VERSION CONTROL

SOFTWARE ENGINEERING

ROBOTICS CORNER

CONTENTS

- Version Control
- Git
- GitHub

ROBOTICS CORNER

VERSION CONTROL

- Version control, also known as source code management (SCM) or revision control, is a system that records changes to files over time. It allows multiple developers to work on the same project simultaneously while keeping track of every change made to the source code or any other type of file. Version control systems (VCS) maintain a history of changes, enabling users to compare different versions, revert to previous states, and collaborate effectively.

ROBOTICS CORNER

KEY ASPECTS OF VERSION CONTROL

- History Tracking
- Branching and Merging
- Conflict Resolution
- Collaboration
- Reproducibility and Auditing

ROBOTICS CORNER

GIT

- A distributed version control system widely used for open-source and commercial projects, i.e. a tool for version control

ROBOTICS CORNER

GIT

- Git solves the problem of having multiple developers and teams having to share the code and resources required for the project.
- By uploading the data into an online repository like a database, this database like online location will also contain some metadata about who and when did the data was uploaded, and what changes were made.
- There exists a local repo and a remote repo.
- When multiple users try to upload at the same time, git tells you that a newer version was uploaded.

ROBOTICS CORNER

INSTALL GIT LINUX

- sudo apt-get install git-all

ROBOTICS CORNER

INSTALL GIT WINDOWS

- https://git-scm.com/download/win

ROBOTICS CORNER

GIT

- Amazon AWS CodeCommit
- Beanstalk
- Bitbucket
- Codebase
- GitLab
- SourceForge
- GitHub
- Azure DevOps Server
- Perforce

ROBOTICS CORNER

GITHUB

- GitHub is a web-based platform built around Git, a distributed version control system (VCS). It provides a range of features and functionalities for hosting, sharing, and collaborating on software development projects. GitHub is widely used by individuals, teams, and organizations for managing code repositories, tracking issues, and facilitating collaboration among developers.

