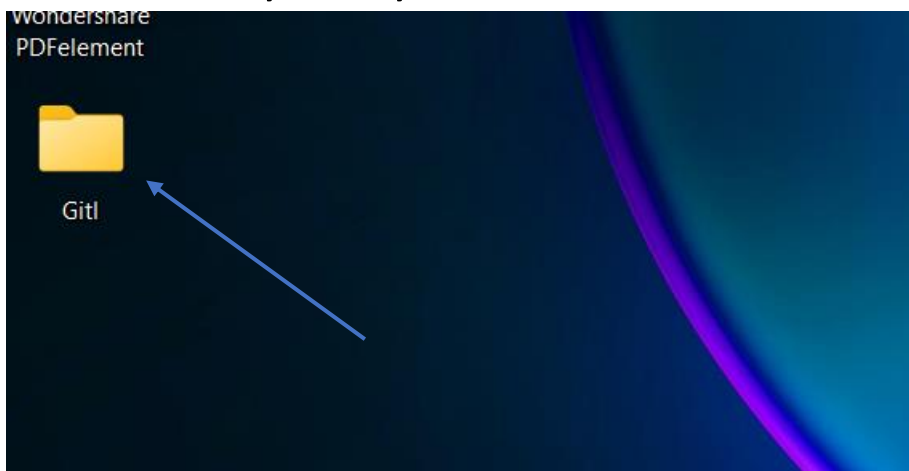

How To Go About With Git

Things to remember:

- Git is an open source version control system to track code changes
- **Repository:** a storage location used to track and manage the changes made in a file.

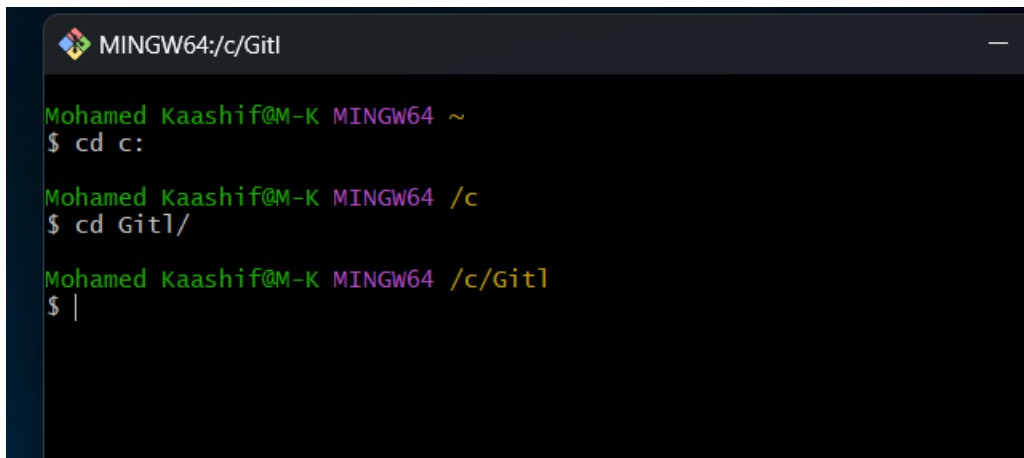
Things to do before starting any program:

- Create a folder on the desktop or in any preferred location on your system.



(I've created a "Gitl" folder located in my 'C:' drive)

- Open Git Bash Terminal and move into that folder with the help of **cd** command.



```
MINGW64:/c/Git1
Mohamed Kaashif@M-K MINGW64 ~
$ cd c:

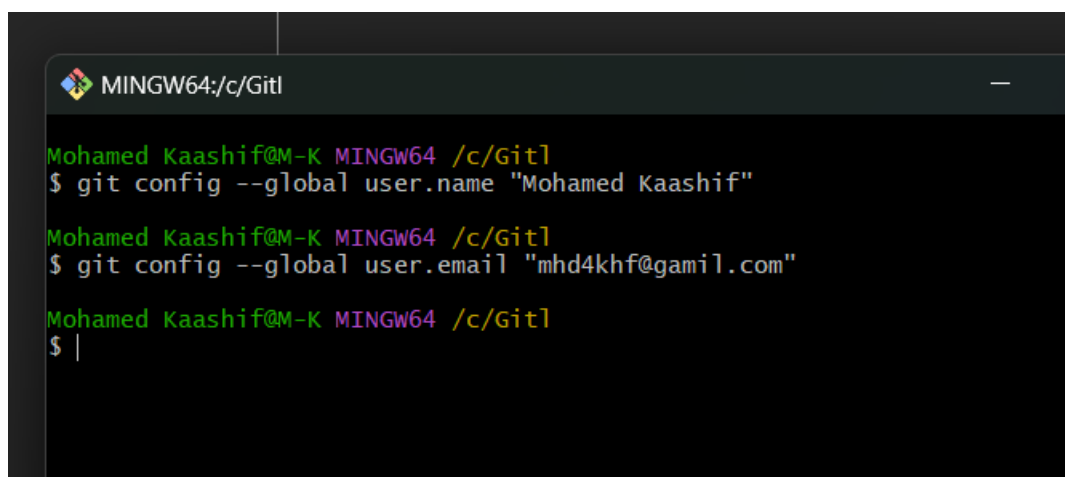
Mohamed Kaashif@M-K MINGW64 /c
$ cd Git1/

Mohamed Kaashif@M-K MINGW64 /c/Git1
$ |
```

