Mastering Integration and Version Control

Session Duration: November 20th

Subtext: An Introduction to Git and GitHub

What is Version Control

- **Version control** is a system that helps track changes in files over time, enabling collaboration and preventing conflicts in a project.
- **Why It Matters?**
  - Keeps a history of all changes.
  - Allows collaboration among multiple developers.
  - Makes it easy to revert changes.
  - Facilitates branching and merging.

Examples: Git, Subversion (SVN), Mercurial, etc.

What is Git?

Git is a distributed version control system for tracking changes in source code during software development.

Why Git?

- **Speed and performance.**
- **Distributed model (local and remote repositories).**
- **Branching and merging made easy.**

Fun Fact: Git was created by Linus Torvalds, the creator of Linux.

Key Git Commands to Know

Command	Purpose
git init	Initializes a new Git repository.
git clone	Clones an existing repository.
git add	Adds files to staging.
git commit	Saves changes to the repository.
git status	Displays the state of the repo.
git push	Pushes changes to a remote repo.
git pull	Fetches and merges changes from a remote repo.
git branch	Creates or lists branches.
git merge	Combines branches.
git log	Shows commit history.

Configuring Git

Step 1: Setting Up Git

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

Step 2: Checking Configuration

```
git config --list
```

Introduction to GitHub

What is GitHub?

GitHub is a cloud-based hosting service for managing Git repositories.

Why Use GitHub?

- Collaborate on code with others.
- Host and manage repositories.
- Share and showcase projects.
- Integration with tools and CI/CD pipelines.

Core Features:

- Pull Requests.
- Issues.
- Actions for CI/CD.
- Project Boards.

Working with Remotes

Remote repositories are versions of your project stored online.

```
git remote add origin <URL>
git remote -v
git push origin main
git pull origin main
```

GitHub Workflow

The Basic Workflow:

- Clone a repository.
- Make changes locally.
- Add and commit changes.
- Push changes to the remote repository.
- Collaborate using branches and pull requests.

Exercises

- Basic Git Commands:
  - Initialize a repository, create a file, add, commit, and check the status.
- Configuring Git:
  - Set up your name and email.
- Create a Remote Repo:
  - Link a local repo to GitHub and push your changes.
- GitHub Workflow:
  - Create a feature branch, make changes, and submit a pull request.

Collaboration Tips

- Always pull changes before pushing.
- Commit often with meaningful messages.
- Use branches for new features or fixes.
- Review pull requests before merging.

Knowledge Sharing

&

FAQs