

# EE368 Lab 1 - Github for Collaborative Development

## Objective:

Students will be able to use Git and GitHub for their teamwork in this course. Students should know about the Github workflow and how to collaborate with other developers in their projects.

## Prerequisite:

- Students should have a **GitHub** account
- Students should have **Git** installed
- Students should know Git Basics, Git basics can be found [here](#)

## Procedure:

### 1. Forking a repo and adding a collaborator

- You should fork [this](#) repository, then add the other person as a collaborator on GitHub
- Clone the repository on your local machine

### 2. Setting up Kanban board and creating issues

- On GitHub, under the **Projects** tab, one person creates a new project. Select **Automated Kanban** as the template
- Create issues, one for each of the bugs found in the **Hello.java** file.
- Add the issues to the Kanban to-do board
- Assign issues, one for each team member
- Link the issue to the newly created project

### 3. Branching, committing, and pushing

- In your local machine, create a new branch with a relevant name to the issue you are addressing (e.g. “fix-spelling”, “add-feature”)
- Fix the issue/bug that you were assigned, then add and commit the changes
  - ◆ Commits should start with a verb, and what it does to the codebase (e.g. “Remove quote”, “Fix issue #101” )
- Push the branch to the remote repository

#### 4. Writing pull requests

- On GitHub, create a pull request to merge the changes from the branch you have just pushed to the main/master branch.
  - ◆ Make sure it is to the main/master of your own forked repository! By default, Github selects the original repository instead of your own forked repository
  - ◆ Name your pull request appropriately
  - ◆ In the description, describe what changes have been made to address the issue, and how have the changes been tested
- Assign the issue you created to the pull request
- Request a partner for a code review!

#### 5. Doing code reviews and resolving merge conflicts

- Review your partner’s code (use the review change button). Approve changes if it looks good
- Once your **PR** has been reviewed, merge your changes! Resolve any merge conflicts that arise accordingly
- Return to the Kanban board, and if you have done things right the issue should automatically move into **Done**