ROBOTICS CORNER

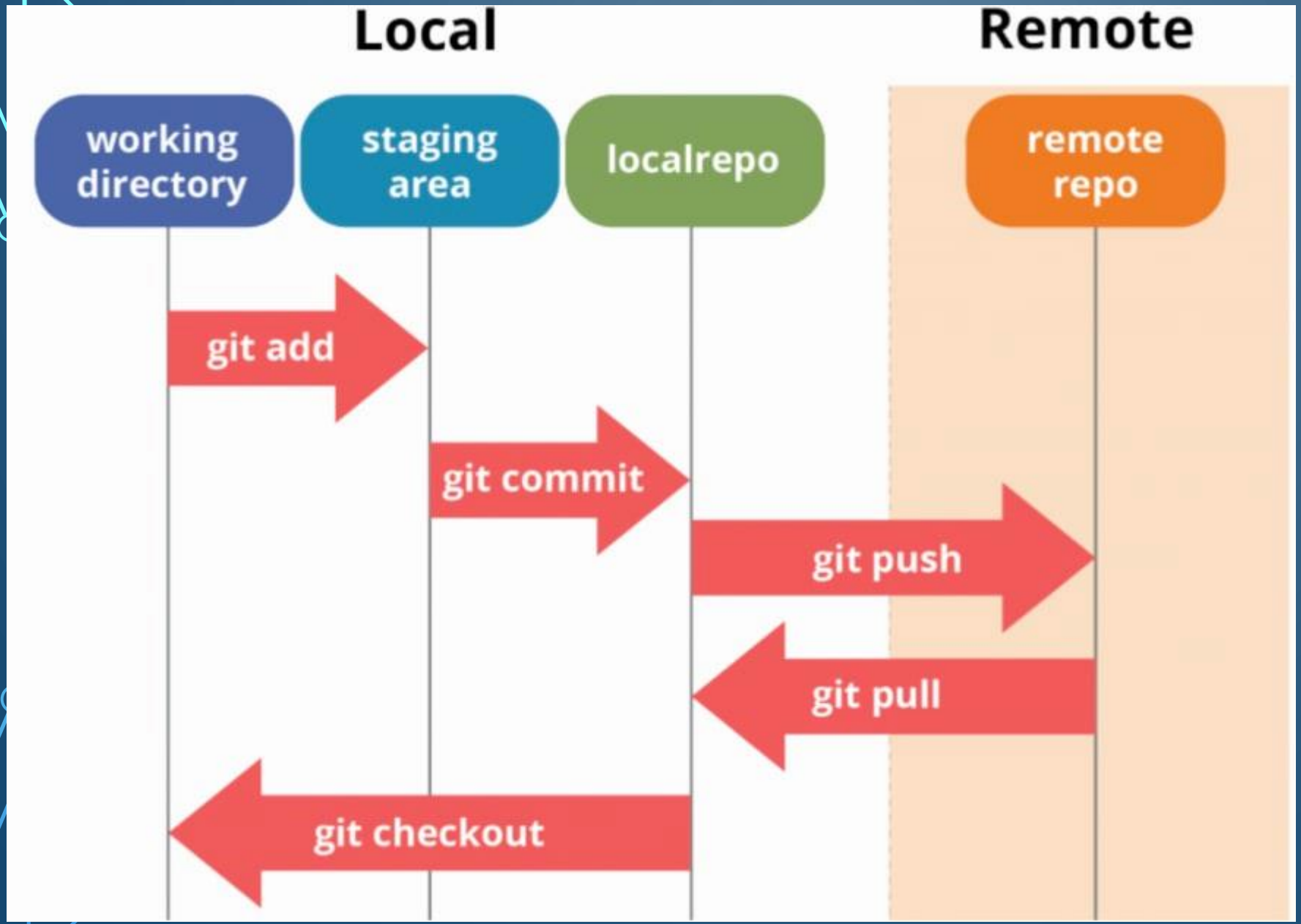
STEPS

- Register Account on github.com
- Create your own Repository on github
- Use git clone to download a copy of the Repo.
- For the first time using git you need to configure your user.email and user.name using:
  - git config –global user.email “email@domain.com”
  - git config –global user.name “name”

ROBOTICS CORNER

- Use git add . (to add files to a memory buffer that will be uploaded to the remote server)
- Use git commit –m “message” (specify your own commit message)
- Go to account – settings – go down to developer settings – personal access tokens – press on tokens(classic) – generate new token – generate new token(classic) – type: new token – set an expiration date – select the repo in the scopes – generate token – copy the generated token
- Using git push –u origin master to upload your files, add the username of the created github account and in the password section add the generated token then press enter

