**Slide Content: Introducing Git Command Line and GitHub Desktop**

**Slide 1: Why Learn the Git Command Line?**

* **What is the Command Line?**
  + A text-based way to interact with your computer and tools like Git.
  + Think of it as giving direct instructions instead of clicking buttons.
* **Why Use It?**
  + Full control over Git’s features.
  + Works everywhere (no need for a graphical interface).
  + Foundation for understanding GitHub Desktop’s magic.
* **Our Plan Today:**
  + Learn key commands first.
  + Then see how GitHub Desktop simplifies them.

**Slide 2: Setting Up the Command Line**

* **Windows:** Use Git Bash (installed with Git) or Command Prompt.
* **Mac/Linux:** Open Terminal (built-in).
* **Check Git is Installed:**
  + Type git --version and press Enter.
  + You should see something like git version 2.39.2.
* **Quick Tip:** Use cd to navigate folders (e.g., cd my-project).

**Slide 3: Essential Git Commands**

* **Initialize a Repository:** git init
  + Creates a new Git project in your folder.
* **Add Files:** git add <filename> or git add . (all files)
  + Prepares changes for a commit.
* **Commit Changes:** git commit -m "Your message"
  + Saves a snapshot of your work.
* **Check Status:** git status
  + Shows what’s changed or staged.
* **Push to GitHub:** git push origin main
  + Sends your commits to the remote repository.

**Slide 4: Transition to GitHub Desktop**

* **Why GitHub Desktop?**
  + Same power, less typing.
  + Visual interface for commits, pushes, and more.
* **How It Works:**
  + Command line: git add . → Desktop: Click “Stage All.”
  + Command line: git commit -m "Update" → Desktop: Enter message, click “Commit.”
  + Command line: git push → Desktop: Click “Push Origin.”
* **Next Step:** We’ll try both methods hands-on!