculptor/test.git'

ganes@LAPTOP-1U5V6JI4 MINGW64 ~/Desktop/hellogitworld-master (main)

$ git push -u origin main

info: please complete authentication in your browser...

To https://github.com/LunaticSculptor/test.git

! [rejected] main -> main (fetch first)

error: failed to push some refs to 'https://github.com/LunaticSculptor/test.git'

hint: Updates were rejected because the remote contains work that you do not

hint: have locally. This is usually caused by another repository pushing to

hint: the same ref. If you want to integrate the remote changes, use

hint: 'git pull' before pushing again.

hint: See the 'Note about fast-forwards' in 'git push --help' for details.

ganes@LAPTOP-1U5V6JI4 MINGW64 ~/Desktop/hellogitworld-master (main)

$ git push -u origin main

To https://github.com/LunaticSculptor/test.git

! [rejected] main -> main (fetch first)

error: failed to push some refs to 'https://github.com/LunaticSculptor/test.git'

hint: Updates were rejected because the remote contains work that you do not

hint: have locally. This is usually caused by another repository pushing to

hint: the same ref. If you want to integrate the remote changes, use

hint: 'git pull' before pushing again.

hint: See the 'Note about fast-forwards' in 'git push --help' for details.

ganes@LAPTOP-1U5V6JI4 MINGW64 ~/Desktop/hellogitworld-master (main)

$ ssh-keygen -t ed25519 -C rachitbodhare@gmail.com

Generating public/private ed25519 key pair.

Enter file in which to save the key (/c/Users/ganes/.ssh/id\_ed25519):

/c/Users/ganes/.ssh/id\_ed25519 already exists.

Overwrite (y/n)? y

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in /c/Users/ganes/.ssh/id\_ed25519

Your public key has been saved in /c/Users/ganes/.ssh/id\_ed25519.pub

The key fingerprint is:

SHA256:VFinoHxpmjjBJIN9kQryNOeIq4FyX1XrRNEWm6DQTt4 rachitbodhare@gmail.com

The key's randomart image is:

+--[ED25519 256]--+

| oo oo...o=oo. |

|+ +=+. o+=o+oo |

|.= Bo o+\*+.oo |

|. + .o \*+ E |

|.. o o.So |

|= . .. . |

|oo . . |

|. . |

| |

+----[SHA256]-----+

ganes@LAPTOP-1U5V6JI4 MINGW64 ~/Desktop/hellogitworld-master (main)

$ eval "$(ssh-agent -s)"

Agent pid 843

ganes@LAPTOP-1U5V6JI4 MINGW64 ~/Desktop/hellogitworld-master (main)

$ ssh-add ~/.ssh/id\_ed25519

Enter passphrase for /c/Users/ganes/.ssh/id\_ed25519:

Identity added: /c/Users/ganes/.ssh/id\_ed25519 (rachitbodhare@gmail.com)

ganes@LAPTOP-1U5V6JI4 MINGW64 ~/Desktop/hellogitworld-master (main)

$ clip < ~/.ssh/id\_ed25519.pub

ganes@LAPTOP-1U5V6JI4 MINGW64 ~/Desktop/hellogitworld-master (main)

$ ssh -T git@github.com

The authenticity of host 'github.com (20.207.73.82)' can't be established.

ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU.

This key is not known by any other names.

Are you sure you want to continue connecting (yes/no/[fingerprint])?

ganes@LAPTOP-1U5V6JI4 MINGW64 ~/Desktop/hellogitworld-master (main)

$ ssh -T git@github.com

The authenticity of host 'github.com (20.207.73.82)' can't be established.

ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU.

This key is not known by any other names.

Are you sure you want to continue connecting (yes/no/[fingerprint])? yes

Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.

Hi LunaticSculptor! You've successfully authenticated, but GitHub does not provide shell access.

ganes@LAPTOP-1U5V6JI4 MINGW64 ~/Desktop/hellogitworld-master (main)

$ ssh -T git@github.com

Hi LunaticSculptor! You've successfully authenticated, but GitHub does not provide shell access.

ganes@LAPTOP-1U5V6JI4 MINGW64 ~/Desktop/hellogitworld-master (main)

$ cd C:/Users/ganes/Desktop/hellogitworld-master

ganes@LAPTOP-1U5V6JI4 MINGW64 ~/Desktop/hellogitworld-master (main)

$ git remote set-url origin git@github.com:LunaticSculptor/test.git

ganes@LAPTOP-1U5V6JI4 MINGW64 ~/Desktop/hellogitworld-master (main)

