**Class Git 01. Date;06-12-2023**

**What is git ??** git is a version control tool , it was discovered by linus torvalds in 2005.

**What is git-bash;** it was discovered by linus in 2005 for window operating system .it is application or software on which git is used or write or accessed by command line interface.

Git was created by Linus Torvalds, the founder of the Linux operating system, in 2005.

What is unix style commands???? Unix is a powerful, multiuser, multitasking operating system originally developed in the 1960s and 1970s at Bell Labs, “Unix-style command" typically refers to the command-line syntax and conventions used in Unix-like operating systems. These commands are entered into a command-line interface (CLI) or terminal to interact with the operating system and perform various tasks.

A Unix-like command-line interface (CLI) refers to a text-based interface used in operating systems A text-based interface is a type of user interface that depends on text characters rather than graphical elements to interact with a computer or software application.

**What do we need git??? To keep track of our java code.**

1. to saves time
2. to saves memory
3. to saves being mess up

**What is version??**

Any change in the code creates its new version.

Updating of an app or tool from previous.

1. **What is git??? global information tracker**; it is version control tool.

**Keep tracking history**

**collaboration**

Git is used to tracking changes in the source code, enabling multiple developers to work together on non-linear development

Git is a version control tool, that keep tracks all the changes in our code which we made and store them into repository

John writer ---------- hire the minion to store and manage data---------------- inside the cupboard

Computer hire --------------------git -----to store and manage data --------------- inside repository

**How to communicate with git???**

We communicate with git with commands. These are UNIX-style commands.

Git is primarily designed for version control and managing source code, but it can be used for versioning and tracking changes in any type of text-based file. While it's not optimized for handling non-text files like binary files (e.g., images, videos, executables), Git can still track changes in these files.

**Where to give those commands**

In window we have git bash where we will give these commands.

1. **This is the folder or path that the current Bash session resides in. pwd (Stands for “print working directory”) — Prints out the current directory. ls — Lists the files in the current directory**

Keyword; accustomed --- aaadi , habitual

1. Command ls ---- > list content; hey terminal please tell us what are the files and folder there you are standing at current location.
2. clear --- command is used to remove or clear all dashboard.
3. cd ---- change directory --🡪 in which directory you want to go.
4. cd Desktop --------- >> go to desktop directory and print the names of all the files and folder present there.
5. cd .. ----- this is command to go back to previous folder
6. ~ tilde --- till-da ----- >> is shortcut to go back at home directory.

Now open your eclipse or IntelliJ and find the location of your project.

1. Open the IntelliJ and find the location of your project.
2. G:\04 java\javaReviewClasses

To move from one drive to another drive the command s

--- > cd /G

Cd and then space and then forward slash and then name of drive in which you want move.

This is condition in git make your repository there, where is you code present.

**Make sure make repository out side the src folder.**

**Terminal is where git has been installed.**

**Class 02 git Date: 07-12-2023**

Move in any drive just write command --------- cd /G ------- mean drive name inside forward slash.

How you can create git repository??

Before creating git repository, you should configure it. Suppose there are many students who are using same computer then you have to give separate account. Your account will be separate to use your data and not mess with other’s data.

1. **Why do You need to configure git;**
2. **You need to configure so that because there is repository to enter inside repository there should be user account, if you have user account then only you can change data from that account no one can change delete or remove your data.**

Configuring Git is an essential step in setting up and using the version control system effectively. Here are some reasons why configuring Git is important: Configuring your name and email in Git is crucial for accurately attributing changes to the correct individuals. When you make a commit in Git, it records not just the changes you made but also who made them. This information is embedded in the commit history, and it's valuable for several reasons: **Accountability:** Knowing who made a particular change is essential for accountability. If there are issues or questions about a specific piece of code, you can identify the person responsible and reach out to them for clarification or assistance. ----- accountability mean------- ahtasaab, zimadaari.

1. You need once to configure git in your computer.

**Important note ; when your terminal is in home directory , after then create git account as below.**

**It is common to have one repository for one single project .**

1. To create user name in git this is the command
2. **git config --global user.name “babarali”** ------ 🡪 git config space – double dash space user.name space “username”
3. **git config --global user.email “babaraliengineer32@gmail.com”** ------ 🡪 git config space – double dash space user.email space “email”
4. --------- > git config --global user.name "Your Name"
5. --------- > git config --global user.email "your.email@example.com"
6. **now go to your eclipse or IntelliJ and find the absolute path of your project and move the terminal inside the git to that path where the project of eclipse or IntelliJ is present.**

The global configuration in Git refers to settings that are applied across all of your Git repositories on a specific machine. When you set up Git for the first time on your computer, it's a good practice to configure some global settings, such as your name and email address.

1. To check the global configuration (system-wide configuration), use:

Bash Copy code

---------------------------------git config --global –list-------------------------------------------------------

This will display a list of configuration settings, including user name and email.

1. To check the configuration for a specific repository, navigate to the repository's directory and use:

Bash Copy code

-------------------------------------------------------git config --local –list----------------------------------------------------------------------

This will show the configuration specific to that repository.

**Command Line Interface (CLI):** The terminal is a CLI that provides a way for users to interact with their computer using text commands.

Emulator mean --------- >> naqal krnay wala , paervi krne wala , taqleed krne wala

**Tracking our first project using git after configuration is done.**

1. Navigate the terminal to that location where you want to create your repository where the code is saved of java.
2. To make repository use the command ----------- git init ----------------------------
3. To save your code into git repository use the following command -----------

git add .

git commit -m “message” ------ commit mean --- soapna, amanat ma dena, Azzam

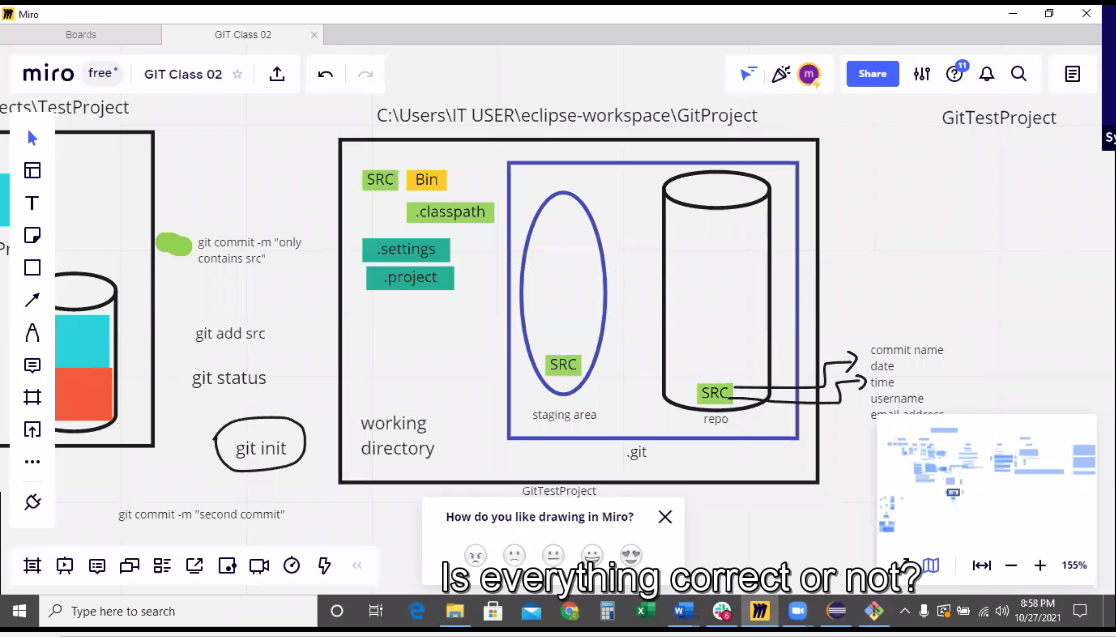
commit mean saving the code into repository