- Here configure your name and email so that git can track who has made the changes, use the command

a. git config --global user.name "Your Github account name"

b. git config --global user.email "Your email"



```
MINGW64:/c/Git1
Mohamed Kaashif@M-K MINGW64 /c/Git1
$ git config --global user.name "Mohamed Kaashif"

Mohamed Kaashif@M-K MINGW64 /c/Git1
$ git config --global user.email "mhd4khf@gamil.com"

Mohamed Kaashif@M-K MINGW64 /c/Git1
$ |
```

- Initialize git int this folder with the command,
a. git init

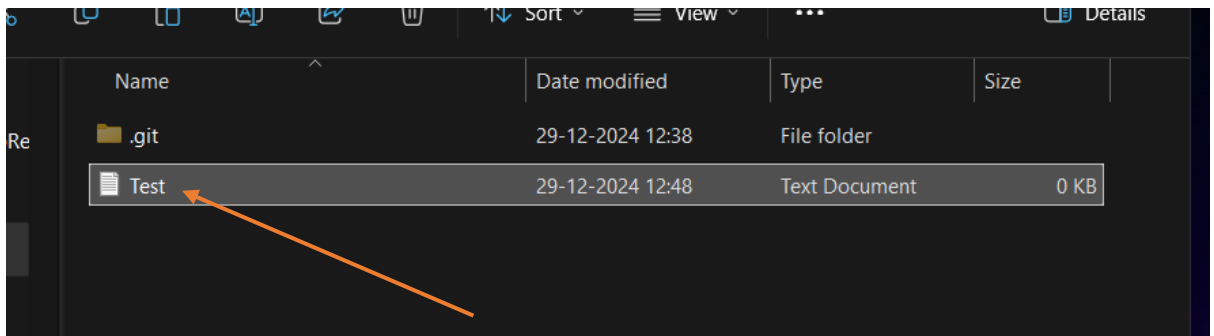
A screenshot of a Windows command prompt window with a dark background. The title bar at the top shows the Windows logo and the text 'MINGW64:/c/Git1'. The prompt shows the user 'Mohamed Kaashif@M-K' in a green font, the shell 'MINGW64' in a purple font, and the current directory '/c/Git1' in a yellow font. The user has entered the command '\$ git init'. The output of the command is 'Initialized empty Git repository in C:/Git1/.git/'. The prompt then shows the user at the '(master)' branch, with a vertical cursor line after the '\$' symbol.

```
Mohamed Kaashif@M-K MINGW64 /c/Git1
$ git init
Initialized empty Git repository in C:/Git1/.git/
Mohamed Kaashif@M-K MINGW64 /c/Git1 (master)
$ |
```

Program 1

FOLLOW THE ABOVE STEPS AND THEN:

- Add a file in the folder you've created ("Gitl") from the file manager



- Get back on git bash and use the command
a. `git add .` (or) `git add Test.txt`
(`git add .` adds all the files present in the folder)
- This adds the Test.txt to the staging area.

```
Mohamed Kaashif@M-K MINGW64 /c/Gitl (master)
$ git add .

Mohamed Kaashif@M-K MINGW64 /c/Gitl (master)
$
```

- Finally commit the change with the command
b. `Git commit -m "Your message"`

```
Mohamed Kaashif@M-K MINGW64 /c/Gitl (master)
$ git commit -m "Added a file named Test.txt"
[master (root-commit) 6d4fe56] Added a file named Test.txt
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Test.txt

Mohamed Kaashif@M-K MINGW64 /c/Gitl (master)
$
```

Program 2

AFTER INITIALIZING GIT IN THE FOLDER,

```
Mohamed Kaashif@M-K MINGW64 /c/Git1 (master)
$
```

- If your terminal doesn't show the "(master)" at the end, simply use the command

1. **git checkout master**

By doing this, you'll get the same as above.

- Create a branch named feature-branch (any name is fine) and move into it with the help of a single command

2. **git checkout -b feature-branch**

```
Mohamed Kaashif@M-K MINGW64 /c/Git1 (master)
$ git checkout -b feature-branch
Switched to a new branch 'feature-branch'

Mohamed Kaashif@M-K MINGW64 /c/Git1 (feature-branch)
$
```

- Once you're in the feature branch, add a file in it by using the command

3. **touch test1.txt**

```
$ git checkout -b feature-branch
Switched to a new branch 'feature-branch'

Mohamed Kaashif@M-K MINGW64 /c/Git1 (feature-branch)
$ touch test1.txt

Mohamed Kaashif@M-K MINGW64 /c/Git1 (feature-branch)
$ |
```

