How to set up the program/work environment (eclipse)

1. Download Eclipse IDE for Java EE (32 bits or 64 bits are fine. If you're unsure, pick 32 bits)

2. Unzip the file, put the eclipse folder somewhere you'll remember.

3. Create a shortcut to desktop, if you want to.

4. Open eclipse.exe (the one with purple circle icon)

5. If it's your first time, Eclipse will ask for the workspace, pick a folder you want your work to be.

6. On the top menu, click help -> eclipse marketplace

7. Search for "EGit" and enter

8. You'll see "EGit - Git Team Provider". Click install

9. Click next, next, confirm, accept agreement, next, next, etc.

10. Restart Eclipse

------------------------------------------------------------------------------------------------------

Set up the project in Eclipse

1. Click File -> import

2. Find git -> project from Git (alternatively you can search for git). Click next.

3. Click clone URI, then next.

4. Fill in the following:

URI: https://github.com/WAMIQP/WAMIQP

User: WAMIQP

Password: wamiqp2014

Store in Secure store: yes...? (I believe this is like "remember me" thing.)

5. Click next

6. Set the directory to be any directory you want. I recommend using a folder in your own computer, not your school drive/folder)

7. Make sure "import existing project" is checked, click next

8. Make sure WAMIQP is checked, then finish.

----------------------------------------------------------------------------------------------------

Git Commit/Fetch/Push/Merge aka Upload/Download

\*\*\*\*\*\*\*\*For all “push” and “fetch” here, I mean “push to upstream” and “fetch from upstream”\*\*\*\*\*\*\*\*\*\*

1. Commit = Make your changes official

2. Fetch = Get commits you don't currently have from the online repository (master)

3. Push = Push all the commits up to the online repository (master) aka updating it

4. Merge = Merge people's \*commits\* together. Un-committed changes usually result in an incomplete merge.

All four things here can be accessed by:

Right click the project -> team -> \_\_\_\_\_\_\_\_

Before you work on something

1. Fetch and merge. Make sure that your version is the same as the online repository (master)

When you want to upload things

1. Commit. Make all of your changes official. Make sure that you check all the files and type meaningful commit message.

* If you had things you don't want other people to see them, put in your folders and commit them too. Think of it as backup.
* If you edit somebody else's work, you shouldn't have. If you really want to work around somebody else's work, copy them somewhere, say your directory, and then work on it.
* If it's something you really want other people to see (racist jokes? personal info?), I don't think it should be in the project folder.

1. Fetch. Make sure your "base" version is the newest
2. Merge. Put your changes and new changes together. If you follow 1), there should not be any conflict. If unresolved conflicts occur (either you violated 1) or I missed something), please refer to <http://wiki.eclipse.org/EGit/User_Guide#Resolving_a_merge_conflict> or just contact me. Resolving conflicts usually takes longer than making changes.
3. Push to upload them.

In layman's term, this is how things work

* Think of the project as jigsaw pieces. If you did not violate 1), you'll just keep adding more pieces to them.

1. Commit. When you're sure that the piece is right, you put them down on \*your\* scrap paper/notebook.
2. Fetch. You want to know if there are any other pieces other people put on the board while you're working/thinking
3. Merge. You combine other people's new pieces on your scrap paper/notebook to see if there is any conflicts (if you do not change somebody else's pieces, you should be fine, right?)
4. Push. Once everything is all set, you update the board with your current version, which includes your changes.

Note: On the project name on the left. You might see something like

WAMIQP [WAMIQP master ^5 v6]

^ = up arrow

The number after ^ means the number of commits that you have made official and you have not pushed them up. Note: the number of commits = number of times you commit, not the number of files you changes

v = down arrow

The number after v means the commits from the online repository that have \*not\* been merged with your current copy.

Errors you may get:

1. Non-fast forward = either your base version is not the newest or you forgot to merge
2. Not permitted = maybe your users/passwords are wrong. To change,

Right click on project -> team -> remote -> configure push to upsteam

On the same row as URI, click change. Type in user and password, check “store in secure store”

1. Conflicts occurs. Easiest way to solve this is:
2. Copy your changes/files somewhere else (not in the project)
3. Right click the project -> team -> Reset
4. Change reset type to Hard
5. Copy your work back in. Then commit and so on.

Note that HARD reset will reset everything to the same version as the online repository.

NOTE: because the projects are, in fact, a bunch of patches (similar to jigsaws), pretty much all the changes are reversible and we can actually went through the list of commits to see who violate 1). I hope I do not have to do that. (If you want to know how, ask me or Google it).