hey git now show us what you have stored inside the repository then there is command

the command is ------- git log -------------

**how repository works lets under stand all the concepts.**

**No lets understand how the git works**

1. Go to intellij and create a new project name it as git/project or any other name.
2. Navigate the terminal to the location where we created the intelli j project. G/java/gitproject inside git-bash.
3. Find the right location where you want to create repository.
4. Use the command -------- git add . --------------- to create a repository.
5. What does this command do. This command add the file into staging area.
6. 

**Lets understand this diagram.**

1. When you give command --------- git inint --------- then
2. a folder with name .git is created inside where java code is present. If you delete the .git folder repository will delete.

Lets understand the architecture of git

Git has three stage of architecture. When a .git folder is created it means internally there three things created.

1. Working directory ---------- the place where our java files are present.
2. Staging area -------------🡪 the place where we add all the files/codes/folder which we want to send the repository------------------- at intially empty staging area
3. Repository ----------🡪 storage which contains all our commits ----------- > intially empty repository….
4. Use the command -------- git add . --------this add all the files in staging area------- to create a repository.
5. What does this command do. This command add the file into staging area.
6. What is -------- git status --------- command. This command only tells the status of working directory and staging area. This tells us that files are not being tracked. The files shown red mean these files are not tracked.
7. When we -------- git add src -------- then it added at staging area
8. To save the file in repository we use the command ------- git commit -m “msge” -------- what is the message should be, the message should be clear what is this file . then
9. what does commit command do ???
10. it gives the name
11. date
12. time
13. username
14. email

When you see the message "changes to be committed" after running **git status**, it means that you have used the **git add** command to move changes from your working directory to the staging area. These changes are now ready to be permanently saved in a commit.

a "commit" refers to a specific snapshot of your project's files at a given point in time. When you make a commit, you are essentially creating a record of the changes you've made to your project. Each commit has a unique identifier, typically a long hexadecimal number called a hash, that distinguishes it from other commits. Each commit also includes a commit message that describes the purpose of the changes.

Change to be committed mean the files are ready to take screen shot in staging area.

If you give command --------- git commit -m “message” ------ then files will be push to repository after screen shot.

**--------- git log ---------- is the command which print all the things present inside the repository.**

**-----------git status ------- command is used to see all the files in working directory and staging area.**

**Lets understand when we change the code inside intelli j then what happened inside git repository.**

**------- untracked mean ------- files are inside working directory.**

**---Changes not staged for commit:--- this mean you have mad changes in your code your file is not in staging area it is still inside working directory .**

**No thing added to commit but there are untracked files----- mean you do not made**

**modified: src/videoLectures/ArratTask03.java ---- this line mean you have made changes in this file.**

**Modified; bin/vaer/aer/ ---- this file is binary file this is independent on git changes.it changes with changing of code .**

**nothing to commit, working tree clean ----- this means working directory is empty there is nothing.**

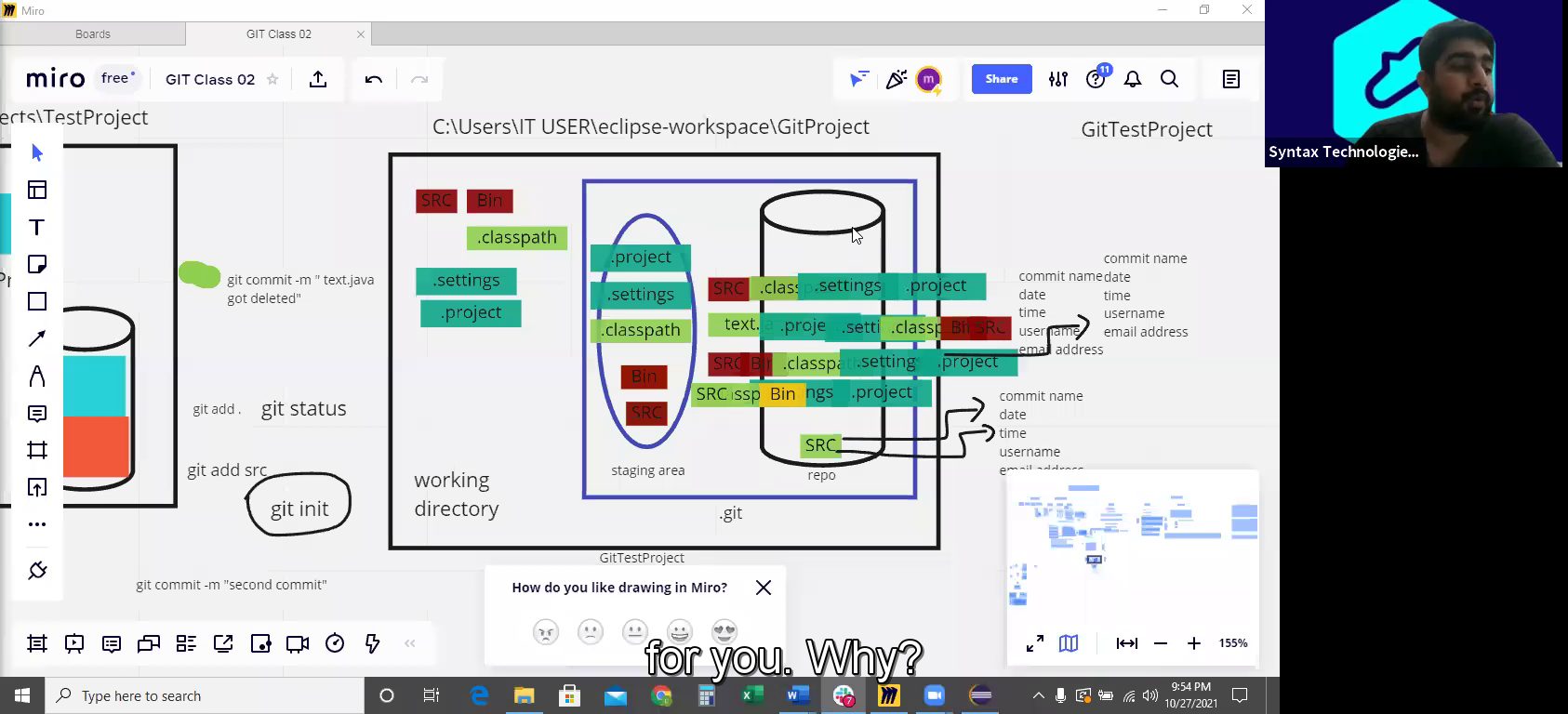
no changes added to commit ---- this line means changed file is not added in repository first add it in staging area and then add inside repository.

he message "nothing added to commit" in Git typically appears when you run the **git commit** command, but Git doesn't find any changes to commit in your working directory. This message indicates that there are no new or modified files that Git recognizes as candidates for a commit.