- Add this test1.txt to the staging area and commit it with a message.(This file is only available to the feature-branch)

```
Mohamed Kaashif@M-K MINGW64 /c/Git1 (feature-branch)
$ touch test1.txt

Mohamed Kaashif@M-K MINGW64 /c/Git1 (feature-branch)
$ git add .

Mohamed Kaashif@M-K MINGW64 /c/Git1 (feature-branch)
$ git commit -m "Added a file test1.txt to the feature-branch"
[feature-branch b495f50] Added a file test1.txt to the feature-branch
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 test1.txt


Mohamed Kaashif@M-K MINGW64 /c/Git1 (feature-branch)
$ |
```

- Next checkout back to the “master” branch

```
create mode 100644 test1.txt

Mohamed Kaashif@M-K MINGW64 /c/Git1 (feature-branch)
$ git checkout master
Switched to branch 'master'

Mohamed Kaashif@M-K MINGW64 /c/Git1 (master)
$
```



- Finally merge the feature branch to the master branch by using the command
4. git merge feature-branch

(This will now add the test1.txt which was only available to the feature branch to the master branch)

```
Mohamed Kaashif@M-K MINGW64 /c/Git1 (master)
$ git merge feature-branch
Updating 6dafa56..b495f50
Fast-forward
 test1.txt | 0
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 test1.txt
```

Program 3

- Here we will make a file Test.txt and a file test1.txt on the master branch. Next we will commit the Test.txt and stash save the test1.txt by adding it to the staging area.

```
MINGW64:/c/Gitl
Mohamed Kaashif@M-K MINGW64 /c/Gitl
$ git init
Initialized empty Git repository in C:/Gitl/.git/

Mohamed Kaashif@M-K MINGW64 /c/Gitl (master)
$ touch Test.txt

Mohamed Kaashif@M-K MINGW64 /c/Gitl (master)
$ git add .

Mohamed Kaashif@M-K MINGW64 /c/Gitl (master)
$ git commit -m "Created a Test.txt file"
[master (root-commit) d40debe] Created a Test.txt file
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Test.txt

Mohamed Kaashif@M-K MINGW64 /c/Gitl (master)
$ touch test1.txt

Mohamed Kaashif@M-K MINGW64 /c/Gitl (master)
$ git add .

Mohamed Kaashif@M-K MINGW64 /c/Gitl (master)
$ git stash save "Stash saving this test1.txt"
Saved working directory and index state On master: Stash saving this test1.txt
```

- Then we'll create a new branch called feature-branch and move into it and apply the stash changes

```
Saved working directory and index state On master: Stash saving this test1.txt

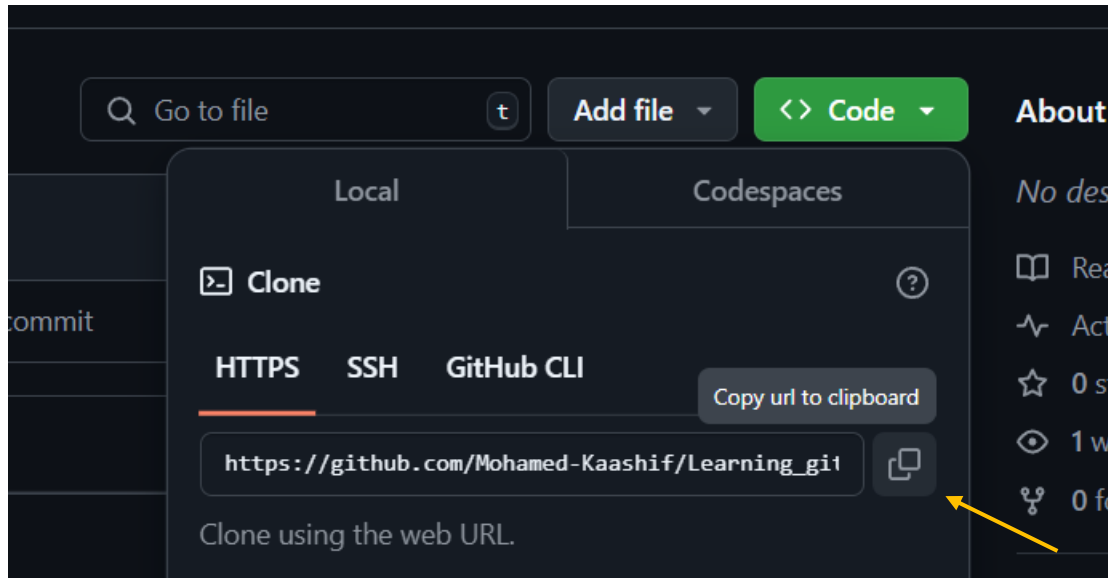
Mohamed Kaashif@M-K MINGW64 /c/Gitl (master)
$ git checkout -b feature-branch
Switched to a new branch 'feature-branch'

Mohamed Kaashif@M-K MINGW64 /c/Gitl (feature-branch)
$ git stash apply
On branch feature-branch
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   test1.txt
```


Program 4

FOR THIS ONE PROGRAM WE CAN SKIP THE PROCESS OF INITIALIZING GIT.

- Go to the github's website and copy the url of any project you'd like to enhance or work on.