$ git init

Reinitialized existing Git repository in C:/Users/ganes/Desktop/hellogitworld-master/.git/

ganes@LAPTOP-1U5V6JI4 MINGW64 ~/Desktop/hellogitworld-master (main)

$ git add .

ganes@LAPTOP-1U5V6JI4 MINGW64 ~/Desktop/hellogitworld-master (main)

$ git commit -m "Initial commit"

On branch main

nothing to commit, working tree clean

ganes@LAPTOP-1U5V6JI4 MINGW64 ~/Desktop/hellogitworld-master (main)

$ git push -u origin main

To github.com:LunaticSculptor/test.git

! [rejected] main -> main (fetch first)

error: failed to push some refs to 'github.com:LunaticSculptor/test.git'

hint: Updates were rejected because the remote contains work that you do not

hint: have locally. This is usually caused by another repository pushing to

hint: the same ref. If you want to integrate the remote changes, use

hint: 'git pull' before pushing again.

hint: See the 'Note about fast-forwards' in 'git push --help' for details.

ganes@LAPTOP-1U5V6JI4 MINGW64 ~/Desktop/hellogitworld-master (main)

$ ssh -T git@github.com

Hi LunaticSculptor! You've successfully authenticated, but GitHub does not provide shell access.

ganes@LAPTOP-1U5V6JI4 MINGW64 ~/Desktop/hellogitworld-master (main)

$ ^C

ganes@LAPTOP-1U5V6JI4 MINGW64 ~/Desktop/hellogitworld-master (main)

$ git pull origin main --rebase

remote: Enumerating objects: 3, done.

remote: Counting objects: 100% (3/3), done.

remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0

Unpacking objects: 100% (3/3), 851 bytes | 60.00 KiB/s, done.

From github.com:LunaticSculptor/test

\* branch main -> FETCH\_HEAD

\* [new branch] main -> origin/main

Successfully rebased and updated refs/heads/main.

ganes@LAPTOP-1U5V6JI4 MINGW64 ~/Desktop/hellogitworld-master (main)

$ git push -u origin main

Enumerating objects: 30, done.

Counting objects: 100% (30/30), done.

Delta compression using up to 8 threads

Compressing objects: 100% (16/16), done.

Writing objects: 100% (29/29), 3.43 KiB | 292.00 KiB/s, done.

Total 29 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)

To github.com:LunaticSculptor/test.git

971a7e3..a1efd4d main -> main

branch 'main' set up to track 'origin/main'.

ganes@LAPTOP-1U5V6JI4 MINGW64 ~/Desktop/hellogitworld-master (main)

$ git checkout -b feature-branch

Switched to a new branch 'feature-branch'

ganes@LAPTOP-1U5V6JI4 MINGW64 ~/Desktop/hellogitworld-master (feature-branch)

$ git branch

\* feature-branch

main

ganes@LAPTOP-1U5V6JI4 MINGW64 ~/Desktop/hellogitworld-master (feature-branch)

$ git checkout main

Switched to branch 'main'

Your branch is up to date with 'origin/main'.

ganes@LAPTOP-1U5V6JI4 MINGW64 ~/Desktop/hellogitworld-master (main)

$ git branch -d feature-branch

Deleted branch feature-branch (was a1efd4d).

ganes@LAPTOP-1U5V6JI4 MINGW64 ~/Desktop/hellogitworld-master (main)

$ ^C

ganes@LAPTOP-1U5V6JI4 MINGW64 ~/Desktop/hellogitworld-master (main)

$ git checkout -b conflict-branch

Switched to a new branch 'conflict-branch'

ganes@LAPTOP-1U5V6JI4 MINGW64 ~/Desktop/hellogitworld-master (conflict-branch)

$ git checkout main

Switched to branch 'main'

Your branch is up to date with 'origin/main'.

ganes@LAPTOP-1U5V6JI4 MINGW64 ~/Desktop/hellogitworld-master (main)

$ git checkout -b feature1

Switched to a new branch 'feature1'

ganes@LAPTOP-1U5V6JI4 MINGW64 ~/Desktop/hellogitworld-master (feature1)

$ git add C:/Users/ganes/Desktop/hellogitworld-master/hello.html

ganes@LAPTOP-1U5V6JI4 MINGW64 ~/Desktop/hellogitworld-master (feature1)

$ git commit -m "Made changes in feature1 branch"