Control + C ---------- >> this command is used to cancel current operation which you have done in git.

**Lets delete text file from the intellij and from the working directory; now**

1. **now check the status ------- git status -------- it shows deleted file option**
2. **then you have to give command ---- git add . ----- it will add new changes , it will also delete the text file from staging area,**
3. **when you give command ----- git commit -m --- it will create new updated version inside the repository it will not delete the previous version from repository.**



**Class number 03 Date; 08-12-2023**

Why we don’t create repository inside the src or .idea or other folder ???

Because if we create the git repository inside any folder like src or idea then git will not be tracked others files like text files, pom.xml files or other files.

Create a new project in intellij and create a repository inside the project.

**What is git ignore file??**

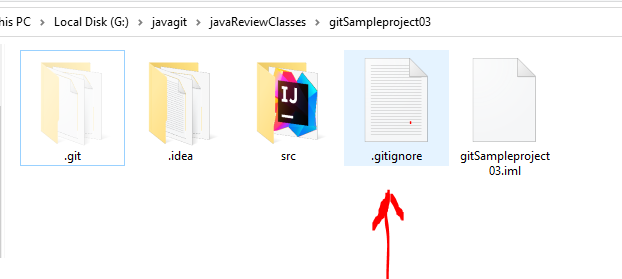
git ignore file ------ > the files and folder inside the working directory which are not important and that we don’t need to save them inside the repository and they don’t have any purpose to save them inside the repo, ignore all of those files and folder which are not important.

1. out folder -------- this is binary folder we don’t need to add this inside the repository
2. --- .idea folder this folder also don’t need to add in repository.

For above purpose we have a git ignore file.

**how create a git ignore file, what is the command to create git ignore file.????**

**-------------------- touch .gitignore --------------- this is the command and a file with the name of .gitignore is created inside the working directory.**



**How you will add file inside the git ignore file**

1. Open the git ignore file present inside the working directory
2. Then write the names of unnecessary files inside the git ignore file and then save the git ignore file and then close the git ignore file.
3. ------------write the name inside git ignore files as ----------/src
4. -----------------------------/out
5. -------------------------------------/.idea

After doing of above steps you are going to add. git ignore file inside the repository by command ------- git add . -----------------

What does it mean un-tracked -------------- >> it means files not stored inside the repository. The red color of files show that the files are inside the working directory.

The green color of files show that the files are inside the staging area. And they need to commit into repository.

----- to see all the files inside the staging area there is a special command ---------

----------------------------- git ls-files------------------------------this command shows all files inside the staging area.

-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Above these lines the commands of git have to be remembered, and now next commands you don’t need to remember

**Class 3 git Date; 09-12-2023**

**How remove the file if it accidently added in staging area.**

**Scenario;**

1. **Suppose you accidently added a file inside the staging area , and now you want to remove this file from staging are due to following reason**
2. **The file is not ready yet , there are errors inside yet. I don’t want to share this file with my colleagues**
3. **This is un-necessary file**
4. **How now I will remove this file**
5. **To remove this file there is a command in that is -------- git rm --cached filename-----------------**
6. **Example = git rm –cached .idea**
7. **Shortcut if you write a command to remove file ---- git rm –cached initial file name + tab ------ this shortcut auto will add the name of file which you want to remove.**
8. **to check all the files in staging area command is-------git ls-files---------**

(HEAD -> master) ------ > this head mean pointer/indicator/cursor. This head indicates that tells that give indication that the commit which you have inside the working directory .This is the commit that represents the current state of your working directory. In simple words this tells the current status of your working directory. This is the commit on which currently worked has been done .

-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

There is another command ----------git diff------------ this command tell the difference in code or files in ‘’staging area’’ and “working directory”. This mean when you change code inside intellij then src file inside the repository tells that modification has been done in src file which is inside the src file which is present inside the working directory. Then --------git diff-------- tell us the difference in file present in working directory and inside the staging area.

Lets take example . suppose you have src folder inside repository , you make changes in any of file or code inside intellij and save and then you check the ------ git diff------ , this gives modified; scr/java/project. This mean it captured the changes.

1. To solve this modification you can remove changes what you made inside the intellij j
2. Or first add the file inside staging area, command -------git add . --------------------- and then inside the repository -------command for that -------------git commit -m “message”.

Here is another command if you want to see the status of file before commit the file into repo and after add into staging area.

The status of file between staging area and repo, use the command.

------------------git diff --staged ---------------

**Important command if you accidently delete src folder. Then you can recover it by**

**--------------------git restore src-----------------------foldername**.

-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**Class -03 present inside java class 08 Date;12-10-2023**

**What is github ?????**

**Central repository on internet to have backup of our code Is called github.**

**1. Suppose your hard disk burnt out in your computer , then your local repository data will also loose. In this case we have a central repository on internet. Just like cloud storage for iphone , ------google drive ---- for google backup.**

**Why we should use git hub ? what is the importance of that ???**

**Scenario;**

**Suppose when there was not git hub how you will share your code with other friends who are working at same project inside a company. Then you will use emails, WhatsApp, and other app , it will be very difficult to communicate every time with every member, but if there is github you simply make changes in your code and add comments what you have done, and other team member will watch your code what you have done , and he will start working on his further task or changes on the code.**

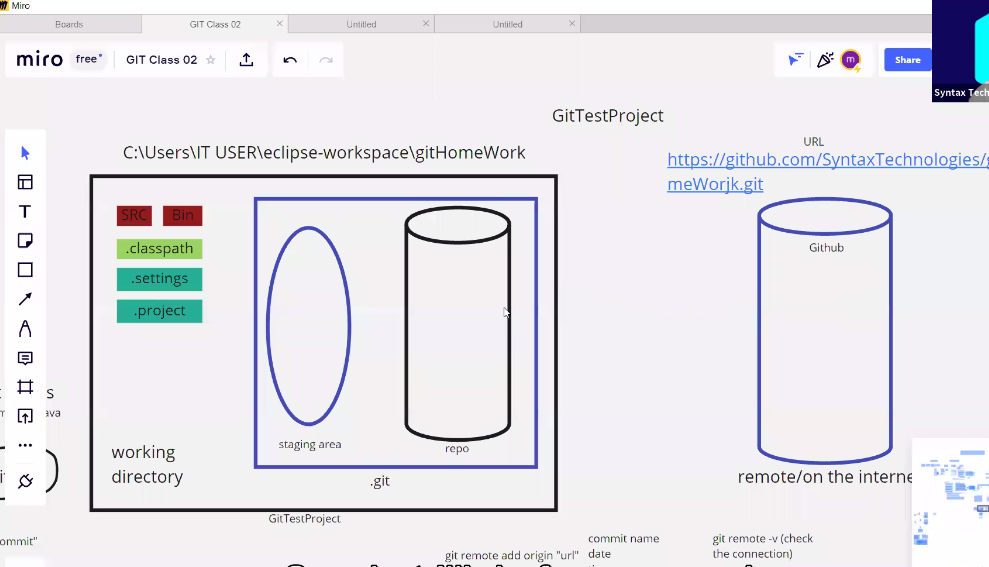
**That’s why git is easy to use.**

**What other task you can do by using github???**

**2. work with collaboration. Mean with team .**

enters the command mode, w is for "write" (save) and q is **for "quit**". You may need to hit escape before :wq to exit the insert mode

important point. Github don not have staging area, it only have repo are.



