**START OF DAY**

Start LOD machine

**Right click on desktop and configure display settings**

**Configure Path for Maven:**

Setting the path of Maven in environment Variables: Search the Environment Variable --> Edit the System Environment variables--> Navigate to Advanced tab --> Environment Variables

i) MAVEN\_HOME : Click New --> Variable Name : MAVEN\_HOME , Variable Value: C:\Program Files\apache-maven-3.5.2

ii) M2\_HOME : Click New --> Variable Name : M2\_HOME , Variable Value: C:\Program Files\apache-maven-3.5.2

iii) Edit the 'Path' Environment Variable --> %M2\_HOME%\bin

Testing whether Maven is installed: mvn -version

**Configure Git**

In GitBash or Windows Command Prompt or PowerShell:

git config --global user.name "Your Name"

git config --global user.email "mail@db.com

**Open IntelliJ.**

Create “working” folder and move into it (cd)

Use Git to pull latest versions of code from GitHub

git clone https://github.com/YOURNAME/YourRepo.git

cd newlycreatedfolder

git fetch --all

git checkout branch-name

**END OF DAY**

How would I merge changes made to a local branch called "reference" back to a master repository called "myGitHubRepo" assuming nothing in the master repository has changed?

To merge changes from a local branch named "reference" back into the master branch of the remote repository "

myGitHubRepo," assuming nothing has changed in the master repository since you last pulled, you can follow these steps:

1. Make sure you are on the "reference" branch:

git checkout reference

1. Fetch the latest changes from the remote repository:

git fetch origin

1. Ensure that your "reference" branch is up to date with the latest changes from the remote "myGitHubRepo" master branch:

git rebase origin/myGitHubRepo

This step is essential to ensure that your "reference" branch is based on the latest version of the master branch. If there are conflicts during the rebase, resolve them, and continue the rebase process.

1. Switch to the "myGitHubRepo" branch:

git checkout myGitHubRepo

1. Merge the changes from the "reference" branch into "myGitHubRepo":

git merge reference

At this point, Git will attempt to perform a fast-forward merge because the "reference" branch already includes all the changes from the "myGitHubRepo" branch (assuming nothing has changed in the "myGitHubRepo" branch since you last pulled).

1. Push the merged changes to the remote "myGitHubRepo" repository:

git push origin myGitHubRepo

REPEAT THE ABOVE FOR EVERY REPO WORKED ON