**Github or the Obsolescence of Flash Drives**

**Step 1:** Account Setup and Installation

**Make Github Account (5 minutes)**

1. Go to <https://github.com/>
2. Sign up
3. Login

**Install Git (6 minutes)**

1. Go to <https://git-scm.com/downloads>
2. Select OS and download
3. Open the installation .exe
4. Click Next once
5. Make sure the “on Desktop” checkbox is checked (first Checkbox at top)
6. Click Next until you get to the Install button, change no other settings
7. Click Install

**Step 2:** All you need to know for using the Date Manager script

Preamble:   
Git is a service that allows you to store files in a remote server by “pushing” and retrieve those files from the server by “pulling”. This service is useful because you can update an entire project directory and immediately retrieve the most up to date changes of the project on every computer with access to your file store (called a repository). No more flash drives

**Starting Up (4 minutes)**

1. Open the Git Bash. This is a special command line git installed for you. There should be a shortcut to it on your Desktop. If not, open the Git installation folder in your Program Files and you will find it. Make a shortcut on desktop
2. Type cd ~/Documents
3. Type mkdir Github
4. Type cd Github
5. You will now be inside of the Github Directory where you can put all of your Github Projects (if you make any more)
6. You can see the full path to your directory by typing pwd
7. To get back to this location in the git bash, just type cd <pathname>  
   where <pathname> is the full path you saw from typing pwd
8. Type git config –global user.name “ENTER YOUR GITHUB USERNAME”  
    but actually type your user name in the quotes  
   i.e. git config –global user.name “jimmydings24”

(You only need to do this once and never again)

**Retrieving DateManager project the first time (30 seconds)**

1. (While in the Github Project Directory) Type   
   git clone <https://github.com/cheeordie1/DateManager.git>
2. You now have everything. Type cd DateManager  
   in order to enter the directory. Alternatively, open a file explorer and find the directory by following the path that pwd tells you
3. You do not need to do this again. This is how you setup an existing project the first time, by cloning it.

**Running the DateManager (3 minutes)**

1. Using a file explorer, find the DateManager Directory you cloned
2. Open HOW TO USE.txt and follow the directions. They look long, but you are just copying and pasting 3 path names
3. Once you’ve done everything in HOW TO USE.txt, your Date Manager is set up and you will receive a new TODO list in the form of a .xlsx every Monday and Thursday at 11:00AM
4. To receive a todo list immediately, double-click manage\_dates.vbs

**You’re Done, Beyond is how to use Git better**

(Actually you need to do the fist part of step 3 eventually when I finish DateManager)

**Step 3:** Using Git in the future (Not required)

**Retrieving most recent version of files (AKA pulling)**

1. Since you are using my repository, you need to request permission to pull. Tell me your username, and I can add you to the contributors of the project. Go to 2 if it is your own project.
2. Open a git bash and cd into the directory of your current project (i.e. the DateManager directory). For me this looks like

cd C:/Users/Nicholas/Documents/Github/DateManager

1. Make your own branch. A branch is a personal version of the project with your own work on it. It won’t affect other people’s work when you push changes from the local (on your computer) version of your branch to the remote (on a server somewhere) version of your branch or pull changes from the remote to the local  
     
   To do this, type git branch mike
2. Type git checkout mike
3. The first command created the branch called “mike” and the second switched you to that branch, so all your work only affects that branch. You don’t need to do steps 3-4 again for this project
4. Type git commit –a –m “This is a comment”  
   Commiting is like marking a point in history. Github will save a snapshot of your entire project for each commit you make, and you can revert or look at the files from that snapshot at any time. Always do this before pushing or pulling (it is required to work)
5. Type git pull origin master  
   This retrieves the most recent version of the master branch and puts it on your branch. Your branch should be up to date now.

**Saving all of your commits and project updates (AKA pushing)**

1. If you make a lot of changes to your project and commit the changes a bunch of times, you haven’t actually saved them in the cloud yet. To do this, you need to pull
2. In your project directory, commit your changes with  
   git commit –a –m “Here are some changes”
3. Type git push origin mike
4. Enter your username and password (password text will be invisible)
5. Now your remote cloud storage will save the data from your project branch and all of the commits you have made since the last time you pushed

**I may update this in the future to show you how to make your own projects so you can back everything up on cloud storage. Right now, you have everything you need for DateManager**

Cheat Sheet:

Pull most recent files from master

git commit –a –m “This is a dumb comment”

git pull origin master

Push updates to your branch

git commit –a –m “This is another dumb comment”

git push origin mike