[feature1 d70f067] Made changes in feature1 branch

1 file changed, 3 insertions(+)

create mode 100644 hello.html

ganes@LAPTOP-1U5V6JI4 MINGW64 ~/Desktop/hellogitworld-master (feature1)

$ git checkout main

Switched to branch 'main'

Your branch is up to date with 'origin/main'.

ganes@LAPTOP-1U5V6JI4 MINGW64 ~/Desktop/hellogitworld-master (main)

$ git merge feature1

Updating a1efd4d..d70f067

Fast-forward

hello.html | 3 +++

1 file changed, 3 insertions(+)

create mode 100644 hello.html

ganes@LAPTOP-1U5V6JI4 MINGW64 ~/Desktop/hellogitworld-master (main)

$ git checkout conflict-branch

Switched to branch 'conflict-branch'

ganes@LAPTOP-1U5V6JI4 MINGW64 ~/Desktop/hellogitworld-master (conflict-branch)

$ git add C:/Users/ganes/Desktop/hellogitworld-master/hello.html

ganes@LAPTOP-1U5V6JI4 MINGW64 ~/Desktop/hellogitworld-master (conflict-branch)

$ git commit -m "Made conflicting changes in conflict-branch"

[conflict-branch 7459eec] Made conflicting changes in conflict-branch

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 hello.html

ganes@LAPTOP-1U5V6JI4 MINGW64 ~/Desktop/hellogitworld-master (conflict-branch)

$ git push --all origin

Enumerating objects: 7, done.

Counting objects: 100% (7/7), done.

Delta compression using up to 8 threads

Compressing objects: 100% (4/4), done.

Writing objects: 100% (6/6), 622 bytes | 207.00 KiB/s, done.

Total 6 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)

remote: Resolving deltas: 100% (1/1), done.

To github.com:LunaticSculptor/test.git

a1efd4d..d70f067 main -> main

\* [new branch] conflict-branch -> conflict-branch

\* [new branch] feature1 -> feature1

ganes@LAPTOP-1U5V6JI4 MINGW64 ~/Desktop/hellogitworld-master (conflict-branch)

$ git checkout conflict-branch

Already on 'conflict-branch'

ganes@LAPTOP-1U5V6JI4 MINGW64 ~/Desktop/hellogitworld-master (conflict-branch)

$ git merge main

Auto-merging hello.html

Merge made by the 'ort' strategy.

hello.html | 3 +++

1 file changed, 3 insertions(+)

ganes@LAPTOP-1U5V6JI4 MINGW64 ~/Desktop/hellogitworld-master (conflict-branch)

$ git commit -m "Resolved merge conflicts between main and conflict-branch"

On branch conflict-branch

Untracked files:

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

hello.txt

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

ganes@LAPTOP-1U5V6JI4 MINGW64 ~/Desktop/hellogitworld-master (conflict-branch)

$ git add C:/Users/ganes/Desktop/hellogitworld-master/hello.html

ganes@LAPTOP-1U5V6JI4 MINGW64 ~/Desktop/hellogitworld-master (conflict-branch)

$ git commit -m "Resolved merge conflicts between main and conflict-branch"

On branch conflict-branch

Untracked files:

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

hello.txt

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

ganes@LAPTOP-1U5V6JI4 MINGW64 ~/Desktop/hellogitworld-master (conflict-branch)

$ git add /c/Users/ganes/Desktop/hellogitworld-master/hello.txt

ganes@LAPTOP-1U5V6JI4 MINGW64 ~/Desktop/hellogitworld-master (conflict-branch)

$ git commit -m "Resolved merge conflicts between main and conflict-branch"

[conflict-branch f6336e7] Resolved merge conflicts between main and conflict-branch

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 hello.txt

ganes@LAPTOP-1U5V6JI4 MINGW64 ~/Desktop/hellogitworld-master (conflict-branch)

$ git remote add origin https://github.com/LunaticSculptor/repository.git

error: remote origin already exists.

ganes@LAPTOP-1U5V6JI4 MINGW64 ~/Desktop/hellogitworld-master (conflict-branch)

$ git remote -v

origin git@github.com:LunaticSculptor/test.git (fetch)

origin git@github.com:LunaticSculptor/test.git (push)

ganes@LAPTOP-1U5V6JI4 MINGW64 ~/Desktop/hellogitworld-master (conflict-branch)

$ git clone https://github.com/LunaticSculptor/hellogitworldforked.git

cd hellogitworldforked

Cloning into 'hellogitworldforked'...

remote: Enumerating objects: 115, done.

remote: Counting objects: 100% (29/29), done.

remote: Compressing objects: 100% (12/12), done.

remote: Total 115 (delta 19), reused 17 (delta 17), pack-reused 86

Receiving objects: 100% (115/115), 12.31 KiB | 900.00 KiB/s, done.