- Next head to git bash and use the command
1. git clone url

```
Mohamed Kaashif@M-K MINGW64 /c/Git1 (master)
$ git clone https://github.com/Mohamed-Kaashif/Learning_git.git
Cloning into 'Learning_git'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.

Mohamed Kaashif@M-K MINGW64 /c/Git1 (master)
$ |
```

Program 5

- Here we create a file Test.txt in the master branch and commit it.

```
MINGW64:/c/Git1

Mohamed Kaashif@M-K MINGW64 /c/Git1 (master)
$ git init
Initialized empty Git repository in C:/Git1/.git/

Mohamed Kaashif@M-K MINGW64 /c/Git1 (master)
$ touch Test.txt

Mohamed Kaashif@M-K MINGW64 /c/Git1 (master)
$ git add .

Mohamed Kaashif@M-K MINGW64 /c/Git1 (master)
$ git commit -m "Created a Test.txt"
[master (root-commit) 008464d] Created a Test.txt
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Test.txt
```

- Next we create and move into a feature-branch and add a file Test1.txt. commit this Test1.txt as well.

```
create mode 100644 Test.txt

Mohamed Kaashif@M-K MINGW64 /c/Git1 (master)
$ git checkout -b feature-branch
Switched to a new branch 'feature-branch'

Mohamed Kaashif@M-K MINGW64 /c/Git1 (feature-branch)
$ touch Test1.txt

Mohamed Kaashif@M-K MINGW64 /c/Git1 (feature-branch)
$ git add .

Mohamed Kaashif@M-K MINGW64 /c/Git1 (feature-branch)
$ git commit -m "Created a Test1.txt in feature branch"
[feature-branch 807c6fb] Created a Test1.txt in feature branch
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Test1.txt
```

- Next we will checkout to the master branch and use the command **git rebase feature-branch**

```
feature-branch 807c01b] created a test1.txt in feature-branch  
1 file changed, 0 insertions(+), 0 deletions(-)  
create mode 100644 Test1.txt  
  
Mohamed Kaashif@M-K MINGW64 /c/Git1 (feature-branch)  
$ git checkout master  
Switched to branch 'master'  
  
Mohamed Kaashif@M-K MINGW64 /c/Git1 (master)  
$ git rebase feature-branch  
Successfully rebased and updated refs/heads/master.
```

Program 6

SAME AS PROGRAM 2

- Here the only added feature is that we are adding a message while merging the feature-branch's file.

```
Mohamed Kaashif@M-K MINGW64 /c/Git1
$ git init
Initialized empty Git repository in C:/Git1/.git/

Mohamed Kaashif@M-K MINGW64 /c/Git1 (master)
$ git add .

Mohamed Kaashif@M-K MINGW64 /c/Git1 (master)
$ git commit -m "Added a Test.txt file"
[master (root-commit) 8fd7605] Added a Test.txt file
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Test.txt.txt

Mohamed Kaashif@M-K MINGW64 /c/Git1 (master)
$ git checkout -b Feature-branch
Switched to a new branch 'Feature-branch'

Mohamed Kaashif@M-K MINGW64 /c/Git1 (Feature-branch)
$ touch test1.txt

Mohamed Kaashif@M-K MINGW64 /c/Git1 (Feature-branch)
$ git add .

Mohamed Kaashif@M-K MINGW64 /c/Git1 (Feature-branch)
$ git commit -m "Added a file test1.txt to the feature-branch"
[Feature-branch 829b71d] Added a file test1.txt to the feature-branch
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 test1.txt

Mohamed Kaashif@M-K MINGW64 /c/Git1 (Feature-branch)
$ git checkout master
Switched to branch 'master'
```

- Once we've got here, then we simply need to use the same merge command with a flag (i.e -m to add a message while merging)

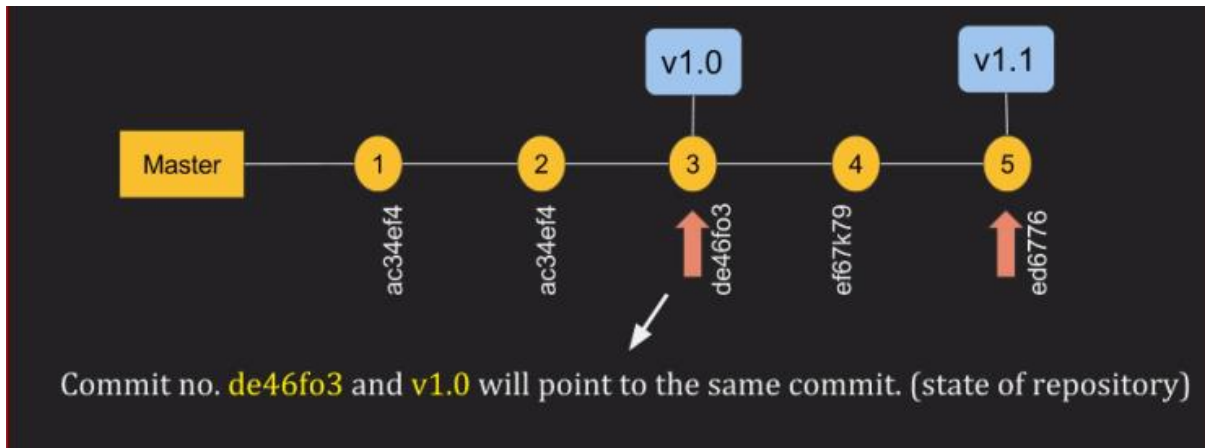
```
Switched to branch 'master'

Mohamed Kaashif@M-K MINGW64 /c/Git1 (master)
$ git merge feature-branch -m "Adding the test1.txt to the master branch"
Updating 8fd7605..829b71d
Fast-forward (no commit created; -m option ignored)
 test1.txt | 0
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 test1.txt
```

Program 7

INITIALIZE GIT IN A FOLDER AND MAKE A COMMIT (AS DONE IN PROGRAM 1)

- Tags are added to a specific commit something like a mile stone.