Now the created repository is git home work on github account. Now how we will make connection with git hub which is present on internet for this we need the URL of that repository.

And go to git bash there use all the commands.

**How you will send code from local to central repository / from git to github.??**

1. first of all create a repository on github.

2. now you need to make a connection between git local and github central.

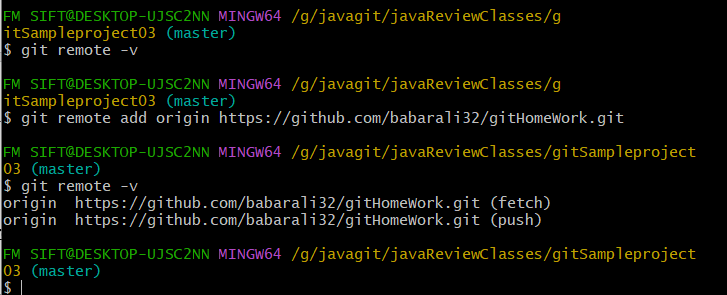
3. to make a connection with git and github repository there is a command which you have put in git bash terminal

---------------command is ------------git remote add origin url of github repository-------------------------

2. to check that the connection is created between git and github there is command

--------------------------------git remote -v-------------------------

There is a command if you want to remove the connection with github repository and you want make connection with other git hub repository .--------------git remote origin-------------------------------



Now in above screen shot there are two options,

1. fetch ---------- do you want to fetch data from central repository

2.push --------- do you push the data from git to the internet central repository

Now we want to push the code because we can not fetch write now because it is empty github central repository.

So now there is another command we have to use, to tell the github to/inside which branch we are pushing the code.

Command is ---------------git branch -M main---------------------------just use once in while creating repository on github.

Now we want to push the code for that there is command

;;;--------------------------- git push -u origin main------------------------------------------------------

(HEAD -> main, origin/main) ---- >> this mean our local and remote repository are synchronized are at one stage.

**Class 03 date; 12-10-2023**

شمار کرنا-----enumerating files

What does this command do?????????

----------------------------------------------git push -u origin main-----------------

This command push all the code/files/folder present in local repository area (not adding staging area code) just only just the code inside repo and push into github repository .

---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**What is git clone????????**

Clone mean making a copy. a plant or animal that has the same genetics as the original from which it was produced

What is difference in download and cloning git.??????

Download from github mean it download only one latest or updated file from the github repository, and not download all the files.

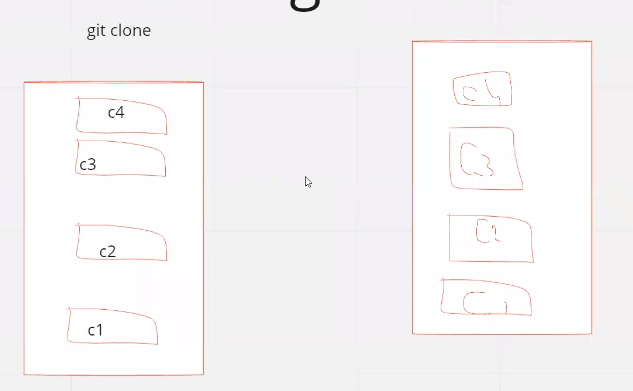
------------clone mean it download the complete repository from the central repository. ----------

Why clone is important in git clone ?????????/

Scenario; suppose we all working on a project and we need to push and download code into central repository.

There is condition in git repository you only can push the files or code when central repository is empty or when central and local repository are at same level. Or both have same history as we read this already.







This is example of cloning and downloading.

**What is method to clone git???????/**

1. create a folder on desktop in you pc

2. now open the git bash inside from this folder/ or navigate your terminal inside the folder

3. go to the link of that repository whom you want to clone. Go on github and copy the link of repository.

4. open your git bash terminal and give a command --------------git clone URL/or link of repository-------------------

5. now navigate your terminal to the folder which is cloned at desktop.

6.now open intellij j go to file select open then navigate to the cloned folder at desktop.

7. now create a class in intellij.

8.now add this class in clone repo by ---------git add . ----------------

9. this is in staging area now commit this file -----------git commit -m “clone file”-------------

10. now push this file from local to central repo -------------------git push-------------

11. now you can check on git hub the file is present there.

Now there are two more important commands.

1.---------------git pull--------------

2. -------------------------git fetch----------this command tells the status of remote repository . is there any update which we need to download.

A **cheat sheet** (also *cheatsheet*) or **crib sheet** is a concise set of notes used for quick reference

**Class 04 Date; 11-12-2023**

**What is git stash???? ----------------------------stash mean-------------- >> hide, conceal, store.**

The git stash command takes your uncommitted changes (both staged and upstaged), saves them away for later use, and then reverts them from your working code.  It's a convenient way to temporarily save changes locally without committing them to the repository.

Scenario; is that when you do not want to add changes inside repo, you want first finalized your code and then you want to add this code into repo. The redundant code or unnecessary code you can save inside stash.

Revert mean ------------------ wapis lotana--------wapis krna-------wapis lotna------------------

Git stash ;

First of all create a java project in intellij and create 2 classes in java project and name them as car build and tester class

When the class is created then go to the folder where project is created inside the computer.

Now open git bash and create repository inside project folder.

Then create a git ignore file and remove .idea and out folder by putting inside the git ignore file.

Now use ----git add .---------command to add all files inside the staging area.

Then use command to add file inside the repository-----------git commit -m “message”-------------

Now change your code inside the intellij j , and now use the command --------git stash -m “white car”-----

**How does git stash command work?? Pictorial understanding of diagram.**

Git stash command is-----------------------------------git stash -m “message” --------------------------

1. when you made some temporary changes inside working directory (mean inside intellij code) an you do not want to store these temporary changes inside the repository.

2. then you use the command---------git stash -m “message” --------------- this command save the code inside stash directory or stash storage area.

3. and this stash command further what also do that , it changes working directory reverted to the state of last commit in the local repo. Simple mean last commit and working directory are at same level.

----------------------------------------------------------------------------------------

----------------------------------------------------------------------------------------

**Lets understand simple working of stash command.**

1. suppose you write a cod inside intellij in a class --------- sum of two number is your code.

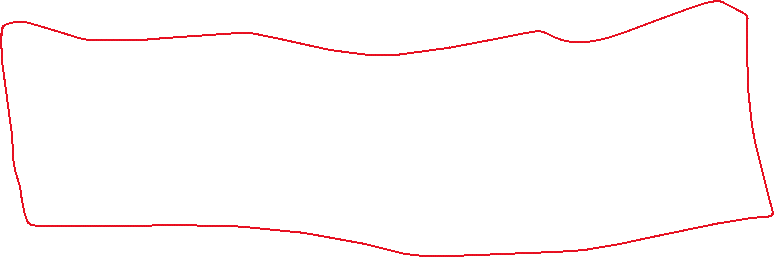
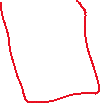
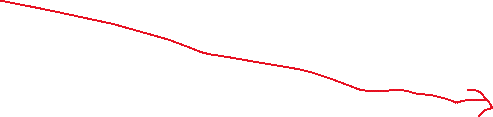
2. you add this code inside local repository by-- git add .----command and ---git commit ---command.

