GitHub Tutorial

GitHub Exercise Flow (How to Go Through Tutorials for Exercise):

1. Joining GitHub
    - Sign up for GitHub
2. Creating a Repository
    - Create and name your Repository
    - Name it something relevant but different from mine to not confuse you
3. Sharing Access to Repository
    - Share repository with teammates
4. Display Git Browser
    - Open NetBeans and make sure you do this every time to view the GitHub File Directory
    - A critical window for managing Branches
        - We aren't making branches in the exercise, but we are likely to for the final project
5. Cloning a Repository
    - Clone mine first
6. Importing a Repository Project
    - Make sure to select the appropriate "src" folder
7. Cloning a Repository
    - Clone your repository
8. Creating a new Repository Project
    - Name the project as you wish. However, make sure everyone names the project the same to keep the files together
9. Copying Over Files
    - See above tutorial to move my files to your repository
10. Pulling from Repository
    - Connects your repository to NetBeans officially
11. Adding and Committing Changes
    - Stage your files to be uploaded
12. Pushing Your Changes
    - Add the base files to your repository

For those that wish to not do this process, they may wait until step 11 is completed.

1. Cloning a Repository
    - Clone the group's repository which has the new files
2. Importing a Repository Project
    - Make sure you select the files
3. <Add Code to Program>
    - Time to code your project

Once everyone has made their changes, each person will push separately onto your group repository

1. Adding and Committing Changes
    - Stage your files to be uploaded
2. Pushing Your Changes
    - Add the base files to your repository

Once all members have updated their file, have everyone pull the new files. You may need to have all 3 files on your repository already. The first person to copy over the files to your repository should have all 3, so they can test the program.

1. Pulling from Repository
    - Get all changes to every file in your project
2. <Make sure the project works>
    - Make sure there are no syntax errors

<u>GitHub Project Flow (How to Go Through Tutorials for Project):</u>

1. Create a Repository
2. Sharing Access to Repository
3. Cloning a Repository
4. <Initiate Repository>
    a. Creating a new Repository Project (1 person)
    b. Importing a Repository Project (Others)

Repeat these until finished

5. Pulling from Repository
6. <Code Project>
7. Adding and Committing Changes
8. Pushing Your Changes

<u>Additional Information:</u>

- Make sure that you don't change the file names, as each file initiates an object using the original file name
- Make sure to follow the steps in order, as attempting to do multiple at the same time will mess up the process and could lead to multiple complications
    - E.G. Multiple group repositories, Sharing issues, file naming issues, project naming issues

If you have any other questions on things I may have not covered, please email me and the professor. There are many issues and problems coming up that I did not run into during testing as I didn't have another individual with me to go through with this process.

Creation of GitHub Repository:

Joining GitHub

- Go to https://github.com/join
- Enter your desired credentials
    - Use your Penn State email
- Press "Select a Plan"
- Select the free plan from the list
- Verify your account through the email sent to your Penn State email
- Go back to the page to select your plan and press "Continue"
- Enter the appropriate answers for the small questionnaire
- Press "Submit"
- You should now be at the homepage of your account

Creating a Repository

- Click the plus symbol with an arrow near the top right
- Select "New Repository"
- Enter the repository's name and description, and make sure "Public" is selected
- Also make sure that a README is created to contain the project's information
- Select "Create Repository"
- You will now be at the repository's homepage

Sharing Access to Repository

- For the owner of the repository
    - Open the repository homepage in GitHub
        - Same page as last step of previous process
    - Click on the "Settings tab"
    - Click on "Manage Access"
    - Click on "Invite a Collaborator"
    - Enter the individual's username
    - Select the individual listed and choose to confirm the invitation
    - Wait for the individual to accept the invitation
- For the person joining the repository
    - Wait for the owner to invite you to the repository
    - Open your emails, and view the email for invitation
    - Click on "accept or decline" link
    - Accept the invitation

Downloading Your Repository:

Displaying Git Browser

- This browser is critical to understand which branch you are working in and the layout of the local repository
- Open NetBeans and select the “Team” tab, hover over “Repository”, and then select “Repository browser”
- Place the browser in the position you’d like
- You now have a visual for the Git Layout within your local repository

Cloning a Repository

- Open NetBeans and go into the “Team” tab, select “Remote” or “Git”, and select “Clone”
- Enter the page URL of the repository homepage from the GitHub website into the directory text field
    - Enter GitHub username and password in appropriate fields
- Specify the folder in which you would like to save your repository to your computer
    - Put in different folder than java applications, or make a new folder used entirely for the repositories
- Click Next
- Make sure to select the master branch, and any other branches that are currently existing on your repository
- Click Next
- Nothing on this final window should need to be changed, just make sure that the checkbox “Scan for NetBeans Projects after Cloning” is selected
    - Scans for the projects in your repository if existent
- A new popup window will now appear asking if you would like to create a new project for the repository to take files from.