- To add a tag, simply use the command
1. `git tag version`

```
MINGW64:/c/Gitl
Mohamed Kaashif@M-K MINGW64 /c/Gitl (master)
$ git add .
Mohamed Kaashif@M-K MINGW64 /c/Gitl (master)
$ git commit -m "Adding a tag"
[master 8edbff9] Adding a tag
1 file changed, 2 insertions(+)
Mohamed Kaashif@M-K MINGW64 /c/Gitl (master)
$ git tag v1.0
Mohamed Kaashif@M-K MINGW64 /c/Gitl (master)
$ git log
commit 8edbff9bb64b18f9ced91bc58a7b687d6f12df79 (HEAD -> master, tag: v1.0)
Author: Mohamed Kaashif <mhd4khf@gamil.com>
Date: Sun Dec 29 15:59:56 2024 +0530

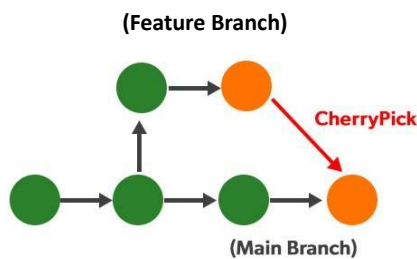
Adding a tag
```

(the “git log” command shows all the commits made by the user)

Program 8

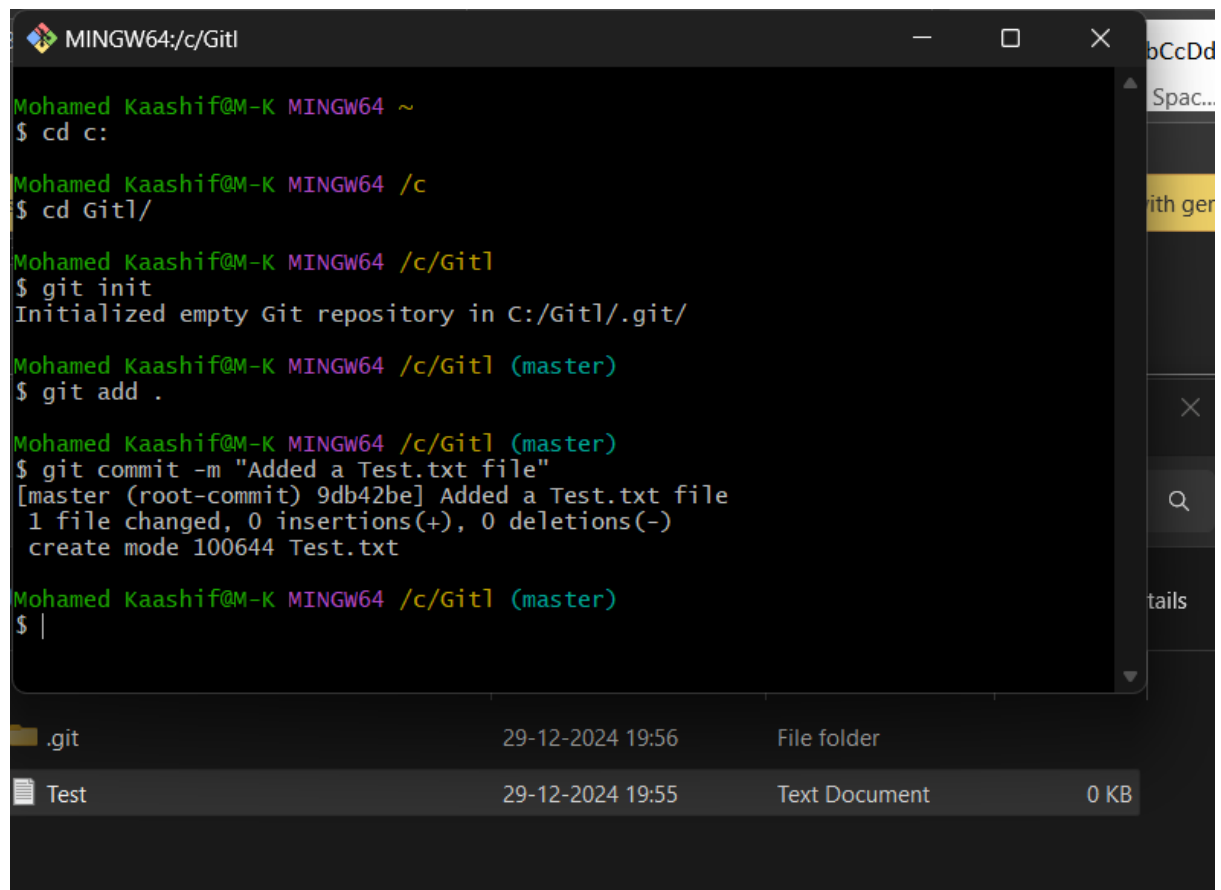
CHERRY-PICK

- Here, we create and switch from the main branch to a feature branch, add in a file and commit the change.
- Once that's done, we switch back to the main branch and use the cherry-pick command.



