

# Lab 1: Git Flow, Merge, and Release

## Objective

- Understand and implement Git Flow: feature, release, and hotfix branches.
  - Experience merge conflicts and conflict resolution.
  - Build a simple user task management system using Java.
- 

## Logistics

- **Worth:** 2.5% of whole course.
  - **Due Date:** Week 2.
  - **TA Supervised:** During the tutorial session.
  - **Penalty:** 5% deduction for unexcused absences.
  - **Group Work:** Individual.
- 

## Instructions

### 1. Initial Repository Setup

1. Forks the provided repository.
  2. Create the following branches from `develop` :
    - `feat/register-user`
    - `feat/user-login`
    - `feat/add-task`
    - `release/1.0` : For preparing the release.
    - `hotfix/1.0.1` : For fixing a post-release bug.
- 

### 2. Detailed Task Assignments

#### Task 1 User Registration( `feat/register-user` )

- Implement `UserRegistration.java` to validate:
  - Username: 5-15 characters
  - Password: minimum 8 characters

- On successful registration, store users into a static `Map<String, String>` in `UserDatabase.java`.
- Modify `config.properties` to simulate a merge conflict:
  - Change `app.version`
- Push and create a pull request into `develop`.

### Task 2 User Login( `feat/user-login` )

- Implement `UserLogin.java` :
  - Validate login credentials using `UserDatabase`.
- Modify `config.properties` to simulate another conflict:
  - Change `app.environment`
- Push and create a pull request into `develop`.

### Task 3 Add Task Functionality ( `feat/add-task` )

- Implement `TaskManager.java` to:
  - Let each user create tasks ( `Task.java` with `title`, `description`, and `done` flag).
  - Store tasks using `Map<String, List<Task>>` (username to tasks).
- Optionally support command-line interaction.
- Push and create a pull request into `develop`.

### Task 4 Release Preparation ( `release/1.0` )

- Create `RELEASE_NOTES.md` summarizing new features.
- Tag the commit as `v1.0.0`.
- Merge `release/1.0` into `main` via PR.

### Task 5 Hotfix ( `hotfix/1.0.1` )

- Introduce a bug in `UserRegistration.java` (e.g., skip null check).
- On `hotfix/1.0.1`, fix it and merge into both `main` and `develop` via PRs.

### Task 6 Simulating Merge Conflicts

- Use `config.properties` for simulated merge conflicts in both `feat/register-user` and `feat/user-login`.
- Resolve during merge into `develop`.

## 5. Deliverables

Place the following files under the `deliverables/` directory in your repo:

- `GitWorkflowDiagram.png` – shows branch/merge flow with annotations
- `Tasks.md` – screenshots of task outputs (registration, login, task addition)
- `RELEASE_NOTES.md` – written during release branch work

---

## Evaluation Criteria

1. Git workflow diagram - 0.6 marks, show correct branch creation/merging/conflicts
  2. Feature Implementation - 0.6 marks, Java files for registration, login, task management
  3. Merge Conflict Resolution - 0.5 marks, resolve simulated conflict in config.properties
  4. Release and Hotfix Process - 0.5 marks, tagging, merging release & hotfix into correct branches
  5. Deliverables - 0.3 marks, screenshots and markdown files provided
-