

Branches

- **main** → stable, working version only.
- **dev** → integration branch where features get merged after review.
- **feature branches** → each member codes on their own branch, then creates a Pull Request (PR).

Example names:

- `feature-input-validation`
- `feature-user-story1`
- `feature-admin-update`

Commits

- Use clear commit messages tied to tasks:
 - `feat: add email validation`
 - `fix: correct password regex`

Pull Requests

Every PR should:

1. Be reviewed by at least one other teammate before merging into dev, and 4 other teammates before merging into main.
2. Reference the **Task or User Story** it addresses.
3. Include testing notes.

Screencasts & Standups

- Save recordings in `/media/` or a shared cloud folder.
- Link them in the `README.md`.

Remember:

- **Never push directly to `main`.**
- **All changes must flow:** `feature` → `dev` → `main` or `media` → `main`

Getting started

In a directory of your choice clone the git repo

```
git clone git@github.com:codydjohnston/cse-360-group12.git
```

You can also use github desktop

Remember: You must be logged into github in vscode, on the github desktop app, or have your device ssh key added to your github account.

Once you have the repo cloned, copy the files from

```
cse-360-group12/src/FoundationCode/src/
```

To the your src folder from homework 1

For me that would be

```
c:\users\cody\ASU-cse-360\FoundationCode\src\
```

Note there should be a more elegant way to do this, I cant get eclipse to work out of my git repo for some reason. Until we find a better way we will just have to copy files back and forth between the workspace and the git repo.

After copying the code into eclipse click file > refresh and project > clean

Branching and Pull Requests

Make sure you're up to date

git checkout dev

git pull origin dev

Create a feature branch

```
git checkout -b feature-input-validation
```

Do your coding work

Edit files in your Eclipse/VS Code workspace (src/...).

Test locally to make sure it works.

Stage and commit your changes

```
git add src/LoginValidator.java  
git commit -m "feat: add basic email format validation for login"
```

Push (or publish) your branch to GitHub

```
git push origin feature-input-validation
```

Open a Pull Request

Go to GitHub

GitHub will show a prompt: "You just pushed a branch — open a Pull Request?"

Open a PR with:

```
Base branch: dev  
Compare branch: feature-input-validation  
PR title: feat: add login input validation  
Description:  
  Addresses User Story #3: Login validation  
- Added regex email validation  
- Added error messages for invalid inputs  
Testing Notes:  
- Enter invalid emails → app blocks submission  
- Enter valid emails → app continues
```

Review & merge

At least 1 other teammate reviews before merging into dev.

After testing, the team lead (or assigned reviewer) presses Merge Pull Request.

Later, once several features are complete and reviewed, dev can be merged into main with **4 reviews** required.

Clean up

```
git branch -d feature-input-validation
git push origin --delete feature-input-validation
```

This example shows the exact flow we expect:

feature-input-validation → dev → main

Branching Off of dev

To start a new feature or task, you should create your branch from the latest dev branch. This ensures you're working with the most up-to-date integration code.

Steps:

```
# Make sure you are on dev
git checkout dev

# Pull the latest changes from the remote dev branch
git pull origin dev

# Create a new feature branch off dev
git checkout -b feature-branch-name
```

Notes:

`git checkout dev` → switches you to the dev branch (the branch you will branch from).

`git pull origin dev` → updates your local dev branch with the latest commits from the remote repository.

`git checkout -b feature-branch-name` → creates and switches to a new branch in one command.

All work for that feature should now happen on this new branch.

Example:

```
git checkout dev
git pull origin dev
git checkout -b feature-login-validation
```

This creates a new branch feature-login-validation that is up to date with dev and ready for work.