To begin,

- We first repeat the process as program 1 (initialize, create a Test.txt and commit the changes)



```
MINGW64:/c/Git1
Mohamed Kaashif@M-K MINGW64 ~
$ cd c:

Mohamed Kaashif@M-K MINGW64 /c
$ cd Git1/

Mohamed Kaashif@M-K MINGW64 /c/Git1
$ git init
Initialized empty Git repository in C:/Git1/.git/

Mohamed Kaashif@M-K MINGW64 /c/Git1 (master)
$ git add .

Mohamed Kaashif@M-K MINGW64 /c/Git1 (master)
$ git commit -m "Added a Test.txt file"
[master (root-commit) 9db42be] Added a Test.txt file
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Test.txt

Mohamed Kaashif@M-K MINGW64 /c/Git1 (master)
$ |
```

File Name	Date	Type	Size
.git	29-12-2024 19:56	File folder	
Test	29-12-2024 19:55	Text Document	0 KB

- Next create and move into the “feature-branch” with the command, **git checkout -b feature-branch**

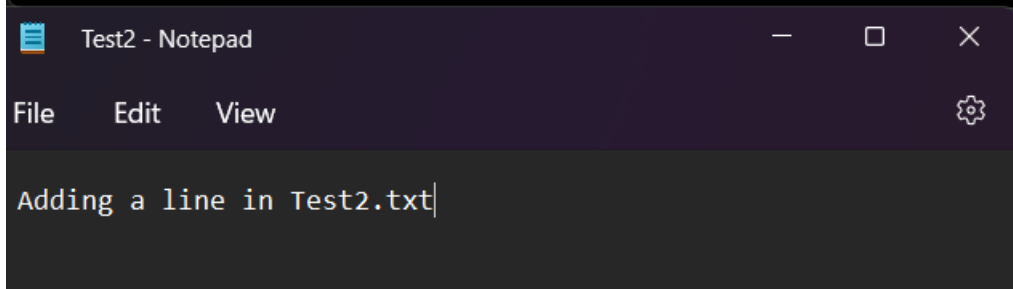
```
Mohamed Kaashif@M-K MINGW64 /c/Git1 (master)
$ git checkout -b feature-branch
Switched to a new branch 'feature-branch'
```

- Create a new file here called “Test2.txt” with the “touch” command.

```
Mohamed Kaashif@M-K MINGW64 /c/Git1 (feature-branch)
$ touch Test2.txt
```

- Open the “Test2.txt” file with the help of the notepad command

```
$ touch Test2.txt
Mohamed Kaashif@M-K MINGW64 /c/Git1 (feature-branch)
$ notepad Test2.txt
```



- After making some change in the Test2.txt file, save the it and close the notepad window to get back on git's terminal.
- Add it to the staging area and commit the changes.

```
$ notepad Test2.txt
Mohamed Kaashif@M-K MINGW64 /c/Git1 (feature-branch)
$ git add .
Mohamed Kaashif@M-K MINGW64 /c/Git1 (feature-branch)
$ git commit -m "Added a Test2.txt with a message"
[feature-branch b84a866] Added a Test2.txt with a message
1 file changed, 1 insertion(+)
create mode 100644 Test2.txt
```

- Next, use the command git log --oneline, to get the commit ids of the feature branch

```
Mohamed Kaashif@M-K MINGW64 /c/Git1 (feature-branch)
$ git log --oneline
b84a866 (HEAD -> feature-branch) Added a Test2.txt with a message
9db42be (master) Added a Test.txt file
```

- Now checkout to the “master” branch

```
b84a866 (HEAD -> feature-branch) Added a Test2.txt with a
9db42be (master) Added a Test.txt file

Mohamed Kaashif@M-K MINGW64 /c/Git1 (feature-branch)
$ git checkout master
Switched to branch 'master'
```


- Copy a commit id from the feature branch (that we got on using the command `git log --online`)
- And use it with the command, **git cherry-pick *commit_id***

```
Switched to branch 'master'  
  
Mohamed Kaashif@M-K MINGW64 /c/Git1 (master)  
$ git cherry-pick b84a866  
[master d3a7adb] Added a Test2.txt with a message  
Date: Sun Dec 29 20:53:14 2024 +0530  
1 file changed, 1 insertion(+)  
create mode 100644 Test2.txt
```