Creating a New Repository Project

- If you are the individual initializing the file names and beginning code, create the Java Project as normal
    - Name the program as you would like
- Save the program to the java application area you normally keep programs from NetBeans
- The program is now connected to the repository, and you are ready to begin the GitHub flow

Importing a Repository Project

- If you are importing java files from the repository that already exist, select create project from existing sources
- A new window will pop up asking questions about the imported project
- Name the project the name of the file you are importing (same name as file)
- Navigate to the local repositories folder you saved earlier when cloning the project, and then find & select the project folder

- Click Next
- Now, go to “Add Folder” for the source package folders and select the “src” folder that is in the project folder you selected in the previous step
- Once all the sources are added from your repository, click Next
- Click Finish
- Your files now contain the same code as the repository, so now you can begin the GitHub flow

<u>Managing Your Repository:</u>

Pulling from Repository

- Go on NetBeans and open the project that is connected to the GitHub repository
    - This process should be done every time you open your project in case a new version has been uploaded to GitHub
- Make sure the master branch is currently selected
    - Read "Switching Branches” Tutorial
- Pull updates by selecting the “Team” tab, hovering over “Remote”, and then select “Pull...”
- Make sure “Select Configured Git Repository Location” is selected
- Click “Next”
- Make sure “master -> origin/master” is selected
- Click “Finish”
- Your master branch has now been updated

Checking Out a Version

- Go on NetBeans and open the project that is connected to the GitHub repository
- Check previous versions by selecting the “Team” tab, hovering over “Checkout”, and then select “Checkout Revision”
- Click “Select”
- In the left field, select the branch you would like to checkout
- In the right field, select the version you want to load
    - Top version is latest, bottom version is oldest
- Click “Select”
- Click “Checkout”
- A new temporary branch will now be made to check the previous version you have selected
    - When done with the older code, switch back to the master branch and NetBeans will remove the temporary branch

Creating a Branch

- Go on NetBeans and open the project that is connected to the GitHub repository
    - Make sure to always pull from the repository to confirm you have the latest version
        - (Read “Pulling from Repository” Tutorial)
- Create a new branch off master by going to the “Team” tab, hovering over “Branch/Tag”, and then select “Create Branch”
    - Enter the name of the new branch (Doesn’t matter, but make it relevant)

- Click “Create”

Switching Branches

- Switch control to new branch by going to the “Team” tab, hovering over “Branch/Tag”, and then select “Switch to Branch”
- Select the desired branch
- Click “Switch”
- You will now be in your desired branch

<u>Updating Your Repository:</u>

Adding & Committing Changes

- This process occurs when you are finished adding all changes to your code, or have reached a critical milestone in the project’s progress
- Go to the “Team” tab and select “Add” to add the changes to the queue
- Now, go to the “Team” tab and select “Commit”
- In the bottom field, select the files you would like to save to your local repository
    - Add a description of the commit in the top field if desired (something relevant)
- Your changes have now been saved to your local repository data

Merging Your Changes to Master

- Once all teammates have agreed to your addition, you may begin to merge your side branch with the master branch
    - We keep your personal changes within a separate branch so that we can keep the master branch up to date without altering your current code
- Switch to the master branch
- Select the “Team” tab, hover over “Branch/Tag”, and select “Merge Revision”
- Click on the “Select” Button
- On the left field, make sure your side branch is selected
- On the right field, make sure the latest commit is selected
    - Latest commit should be at the top of the list (Look for your description)
- Click “Select”
- Click “Merge”
- Your master branch will now be merged with your side branch changes

Deleting A Side Branch

- Once the changes have been brought over to master, delete your working branch as there is no use for it anymore
- Right click the branch within your Git Repository Browser
    - Read “Display Git Browser” Tutorial
- Select “Delete Branch”

- Make sure the popup message warns you of deleting the correct branch, and then select "Yes/Ok"
- Your side branch has now been deleted

Pushing Your Changes

- Next, after the team has reviewed your class and are aware of your push, continue with this process to add your local repository changes to the GitHub Repository
- Select "Team", hover over "Remote", and then select "Push..."
    - You will select the master branch only
- Make sure "Select Configured Git Repository Location" is selected
- Click "Next"
- Make sure only "master -> master [U]" is selected
    - If you don't delete your side branch, it is possible to push your coding branch into the repository as well
        - (Useful for purposes outside the project)
- Click "Next"
- Make sure only "master -> origin/master [U]" is selected
- Click "Finish"
- Your master branch has now been uploaded to the direct GitHub repository, and you will be able to see these changes on the GitHub Repository site

Copying Over Files (Copy Files from My Repository to Yours)

- Make sure to clone both repositories to your NetBeans IDE
    - Read on "Cloning a Repository"
- Open the location of my Repository on your local computer
- Copy the files from my "src" folder
- Paste the files into your Repository into your source folder
- Make sure now that you are editing your repository files
- Your project now contains the files needed and is connected to your repository