ROBOTICS CORNER



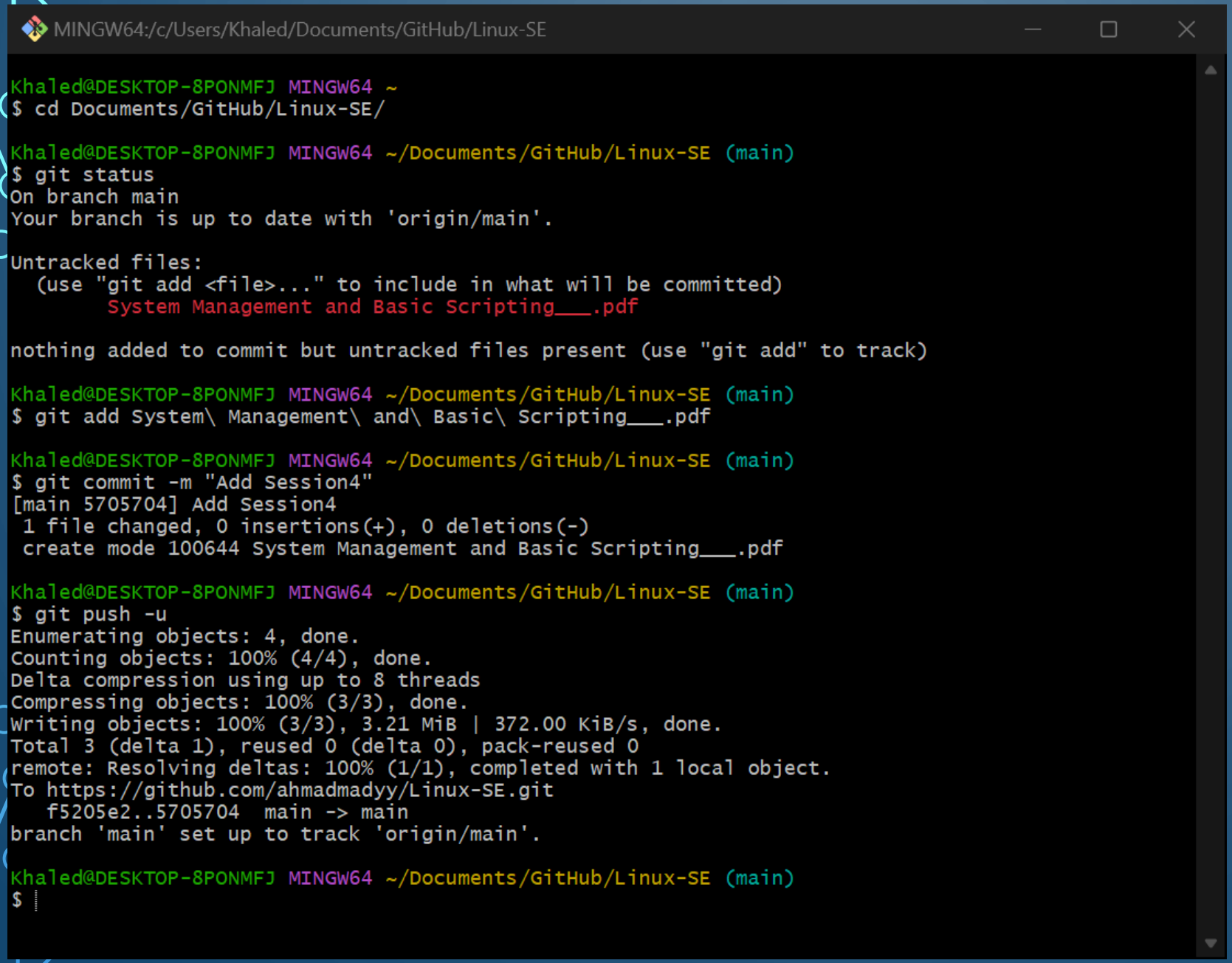
ROBOTICS CORNER

- git status: gets the status of the uploaded data
- git add * → uploads all data in this directory
- Metafiles are stored locally in .git folder and online on the GUI
- To view all commands in git: git –help
- git diff: tells what changes were made on a file

ROBOTICS CORNER

```
ahmedmady@HP:~/Documents/Git_test$ git clone https://github.com/roboticscorner
/Test.git
Cloning into 'Test'...
warning: You appear to have cloned an empty repository.
ahmedmady@HP:~/Documents/Git_test$ ls
Test
ahmedmady@HP:~/Documents/Git_test$ cd Test
ahmedmady@HP:~/Documents/Git_test/Test$ ls
'Introduction to Linux.pptx'
ahmedmady@HP:~/Documents/Git_test/Test$ git add Introduction\ to\ Linux.pptx
ahmedmady@HP:~/Documents/Git_test/Test$ git commit -m "first commit"
[main (root-commit) a8f30cb] first commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 Introduction to Linux.pptx
ahmedmady@HP:~/Documents/Git_test/Test$ git push -u
Username for 'https://github.com': roboticscorner02
Password for 'https://roboticscorner02@github.com':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 3.98 MiB | 389.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/roboticscorner02/Test.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
ahmedmady@HP:~/Documents/Git_test/Test$
```