Program 9

DO PROGRAM 1, AND THEN

- Open the Test.txt from your file explorer and add any text. Save the file and on git bash add it to the staging area and commit it.

```
MINGW64:/c/Git1
Mohamed Kaashif@M-K MINGW64 /c/Git1 (master)
$ git add .

Mohamed Kaashif@M-K MINGW64 /c/Git1 (master)
$ git commit -m "Added a test file"
On branch master
nothing to commit, working tree clean

Mohamed Kaashif@M-K MINGW64 /c/Git1 (master)
$ git add .

Mohamed Kaashif@M-K MINGW64 /c/Git1 (master)
$ git commit -m "Added text to test file"
[master 5e6766c] Added text to test file
1 file changed, 1 insertion(+)

Mohamed Kaashif@M-K MINGW64 /c/Git1 (master)
$ |
```

After this command, open and edit the text file

- To view the commit id's use the command
1. git log --oneline

```
Mohamed Kaashif@M-K MINGW64 /c/Git1 (master)
$ git log --oneline
5e6766c (HEAD -> master) Added text to test file
a5dd3fc removed the test.txts contents
8edbff9 (tag: v1.0) Adding a tag
6fd5754 added a test file
```

- Copy any of the commit id and use it with the command

2. `git show commit_id`

```
Mohamed Kaashif@M-K MINGW64 /c/Git1 (master)
$ git show 5e6766c
commit 5e6766c3bbc287576f4f2d90d59dd33ece6113c7 (HEAD -> master)
Author: Mohamed Kaashif <mhd4khf@gamil.com>
Date: Sun Dec 29 16:57:09 2024 +0530

    Added text to test file

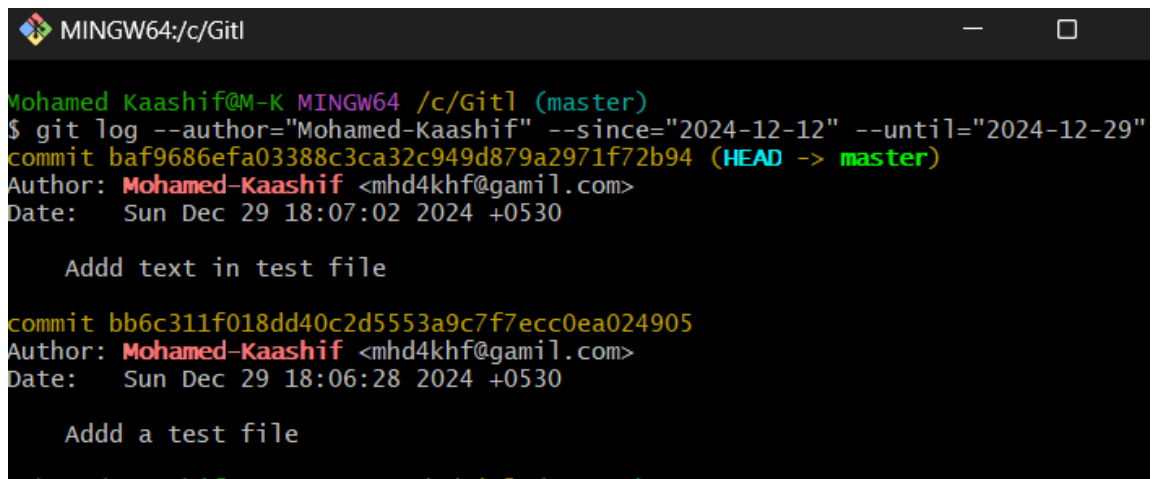
diff --git a/test.txt b/test.txt
index e69de29..e833ecb 100644
--- a/test.txt
+++ b/test.txt
@@ -0,0 +1 @@
+new line added
\ No newline at end of file

Mohamed Kaashif@M-K MINGW64 /c/Git1 (master)
$
```

Program 10

USING THE COMMAND 'GIT LOG'

- Perform at least 1 commit in you repository.
- Next use the command
 1. git log --
author="*the_user_name_you_gave_while_confi*
guring" --since="*any_date*" --
until="*current_date*"

A screenshot of a terminal window titled 'MINGW64:/c/Gitl'. The prompt is 'Mohamed Kaashif@M-K MINGW64 /c/Gitl (master)'. The command '\$ git log --author="Mohamed-Kaashif" --since="2024-12-12" --until="2024-12-29"' is entered. The output shows two commits. The first commit has hash 'baf9686efa03388c3ca32c949d879a2971f72b94' and message 'Addd text in test file'. The second commit has hash 'bb6c311f018dd40c2d5553a9c7f7ecc0ea024905' and message 'Addd a test file'. Both commits are attributed to 'Mohamed-Kaashif <mhd4khf@gamil.com>' and dated 'Sun Dec 29 18:07:02 2024 +0530' and 'Sun Dec 29 18:06:28 2024 +0530' respectively. The terminal window has standard Windows window controls (minimize, maximize, close) in the top right corner.

```
MINGW64:/c/Gitl
Mohamed Kaashif@M-K MINGW64 /c/Gitl (master)
$ git log --author="Mohamed-Kaashif" --since="2024-12-12" --until="2024-12-29"
commit baf9686efa03388c3ca32c949d879a2971f72b94 (HEAD -> master)
Author: Mohamed-Kaashif <mhd4khf@gamil.com>
Date:   Sun Dec 29 18:07:02 2024 +0530

    Addd text in test file

commit bb6c311f018dd40c2d5553a9c7f7ecc0ea024905
Author: Mohamed-Kaashif <mhd4khf@gamil.com>
Date:   Sun Dec 29 18:06:28 2024 +0530

    Addd a test file
```

