**What is Software Engineering?**

Imagine building a **robot** 🤖 or creating an app like **Instagram**!

Software Engineering is all about:  
🔹 **Writing instructions** (called **code**) for computers.  
🔹 **Building apps, games, and websites** that make our lives easier.  
🔹 **Solving real-world problems** through technology.

**What Do Software Engineers Do? 🛠️**

Think of software engineers as **architects** and **builders**! 🏗️ Here’s what we do:

1. **Design** apps and systems that users love. 📱
2. **Write code** that makes things work smoothly. 👨‍💻
3. **Test and fix bugs** (nobody likes bugs 🐛 in their apps, right?).
4. **Collaborate** with a team to create amazing stuff. 🤝

**The Magic Tools of a Software Engineer 🧰**

Here are some cool tools you'll learn to use:  
🔹 **Programming Languages** like Python 🐍, Java ☕, and JavaScript ⚡  
🔹 **Code Editors** like VS Code 🔤, where you’ll write your magic spells (code)!  
🔹 **Version Control** (Git 🦸) to keep track of all your changes.

**Meet Some Cool Programming Languages! 🌟**

💡 Programming languages are how we talk to computers.  
Here are some fun ones you’ll get to meet:

* **Python** 🐍: Simple and super beginner-friendly.
* **HTML, CSS & JavaScript** ⚡: For making websites interactive.
* **Dart & Flutter, Java** ☕: Powerful for building mobile apps!

**Welcome to CLI & Version Control Adventure! 🚀**

👋 **Hey there, Future Coders!**  
Today, we're diving into the **Command Line Interface (CLI)** and **Version Control** systems like **Git** & **GitHub**! You’ll learn to control your files like a pro ninja 🥷 and collaborate like a tech wizard 🧙‍♂️.

**What is the CLI? 🖥️**

💡 The **Command Line Interface** (CLI) is like chatting directly with your computer in its own language!  
While you can point and click with a mouse, using CLI lets you **type commands** to get things done **super fast**! 🎯

**Windows Terminal Commands 🪟**

Let's start by **creating, renaming, copying, cutting, and moving files** in the Windows terminal.

**Creating a Word Document:**

In Windows, you can create a file using the echo command:

echo "This is a Word document" > my\_document.docx

This creates a file named **my\_document.docx**!

**Renaming a File:**

Use the ren command to rename files:

ren my\_document.docx new\_document.docx

**Copying a File:**

To copy files, use the copy command:

copy new\_document.docx backup\_document.docx

**Cutting (Moving) a File:**

To move a file (cut it from one place and paste it elsewhere):

move new\_document.docx C:\Documents\Moved\_Document.docx

**Linux Terminal Commands 🐧**

Now let’s talk Linux!  
Here’s how to **create, rename, copy, cut, and move files** in Linux.

**Creating a Word Document:**

You can use the touch command to create an empty Word document:

touch my\_document.docx

**Renaming a File:**

To rename a file in Linux, use the mv command:

mv my\_document.docx new\_document.docx

**Copying a File:**

To copy a file, use the cp command:

cp new\_document.docx backup\_document.docx

**Cutting (Moving) a File:**

For cutting (moving) files, use the same mv command:

mv new\_document.docx ~/Documents/Moved\_Document.docx

**Mac Terminal Commands 🍏**

On Mac, the commands are quite similar to Linux, but let’s review them with a fun twist!

**Creating a Word Document:**

Use touch to create a new file:

touch my\_document.docx

**Renaming a File:**

To rename a file, again use the mv command:

mv my\_document.docx new\_document.docx

**Copying a File:**

To copy a file on Mac:

cp new\_document.docx backup\_document.docx

**Cutting (Moving) a File:**

And finally, to move a file:

mv new\_document.docx ~/Documents/Moved\_Document.docx

**What is Git? 🤔**

**Git** is a **version control system** that helps you track changes to your files (especially code!) over time.  
Think of Git as a **time machine** ⏳ for your project—if you make a mistake, you can go back and fix it!

**What is GitHub? 🌐**

**GitHub** is a cloud-based platform where developers can store and manage their Git repositories.  
It’s like **social media for code**—you can collaborate, review, and share your projects with the world! 🌍

**How Git and GitHub are Related? 🔗**

* **Git** runs locally on your machine, managing your project versions.
* **GitHub** stores these versions online so you can share your project, collaborate, or back it up safely.  
  Together, they make sure you never lose track of your awesome work! ✨

**Git Commands – Let’s Get Our Hands Dirty! 🧑‍💻**

Here are some **basic Git commands** you’ll use every day:

**1. Git Configuration:**

Set up your Git with your name and email (so Git knows who’s making changes):

git config --global user.name "Your Name" git config --global user.email "you@example.com"

**2. Initializing Git in a Project:**

To start tracking files in a folder:

git init

**3. Encrypting Git (SSH Key Setup):**

For secure access to GitHub, set up SSH:

ssh-keygen -t rsa -b 4096 -C "you@example.com"

**4. Adding Files to Git:**

Tell Git which files to track:

git add .

**5. Committing Changes:**

Save a snapshot of the current version of your files:

git commit -m "Add awesome new feature"

**6. Cloning a Repository:**

To download a project from GitHub:

git clone https://github.com/username/repository.git

**7. Creating a Branch:**

Want to try something new without messing up your main project? Create a **branch**!

git branch new\_feature

**8. Switching Between Branches:**

Move to a different branch:

git checkout new\_feature

**9. Merging Branches:**

Once your changes are perfect, merge them back into the main project:

git merge new\_feature

**10. Forking a Repository:**

To make a copy of someone else's project in your own GitHub account (useful for collaboration):

1. Go to the GitHub repository you want to fork.
2. Click the **Fork** button (on GitHub).

**11. Pushing to GitHub:**

Send your committed changes to GitHub:

git push origin main