ROBOTICS CORNER

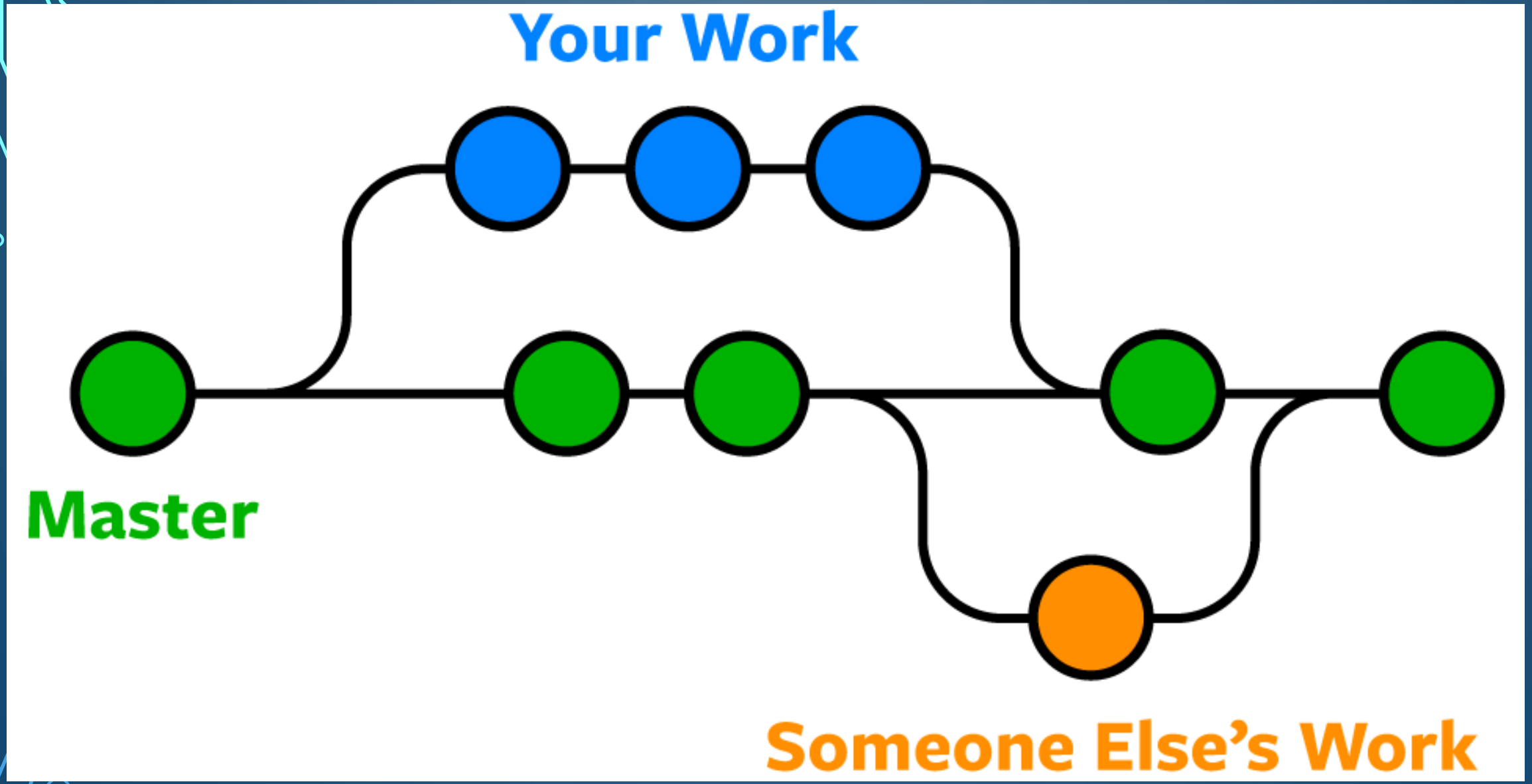


ROBOTICS CORNER

CREATE A NEW BRANCH

- For example if a person wants to test part of the code, but doesn't want to change the existing code, a branch is to be created:
- git branch [BRANCH_NAME]
- To show all branches: git branch –a
- Navigate to the branch: git checkout [BRANCH_NAME]
- Go back to the main directory: git checkout main

ROBOTICS CORNER



ROBOTICS CORNER

BRANCHES

- To delete a branch: git branch –D [BRANCH_NAME]
- To merge branches, you need to be on the branch you want to merge into:
- git merge [BRANCH_NAME] then git push -u
- To delete a merged branch: git branch –d [BRANCH_NAME]
- To ignore files from being uploaded, add a git ignore file and write the files to be ignored or write *.exe

ROBOTICS CORNER

WHAT GUARANTEES PERFECTION

ROBOTICS CORNER

GIT COMMANDS

- https://about.gitlab.com/images/press/git-cheat-sheet.pdf

ROBOTICS CORNER

TASK

- Create a GitHub account and add some files from your laptop!
- Create a file and push it to GitHub
- Go back to the local repo and edit the file, use git status to check that the file was changed then push the edited file again on GitHub.

ROBOTICS CORNER