Program 11

MAKE 5 COMMITS IN A REPOSITORY

- To make 5 commits, you can add in one line or keep making a change in the “Test.txt”.

```
Mohamed Kaashif@M-K MINGW64 /c/Git1 (master)
$ git add .

Mohamed Kaashif@M-K MINGW64 /c/Git1 (master)
$ git commit -m "Added line 1 to Test.txt"
[master 3550b72] Added line 1 to Test.txt
1 file changed, 2 insertions(+), 1 deletion(-)

Mohamed Kaashif@M-K MINGW64 /c/Git1 (master)
$ git add .

Mohamed Kaashif@M-K MINGW64 /c/Git1 (master)
$ git commit -m "Added line 2 to Test.txt"
[master 5f10a6e] Added line 2 to Test.txt
1 file changed, 1 insertion(+)

Mohamed Kaashif@M-K MINGW64 /c/Git1 (master)
$ git add .

Mohamed Kaashif@M-K MINGW64 /c/Git1 (master)
$ git commit -m "Added line 3 to Test.txt"
[master 41e8197] Added line 3 to Test.txt
1 file changed, 1 insertion(+)

Mohamed Kaashif@M-K MINGW64 /c/Git1 (master)
$ git add .

Mohamed Kaashif@M-K MINGW64 /c/Git1 (master)
$ git commit -m "Added line 4 to Test.txt"
[master a13ec94] Added line 4 to Test.txt
1 file changed, 1 insertion(+)

Mohamed Kaashif@M-K MINGW64 /c/Git1 (master)
$ git add .

Mohamed Kaashif@M-K MINGW64 /c/Git1 (master)
$ git commit -m "Added line 5 to Test.txt"
[master 90ea576] Added line 5 to Test.txt
1 file changed, 1 insertion(+)
```

After this command,
open and edit the
text file

- Then use the command

1. git log -n 5

```
Mohamed Kaashif@M-K MINGW64 /c/Git1 (master)
$ git log -n 5
commit 90ea5762ebc18a0ea7e89907352a0a5155c110a9 (HEAD -> master)
Author: Mohamed-Kaashif <mhd4khf@gamil.com>
Date: Sun Dec 29 18:40:03 2024 +0530

    Added line 5 to Test.txt

commit a13ec94c445af3bf946f0fcaab86432b03422c9d
Author: Mohamed-Kaashif <mhd4khf@gamil.com>
Date: Sun Dec 29 18:39:43 2024 +0530

    Added line 4 to Test.txt

commit 41e8197de6f8c35c36619769402262bf692e64f0
Author: Mohamed-Kaashif <mhd4khf@gamil.com>
Date: Sun Dec 29 18:39:30 2024 +0530

    Added line 3 to Test.txt

commit 5f10a6e0ed8b1754d52c2f3c01611a5bd4680ffc
Author: Mohamed-Kaashif <mhd4khf@gamil.com>
Date: Sun Dec 29 18:39:14 2024 +0530

    Added line 2 to Test.txt

commit 3550b72befcdc257c915f2f125ea3a2981ee9d86
Author: Mohamed-Kaashif <mhd4khf@gamil.com>
Date: Sun Dec 29 18:38:55 2024 +0530

    Added line 1 to Test.txt
```

Program 12

MAKE ATLEAST 1 COMMIT TO REVERT

- After making a commit, use the command
1. `git log --oneline`

```
MINGW64:/c/Gitl
Mohamed Kaashif@M-K MINGW64 /c/Gitl
$ git init
Initialized empty Git repository in C:/Gitl/.git/

Mohamed Kaashif@M-K MINGW64 /c/Gitl (master)
$ git add .

Mohamed Kaashif@M-K MINGW64 /c/Gitl (master)
$ git commit -m "Added a file Test.txt"
[master (root-commit) af90fb3] Added a file Test.txt
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Test.txt

Mohamed Kaashif@M-K MINGW64 /c/Gitl (master)
$ git add .

Mohamed Kaashif@M-K MINGW64 /c/Gitl (master)
$ git commit -m "Added line 1 to file Test.txt"
[master 12dfe47] Added line 1 to file Test.txt
1 file changed, 1 insertion(+)

Mohamed Kaashif@M-K MINGW64 /c/Gitl (master)
$ git add .

Mohamed Kaashif@M-K MINGW64 /c/Gitl (master)
$ git commit -m "Added line 2 to file Test.txt"
[master 9e65d8f] Added line 2 to file Test.txt
1 file changed, 2 insertions(+), 1 deletion(-)

Mohamed Kaashif@M-K MINGW64 /c/Gitl (master)
$ git log --oneline
9e65d8f (HEAD -> master) Added line 2 to file Test.txt
12dfe47 Added line 1 to file Test.txt
af90fb3 Added a file Test.txt
```

After this command, open and edit the text file

- Copy any of the above commit ids
- And use the command,
2. `git revert commit_id`

