File Operations & Command Reference

AiCE Software Engineering Foundations — The Developer Environment

You can now find your way around, create folders and files, and read what's in them. This final Lesson in the set covers the commands that move things, duplicate things, and — permanently — remove things: cp, rm, mv, and clear. It closes with a full reference table of every command covered across this unit.

1. Copying Files and Directories: cp

To copy a file from one location to another, use the cp command:

```
cp manifesto.txt AfricaTechHub/
```

This copies manifesto.txt into the AfricaTechHub directory. The original file stays exactly where it was — cp duplicates, it does not move.

Trying to copy a directory with plain cp will fail with an error. To copy a directory — and everything inside it — you need the -r flag, which stands for recursive:

```
cp -r AfricaTechHub BackupHub
```

This creates a complete copy of AfricaTechHub, including all its contents, under the new name BackupHub. The -r flag tells cp to step inside the directory and copy everything it finds there, one level after another.

Command	Purpose	Works On
cp file.txt destination/	Copy a file	Files only
cp -r folder/ destination/	Copy a directory	Directories only

2. Deleting Files and Directories: rm

To delete a file, use the rm command:

```
rm welcome.txt
```

The file is deleted immediately — there is no confirmation dialog and no trash can to recover it from. Unlike a graphical interface, where deleted files typically go to a Recycle Bin first, anything removed with rm in the terminal is gone for good.

> **Warning — rm Is Permanent**
> *Before running rm, double-check that you are deleting the right file. There is no recovery. Many experienced developers make a habit of running ls first to confirm exactly what's there before they run rm. Treat this command with real caution.*

Directories cannot be deleted with plain rm either — they require the -r flag:

```
rm -r BackupHub
```

This deletes the directory along with every file and subdirectory inside it. Double-check before pressing Enter.

Command	Target	Can It Be Undone?
rm filename.txt	A single file	No
rm -r directoryname	A directory and all contents	No

3. Moving and Renaming: mv

The mv command moves a file or directory from one location to another. Unlike cp, the original disappears from its source location once the move completes:

```
mv manifesto.txt AfricaTechHub/
```

After this command, manifesto.txt no longer exists in its original location — it now lives inside AfricaTechHub.

mv also works on directories, and unlike cp, it needs no -r flag at all:

```
mv AfricaTechHub ArchivedProjects/
```

There is a second, less obvious use of mv that many beginners don't discover right away: renaming. When you “move” a file to the same directory but give it a different destination name, you've effectively renamed it:

```
mv old_name.txt new_name.txt
```

This is how renaming works in the terminal — there is no separate, dedicated rename command.

4. Clearing the Screen: clear

As you work, the terminal fills up with old commands and their output. When you want a clean screen without closing the terminal window, type:

```
clear
```

The screen clears, but you remain in the same directory, and your command history is preserved — you can still scroll up to see everything you typed before. Visually, though, you start fresh.

5. Full Command Reference

Every command introduced across this unit, in one place:

Command	Description
pwd	Print the full path of your current working directory
ls	List files and directories (non-hidden)
ls -a	List all files, including hidden ones
ls dirname	List contents of a directory without entering it

Command	Description
cd dirname	Change directory (move into a directory)
cd ..	Go up one level to the parent directory
mkdir name	Create a new directory
echo "text" > file.txt	Create a file with content
echo -e "line1\nline2" > file.txt	Create a multi-line file
cat file.txt	Read and display a file's content
cp file.txt dest/	Copy a file to a destination
cp -r dir/ dest/	Copy a directory to a destination
rm file.txt	Delete a file permanently
rm -r dirname	Delete a directory permanently
mv file.txt dest/	Move a file to a new location
mv oldname.txt newname.txt	Rename a file
clear	Clear the terminal screen

Key Takeaways

cp duplicates, mv relocates (and can rename along the way), and rm removes — permanently, with no safety net. Directories always need the -r flag with cp and rm, but never with mv. Together with the commands from the previous two documents, you now have a complete, working vocabulary for organizing a project from the terminal alone, without ever touching a mouse.