OOPROG 21
Introduction to Version Control

-

References:
www.atlassian.com

Let's Get Started with GiT! (Version Control)

<u>Objectives:</u>

- Introduction to Version Control
- What is Git?
- Setting Up Git
- Basic Git Commands
- Working with Branches
- Collaborating with Remote Repositories
- Resolving Conflicts
- Common Tips and Best Practices
- Resources and Further Learning

Activities:

Activity No. 1 : Creating a GITHUB Repository

Deadline:	**Points:**	**Submission:**
August 14, 2024 (11:59pm)	10 Points	LMS Canvas

Instructions:

Create a personal *GitHub public repository* that will serve as a repository for this semester's activities, assignments and projects.

- Create a GitHub account -- name should follow this format "FirstnameLastname-UCM" (e.g. "CarlsanKim") **(Note: Avoid numbers, other special symbols and nicknames.)**
- Create your first repository and it should be in this format "ooprog21_2025"\
- The repository should contain *a folder called "Chapter_0" and "Chapter_1"*
- The Repository should be set to "*Public"*

Introduction to Version Control

- **What is Version Control?**
  - System to track changes to files over time.
  - Enables collaboration and keeps history of modifications.
- **Why Use Git?**
  - Open-source and widely used.
  - Allows multiple people to work on the same project.
  - Keeps track of changes, branches, and merges.

What is Git?

- **Definition:**
  - A distributed version control system.
- **Key Features:**
  - Local repositories.
  - Branching and merging.
  - History tracking.
  - Collaboration through remote repositories.

Setting Up Git

1. **Install Git:**
   - **Windows:** Download from Git for Windows.
   - **Mac:** Install via Homebrew (`brew install git`) or download from Git for Mac.
   - **Linux:** Install via package manager (`sudo apt-get install git` for Debian-based, `sudo yum install git` for Red Hat-based).
2. **Configure Git:**
   - Open terminal and set up your name and email:

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

Basic Git Commands

1. **Creating a Repository:**
   Initialize a new repository in a project directory

```
> git init
```

2. **Cloning a Repository:**
   Copy an existing repository:

```
> git clone “Repository Link”
```

3. Checking Repository Status:
View current changes

```
> git status
```

4. Adding Changes:
Stage files for commit

```
> git add “File Name”
```

Or (For staging all changes)

```
> git add .
```

5. Committing Changes:
Save changes to the repository

```
> git commit -m “Commit Message”.
```

6. **Viewing Commit History:**
   See the history of commits

```
> git log
```

Working with Branches

1. **Creating a New Branch:**
   Create a branch to work on new features:

```
> git branch branch-name
```

2. **Switching Branches:**
   Move to a different branch:

```
> git checkout branch-name
```

3. **Merging Branches:**
   Merge changes from one branch into another:

```
> git checkout main
> git merge branch-name
```

4. **Deleting a Branch:**
   Remove a branch when it's no longer needed:

```
> git branch -d branch-name
```

Collaborating with Remote Repositories

1. **Adding a Remote Repository:**
   Link your local repository to a remote one:

   ```
   > git remote add origin https://github.com/user/repo.git
   ```

2. **Pushing Changes:**
   Upload changes to the remote repository:

   ```
   > git push origin branch-name
   ```

3. **Pulling Changes:**
   Fetch and merge changes from the remote repository:

   ```
   > git pull origin branch-name
   ```

Resolving Conflicts

- **What are Conflicts?**
  - Occur when changes in different branches are incompatible.

- **Resolving Conflicts:**
  - Manually edit conflicting files.

    - Use `git add` to mark conflicts as resolved.
    - Commit the resolved changes.

Common Tips and Best Practices

- **Commit Messages:**
    - Be descriptive and concise.
- **Regular Commits:**
    - Commit often to avoid losing work.
- **Branching Strategy:**
    - Use branches for features and fixes.
- **Pull Before Pushing:**
    - Ensure you have the latest changes before pushing.

Resources and Further Learning

- **Official Git Documentation:** git-scm.com
- **GitHub Learning Lab:** github.com/lab
- **Interactive Tutorial:** learn-git-branching.js.org

Good Luck Aspiring Devs!
Your Journey Starts Here…