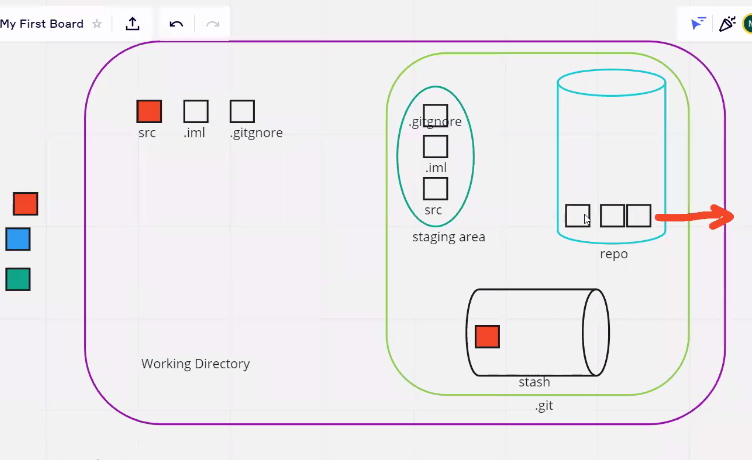
3.now go to intellij add further line of code ---- subtraction of two numbers.

4.now go to git bash terminal and don’t add these changes inside the repository. But you will send these changes subtraction of code inside another storage area which is called stash using the commands--------git stash -m “message”

5. when you run this command then the changes which you made in intellij they are modified inside working directory and but you don’t add them inside repo but you add them inside stash storage. And the status of working and repo is again same level at first commit level.

The key point is that the file remains in your working directory even after you have added and committed it. The committed version is stored in the Git repository, and the working directory maintains the most recent changes you've made. If you make further modifications to the file, you would need to **git add** again to stage those changes for the next commit.



****

**How data bring back from stash storage area????????**

1. There is a command to check the status of stash -----------git stash list----------------this show what all present inside stash.
2. now which color of car’s data you want to putt back, data is stored inside the stash on the basis of index number.
3. to catch back data from stash to working directory there is a command -------git stash apply index number--------------the data in stash will also remain there it will not be deleted.
4. now here is important point, if you want that, you need green car instead of blue or indigo, but you can-not use the command again ------git stash apply 2---------because first you have to empty the working directory or you have to make the working directory and repo at same level.

now you will again use ---------git stash -m “blue car 2” ---------- to push back the code inside stash. And now working directory is at same level with repo.

now you can fetch back any of your choice car by --------git stash apply 5 -------------------

----------------------------------------------git stash drop --------delete latest commit ---------------------

---------------------------------------------git stash drop 2-------------------it deletes commit index 2 added in stash area.

now you have finalized suppose green car.

now you will add this car inside staging and the make the commit of this car.

now your boss says that I need blue car.

**now how you will fetch blue car from stash????????**

1. first the question is that why you can not use git stash apply command to fetch blue car from stash.

because now your working directory and repo are at same level. If you will use command git stash -m “message”----------

1. This command put your green car inside stash and at same time it will revert your first commit from repo and again it will add the green car inside working directory.

**Now there is conflict what is conflict ????????**

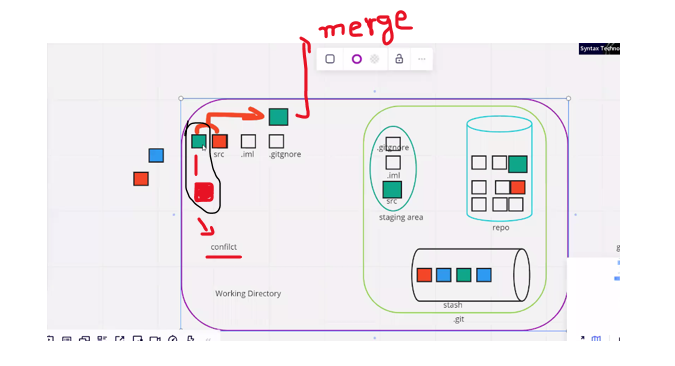
1. Green car is already present inside working directory and you fetch blue car from stash and add it inside the working directory.
2. Now there is conflict git is confused that there are two cars at same point which one I have to select.
3. Git is very vigilant about history. It now that at this line first was what and now what is at this line.
4. To overcome this issue git will merge both the cars and will ask you which car you want to select
5. Now you choose your own choice car and delete the other cars code in intellij and then add and then commit it.

Important question??? If you have stashed applied and you want to apply previous stash how will do ????

1. first you will stash previously applied stash and then you will apply new stash

2. or you will send previous stash to repository by commit command and then you will apply previous stash but in this case there is conflict .

2. stash save the changes also it reverts the first commit back into working directory.



**Class 04 Date:12-12-2023**

**Important Scenario;**

**Suppose you are working on new class headlight because car color has been finalized.**

**But your boss say hay , do some changes in previous car class and then send it to me on github. But you are working on headlight class, now what you will do.?????**

**The answer is**

**1. you will stash your head light class**

**2.now you will make changes in your car class**

**3. and these changes you will send to repo by commit it.**

**4. now you will apply stash on headlight and continue your working on headlights.**

**Now there is another scenario , your headlight class is untracked and there is a command**

**----git stash -u--------- this command is used to add untracked files inside stash area.**

**Now you made changes on green car code and then you add it to repo**

**Now you want to pull back headlight class , is there will be any conflict when you will pull head light class??**

**The answer is no, because headlight is separate class, it is not conflict because git track the changes in code by line wise not by class to class.**

**When does conflict occur in stash?????? When there is tracked file is present in working directory and your pulling the code from stash area back.**

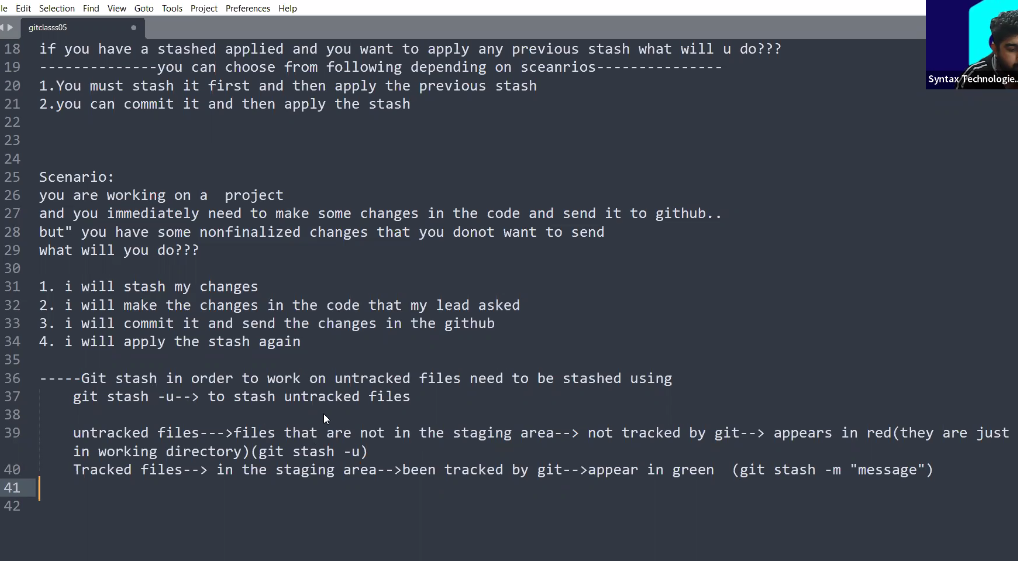
**------- untracked mean ------- files which are inside working directory.----red in color**

