

```
MINGW64/c/Users/ShwetaPC/Desktop/oops_lab
$ ls
git_commands.pdf  new.txt  oops1.java
ShwetaPC@shweta MINGW64 ~/Desktop/oops_lab (main)
$ pwd
/c/Users/ShwetaPC/Desktop/oops_lab
ShwetaPC@shweta MINGW64 ~/Desktop/oops_lab (main)
$ whoami
ShwetaPC
ShwetaPC@shweta MINGW64 ~/Desktop/oops_lab (main)
$ nano new.txt
ShwetaPC@shweta MINGW64 ~/Desktop/oops_lab (main)
$ cat new.txt
hey!yashi this side.
How are you?
ShwetaPC@shweta MINGW64 ~/Desktop/oops_lab (main)
$ cd ..
ShwetaPC@shweta MINGW64 ~/Desktop
$ cp explore
cp: missing destination file operand after 'explore'
Try 'cp --help' for more information.
ShwetaPC@shweta MINGW64 ~/Desktop
$ cp ./explore
cp: missing destination file operand after './explore'
Try 'cp --help' for more information.
ShwetaPC@shweta MINGW64 ~/Desktop
$ cp explore/
cp: missing destination file operand after 'explore/'
Try 'cp --help' for more information.
ShwetaPC@shweta MINGW64 ~/Desktop
$ cp oops_lab
cp: missing destination file operand after 'oops_lab'
Try 'cp --help' for more information.
ShwetaPC@shweta MINGW64 ~/Desktop
$ ^C
ShwetaPC@shweta MINGW64 ~/Desktop
$ cd oops_lab
ShwetaPC@shweta MINGW64 ~/Desktop/oops_lab (main)
```

Git Bash commands

✓ 1. touch

Purpose: Creates a new empty file.

→ Useful for quickly making new files for code or notes.

Example-

touch index.html

✓ 2. ls

Purpose: Lists files and folders in the current directory.

→ Helps see what's inside your working directory.

Example

ls

✓ 3. git init

Purpose: Initializes a new Git repository in your current folder.

→ Creates a hidden .git/ folder that starts version tracking.

Example:

git init

✓ 4. git status

Purpose: Shows the current state of your working directory and staging area.

→ It tells you:

- What files are **untracked**
- What files are **staged**
- What files are **modified**

Example:

git status

```
MINGW64/c/Users/ShwetaPC/Desktop/oops_lab
ShwetaPC@shweta MINGW64 ~/Desktop/oops_lab (main)
$ cat oops1.java
public class oops1 {
    public static void main(String[] args) {
        // Your code starts here
        System.out.println("Hello from oops.java!");
    }
}
ShwetaPC@shweta MINGW64 ~/Desktop/oops_lab (main)
$ chmod 700 oops1.java
ShwetaPC@shweta MINGW64 ~/Desktop/oops_lab (main)
$ touch intro.txt
ShwetaPC@shweta MINGW64 ~/Desktop/oops_lab (main)
$ cp new.txt intro.txt
ShwetaPC@shweta MINGW64 ~/Desktop/oops_lab (main)
$ cat intro.txt
hey/yashi this side.
how are you?
ShwetaPC@shweta MINGW64 ~/Desktop/oops_lab (main)
$ rm new.txt
ShwetaPC@shweta MINGW64 ~/Desktop/oops_lab (main)
$ ls
git_commands.pdf  intro.txt  oops1.java
ShwetaPC@shweta MINGW64 ~/Desktop/oops_lab (main)
$ git add .
warning: in the working copy of 'intro.txt', LF will be replaced by CRLF the next time Git touches it
ShwetaPC@shweta MINGW64 ~/Desktop/oops_lab (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    deleted:    git_commands.doc
    new file:   git_commands.pdf
    new file:   intro.txt
ShwetaPC@shweta MINGW64 ~/Desktop/oops_lab (main)
$
```

✓ 5. git add <filename>

Purpose: Adds a file to the staging area.

→ Prepares the file to be committed (saved).

Example:

git add main.cpp

✓ 6. git commit -m "message"

Purpose: Saves the staged changes to the Git repository with a message.

→ Each commit is like a snapshot of your project.

Example:

git commit -m "Added login feature"

✓ 7. git branch -M <branchname>

Purpose: Renames the current branch.

→ Useful for renaming master to main or creating a custom name.

Example:

```
git branch -M main
```

✅ 8. Connecting with a remote repo

Purpose: Links your local repo to a remote repo like GitHub.

Steps:

```
git remote add origin <repo-url> # Sets the remote repo
```

```
git push -u origin main # Pushes code to remote branch
```

→ origin is the nickname for your remote.

→ Use main or your current branch name.

✅ 9. pwd

Purpose: Prints the current working directory (folder path).

Example:

```
pwd
```

✅ 10. clear

Purpose: Clears the terminal screen.

→ Makes the terminal cleaner and easier to view.

Example:

```
clear
```

✅ 11. whoami

Purpose: Displays your username on the system.

Example:

```
whoami
```

✅ 12. vim / nano

Purpose: Opens a terminal-based text editor.

→ vim: Powerful but advanced

→ nano: Simple and beginner-friendly

Examples:

nano notes.txt

vim code.java

✓ 13. **cp** <source> <destination>

Purpose: Copies files or folders.

Example:

cp file.txt backup.txt

✓ 14. **cd** <foldername>

Purpose: Changes your current directory.

Example:

cd Desktop

✓ 15. **cat** <filename>

Purpose: Displays the contents of a file.

Example:

cat readme.txt

✓ 16. **rm** <filename>

Purpose: Deletes a file.

Example:

rm old.txt

✓ 17. **mv** <source> <destination>

Purpose: Moves or renames a file or folder.

Examples:

mv file.txt new_folder/ # Move

mv oldname.txt newname.txt # Rename

✓ 18. **rm -r** <foldername>

Purpose: Deletes a folder **and** all its contents recursively.

✅ 19. git push

Purpose: Uploads your local commits (changes) to a remote repository (like GitHub).

- Sends committed code from your local branch to the remote branch.
- Commonly used **after** git commit.

Example:

```
git push <remote-name> <branch-name>
```

◆ Most common form

```
git push origin main
```

- origin: the name of your remote (default for GitHub).
 - main: the branch you're pushing to.
-

◆ First-time push (after setting remote):

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

- -u sets upstream so future pushes can be done with just git push.
-

◆ After this, you can simply do:

```
git push
```

- Git remembers the branch and remote for next pushes.

✅ 20. git log

Purpose: Shows the **history of commits** in your Git repository.

- Helps you see who made what changes and when.
-

◆ Basic Usage:

```
git log
```