Resolving deltas: 100% (35/35), done.

ganes@LAPTOP-1U5V6JI4 MINGW64 ~/Desktop/hellogitworld-master/hellogitworldforked (master)

$ git checkout -b new-feature-branch

Switched to a new branch 'new-feature-branch'

ganes@LAPTOP-1U5V6JI4 MINGW64 ~/Desktop/hellogitworld-master/hellogitworldforked (new-feature-branch)

$ git add .

git commit -m "Added new feature"

On branch new-feature-branch

nothing to commit, working tree clean

ganes@LAPTOP-1U5V6JI4 MINGW64 ~/Desktop/hellogitworld-master/hellogitworldforked (new-feature-branch)

$ git push origin new-feature-branch

Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)

remote:

remote: Create a pull request for 'new-feature-branch' on GitHub by visiting:

remote: https://github.com/LunaticSculptor/hellogitworldforked/pull/new/new-feature-branch

remote:

To https://github.com/LunaticSculptor/hellogitworldforked.git

\* [new branch] new-feature-branch -> new-feature-branch

ganes@LAPTOP-1U5V6JI4 MINGW64 ~/Desktop/hellogitworld-master/hellogitworldforked (new-feature-branch)

$ git config --global alias.lol "log --oneline"

ganes@LAPTOP-1U5V6JI4 MINGW64 ~/Desktop/hellogitworld-master/hellogitworldforked (new-feature-branch)

$ git lol

ef7bebf (HEAD -> new-feature-branch, origin/new-feature-branch, origin/master, origin/HEAD, master) Fix groupId after package refactor

ebbbf77 Update package name, directory

45a30ea Update package name, directory

9805760 Fix YAML name-value pair missing space

ef7bebf (HEAD -> new-feature-branch, origin/new-feature-branch, origin/master, origin/HEAD, master) Fix groupId after package refactor

ebbbf77 Update package name, directory

45a30ea Update package name, directory

9805760 Fix YAML name-value pair missing space

ef7bebf (HEAD -> new-feature-branch, origin/new-feature-branch, origin/master, origin/HEAD, master) Fix groupId after package refactor

ebbbf77 Update package name, directory

45a30ea Update package name, directory

9805760 Fix YAML name-value pair missing space

435ffce Merge pull request #35 from githubtraining/travis-yml-docker

7c5ee64 Add special option flag for Travis Docker use case

a13dba1 Even I can change the readme file

8d80474 Travis CI

78f4771 Fixes #1

c594793 Added gradle build file

d2280d0 Adding maven build script

bf7a9a5 .gitattributes to make this play nicely onWindows

e83ece7 Adding log files and build output to ignore

8d2636d Added text about image diff example

2a52e96 Merge branches 'feature\_division\_polished' and 'feature\_subtraction\_polished'

57d0bf6 Removing subtraction from the division line of code

cb2d322 Added subtract feature

818f033 Added call and import for subtraction

369adba Used static imports to simpify the math calls

2956e1d Removed unused Square class import

333fd9b Added sum function and test call

f5ed019 Added shell script to drive the execution of the app

9cdb160 Ignored \*.class files

dc64fe4 Refactored squaring into a separate class

32c2737 Added sqaring method

755fd57 Addition of the README and basic Groovy source samples.

...skipping...

435ffce Merge pull request #35 from githubtraining/travis-yml-docker

7c5ee64 Add special option flag for Travis Docker use case

a13dba1 Even I can change the readme file

8d80474 Travis CI

78f4771 Fixes #1

c594793 Added gradle build file

d2280d0 Adding maven build script

bf7a9a5 .gitattributes to make this play nicely onWindows

e83ece7 Adding log files and build output to ignore

8d2636d Added text about image diff example

2a52e96 Merge branches 'feature\_division\_polished' and 'feature\_subtraction\_polished'

57d0bf6 Removing subtraction from the division line of code

cb2d322 Added subtract feature

818f033 Added call and import for subtraction

369adba Used static imports to simpify the math calls

2956e1d Removed unused Square class import

333fd9b Added sum function and test call

f5ed019 Added shell script to drive the execution of the app

9cdb160 Ignored \*.class files

dc64fe4 Refactored squaring into a separate class

32c2737 Added sqaring method

755fd57 Addition of the README and basic Groovy source samples.

(END)