**-------if you want to stash untracked files -------git stash -u “message”-----command is used.**

**--------tracked files mean---------the files which are in staging area-------green color files**

**--------to stash tracked files what is command------git stash -m “message”-------- this is for stash tracked files.**

**-----git stash -m “message”----------this command is used to stash tracked files**

****

Yes, the stash area in Git is preserved across Git sessions as long as you do not perform any actions that explicitly remove or clear the stash. If you stash changes using **git stash save** and then close Git Bash or exit your terminal, the stash will still be available when you open Git Bash again in the same repository.

---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**Lets learn some mor git stash commands;**

**Delete git stash----------**

**The command to delete any stash is -----------git stash drop 4----------------- here 4 is index. To check index ---git stash list------**

Another command which is used to add stash and after adding delete stash is --------git stash pop 4------------

Scenario; all above code which we done, now push all code in github

Is we create a new remote repository for every local repository????????

Its your own choice, you can create a single remote repo for many local repository.

Or you can have one to one repo , mean for every local repo, every remote separate repo.

How you will push the new local repo on new remote repo

1. create a connection between local and remote repo----------git remote add origin main url-------------

2. now in which branch you want to send code--------------------git branch -M main-----------------------

3. not push the code to remote repo --------------------------------git push -u origin main----------------

-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

---------------------------------------------------------Clone the project --------------------------------------------------------------

Scenario; is , there are two persons which are working of car project they have clone repository of same project in theirs pc.

They asked the leader hey send us the link of car project. He send the link of github repository. Both team member clone from that link.

(HEAD -> master, origin/master, origin/HEAD) ---- >> this mean local and remote are at same level are synchronized.

**HEAD -> master:**

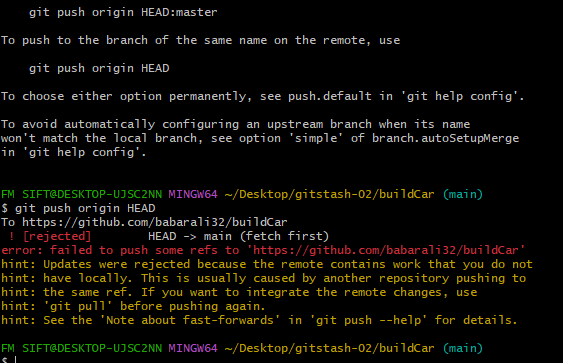
* + **HEAD** is a special pointer in Git that points to the latest commit in the branch that you have currently checked out.
  + **master** is the name of the branch you currently have checked out.

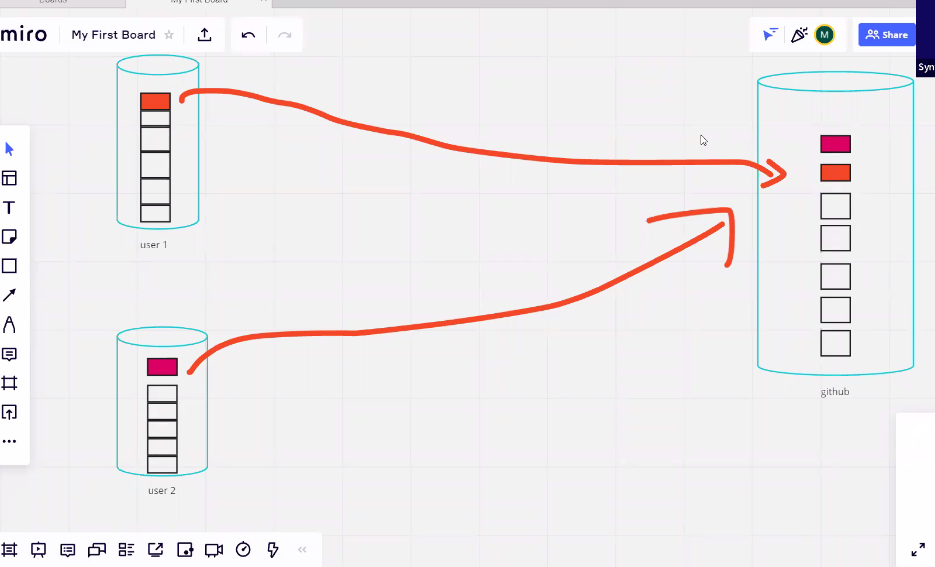
This part indicates that your local repository's working directory is on the **master** branch, and the latest commit on this branch is the one that **HEAD** is pointing to.

1. **origin/master:**
   * **origin** is a default name for the remote repository from which you cloned your local repository.
   * **master** is the branch on the remote repository (origin) that corresponds to your local **master** branch.

This part indicates that the last fetched or pulled commit from the remote repository's **master** branch is stored locally as **origin/master**. This is not necessarily the latest commit on the remote; it's just the last known state of the remote's **master** branch in your local repository.

1. **origin/HEAD:**
   * **origin/HEAD** refers to the default branch on the remote repository. It's a symbolic reference to the default branch on the remote, often the same as the local default branch (e.g., **master**).





1. both team member work on each cloned repo inside their pc.

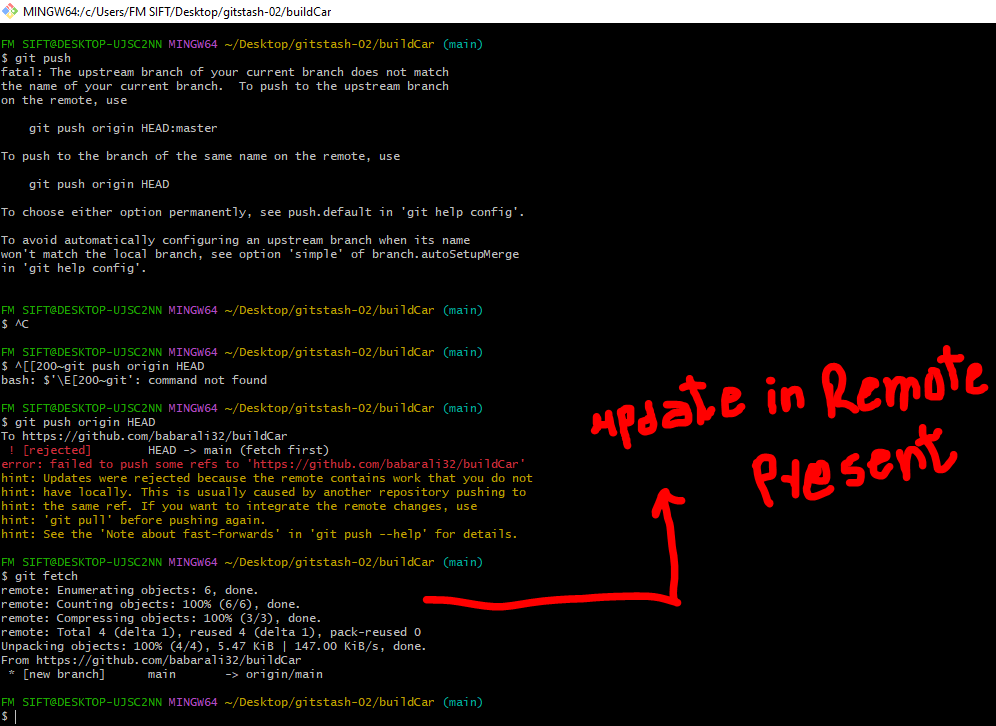
2.both make some changes in code and after complete they push the code.

3. the code who user will push the first will be save in remote repo.

4.but the user who will send the code later, he can not push the code he will get error.

------------------------------git fetch--------------command is used to tell that there are update on remote repo or not.

Updates will shown for that user who will submit file in remote later. Who will be on 2nd position.

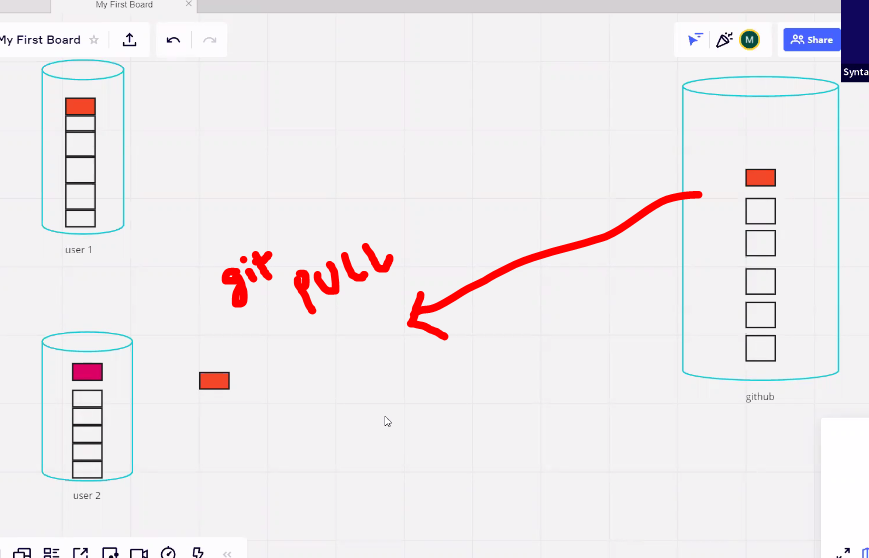


I checked in remote repo there was update now what should I do now.????

----------use command --------------git pull------------------------

This pull command pull the update and send it to user 2 but there is already house full , then what should I do.

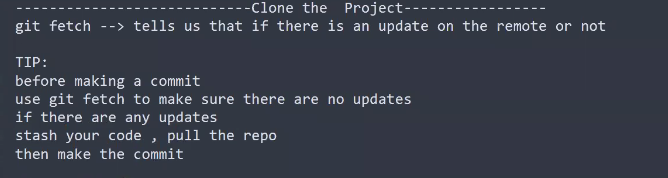
Use stash



2. some one says use stash. But remember stash is useful only when the file has not committed. But in this case file is committed

Now what we should do ??????

**Tips; for user 2, it’s a tip for everyone. Just in this scenario use this.**

****

To take this tip concept watch last 10 minutes video number 4 of git.

------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

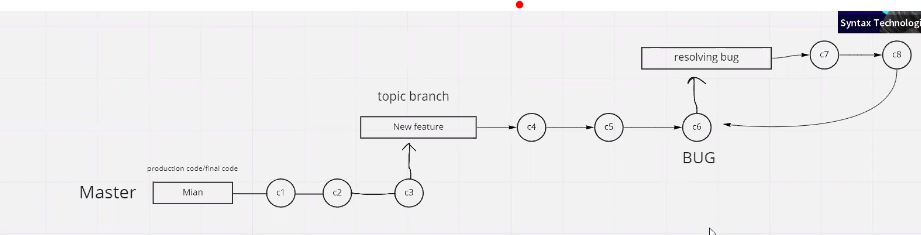
**Class 05 date;13-12-2023**

**In git why do we need branch?????**

**Branches** allow **you** to work on different parts of a project without impacting the main **branch**. When the work is complete, a **branch** can be merged with the main ...

**Scenario**; suppose main branch ------- > has facebook production code-------- > and mark Zuckerberg hires a developer to update an emoji in facebook. Now tell what the developer will work on main branch or he create the new branch ---------- the copy of main branch to update the emoji. because may be during update of emoji the other code of facebook in main branch get affected. Billions of dollars lose.

The other features of facebook may disable. to overcome this issue that’s why we need branch.



Hassel --------------- >> larai …. > unban ……. > problem……. Paryshani

Some terminology

1. main/master branch

2. topic branch ------- > new feature branch is called topic branch

Head --- > master ------by default when you are working on branch this is master branch in local repository.

**Naming command…….!!!!!!**

Certainly! The command **git branch -M main** the purpose of this command is to tell the remote repos , hey remote repo i am local repo my name is main please save me as a main branch .it I important to tell remote repo that we are sending you the code is main.

* **git branch**: Command for managing branches in Git.
* **-M**: Option to forcefully rename the branch.
* **main**: The new name you want to give to the branch.

Master ------- >> this is traditional name of main branch in local repository branch

Main --------- >> is fancy and github accept ” main “ name instead of master now a days

-------**git push -u origin main**----------- >>

This is second command to create -----git push --set-upstream origin sun------------

this command is used to create the same branch in git hub repository. Because local repo says that hey, I can not find any branch on git hub with this name. its important because history must be same.

* **git push**: This command is used to push your changes from your local repository to a remote repository.
* **-u**: This option is short for **--set-upstream**. It establishes a tracking relationship between your local branch and the remote branch. After setting the upstream branch, you can simply use **git push** and **git pull** without specifying the branch names.
* **origin**: This is the name of the remote repository. In Git, "origin" is a common default name given to the remote repository from which your local repository was cloned.
* **main**: This is the branch that you are pushing to the remote repository. In this case, you are pushing the changes from your local "main" branch to the remote "main" branch.

So, when you run **git push -u origin main**, you are pushing the changes from your local "main" branch to the "main" branch on the remote repository named "origin," and you're setting up tracking for future pushes and pulls

------------------------------------------------------------------------------------------------------

The current branch main has no upstream branch. --- >> this line mean that in git hub repository there is no branch at this name “main” please create the same branch in main so that history become same.

(HEAD -> main, origin/main)🡪> this line mean , local repo and remote repo are synchronized. Main branch of local repo and main branch of origin repo are at same commit.

---- >>>> **how create a branch in repo ?????????**

The command to create a branch is -------------------git branch sun--------------sun is name you can choose any name

But when you want to rename the branch then ----------git branch -M main----------here -M is used for rename the branch . in local repo the name of branch which is default created is master, and then we rename this branch as main.

"Checking out" an item usually **means borrowing it in a way that records that you are the one who borrowed it**.

**How to switch the branch??????**

We are in local main branch , now we want to switch from local main branch to topic sun branch.

----------------------------------------------------------git switch sun-------------------

To switch from branch to branch command is -----------------git checkout sun---------sun is name of branch.

**How to check in which branch we are present????????**

The command for checking the branch is ----------git branch---------------------- \* show the working branch.

Now I want to create another branch from sun branch or topic branch.

**How can merge branch????????**

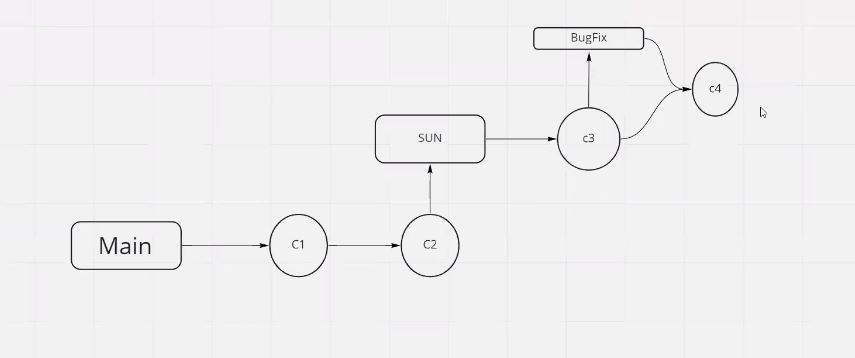
1. you want to merge bug fix branch with sun branch.

2. go to sun branch and there type a command to merge bug fix command with sun branch the command is

-----------------------------git merge bugfix-------------------------------

**What is fast-forward strategy??????**

This is the strategy which git uses to merge the branches.



What is command to delete a branch ???????//

-------------------------------------------------------------------------git branch -d BugFix-------------------this will delete branch..

**Whenever we have new branch on git hub and we want those branches in local,**

**To download branches from github there is command as that we have to do?????**

-------------------------------**git branch -a-----------------------------** command is used to list all branches in a Git repository, including both local and remote branches. The **-a** flag stands for "all."

When you run this command, you'll see a list of branches in your repository. Local branches are prefixed with **\***, and remote branches are listed with the remote name (such as **origin/branch\_name**). Here's an example output:

---------git branch -a--------- this command will show you all the branches in git hub.

To download a an branch from the list.

----------switch to that branch---------------------git switch planet01---------------this command will download for you branch with name planet01.

-----git pull-------- does not download all branches it just update the current branch.

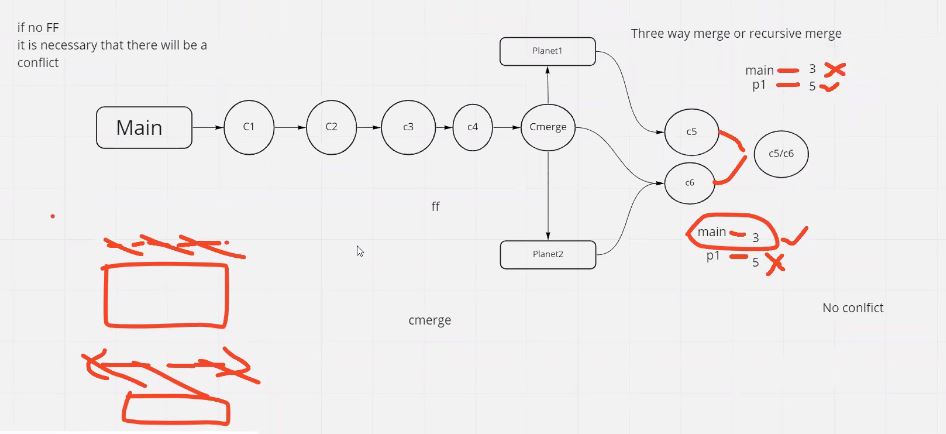
Class 06 date;16-12-2023

**What is Three-way merge or recursive merge??**

When fast forward merge occurs?? It is occur then no changes has been made inside the branch from which a new branch has been created. And new commit is added back to parent commit this is fast forward strategy.

Three way strategy is used when there is change in parent commit from which a child branch has been created, and some one other added his code in parent branch. Now there is update and you don not add his code. That’s why three way merge is used.

When there is conflict between the code then there use a cursive merge technique.



A three-way merge is a merging strategy used by Git when combining changes from two different branches. It involves three commits:

1. **The common ancestor commit (the base):** This is the commit from which both branches diverged. It serves as a reference point to identify the changes made on each branch.
2. **The current branch's commit (ours):** The changes made on the branch where the merge is being initiated.
3. **The other branch's commit (theirs):** The changes made on the branch that is being merged into the current branch.

Git compares the changes introduced in the "ours" and "theirs" commits relative to the common ancestor to automatically merge the changes. In cases where changes do not conflict, Git can perform an automatic merge. However, when conflicting changes are detected (i.e., changes made in both branches that overlap), Git requires manual intervention to resolve the conflicts.

The **git pull** command is used to fetch changes from a remote repository and integrate those changes into the current branch. It's a combination of two commands: **git fetch** and **git merge**.

------------------------git pull origin master-----------------------

* + This command is used to update your local repository with changes made by others. It's often used when you're working on a branch, and you want to incorporate the latest changes from the remote repository into your local branch.

In summary, **git clone** is used to create a local copy of a remote repository, while **git pull** is used to update your local repository with changes from a remote repository. They serve different purposes in the Git workflow.

**What is difference in fetch and pull commands??????**

Fetch command show the any update in remoter repo and pull is used to download those updates in your local and merge them in your working branch.

When same lines changed by two different person then conflict occur. When conflict happen occur?

When you want to delete the commit use following commands.

**Undo the Last Commit (Soft Reset):**

If you just want to undo the last commit but keep the changes staged in your working directory, you can use the following command:

**git reset --soft HEAD^**

This command moves the **HEAD** pointer and the branch pointer back one commit, but keeps the changes from the undone commit in your working directory as staged changes.

**Completely Remove the Last Commit (Hard Reset):**

If you want to completely remove the last commit and discard the changes, you can use a hard reset:

**git reset --hard HEAD^**

Be cautious with **git reset --hard** as it discards all changes from the specified commit onward, and those changes will be lost permanently.

**Remove a Specific Commit:**

If you want to remove a commit that is not the last one, you can use the following command:

**git rebase -i HEAD~n**

Replace **n** with the number of commits you want to go back. This will open an interactive rebase where you can choose to delete, edit, or squash commits.

How delete the files from staging area;

o remove a file from the staging area in Git, you can use the following command:

**git reset filename**

Replace **filename** with the actual name of the file you want to unstage. This command resets the changes for the specified file, effectively removing it from the staging area while keeping the modifications in your working directory.

If you want to unstage all files, you can use:

**git reset**

This command will un-stage all changes, but it won't discard the changes in your working directory; it just moves them from the staging area back to the working directory.

----------------------------------------------git push –force ---------this command will delete the commit from remote repository.

This command will delete the commit number 01 from the local repo.

---------------------------------- **git reset --hard HEAD ~1-------------------this command will delete commit from local repo**