|  |
| --- |
| MIS 478 |
| GitHub Instructions |
|  |
|  |
| **Brandon Casey** |
| **3/25/2015** |

|  |
| --- |
|  |

Contents

[Terminology 2](#_Toc415083238)

[First Setup (Only needs to be done first) 3](#_Toc415083239)

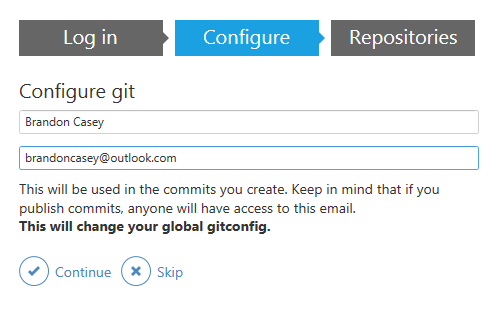
[Updating the Project 5](#_Toc415083240)

# Terminology

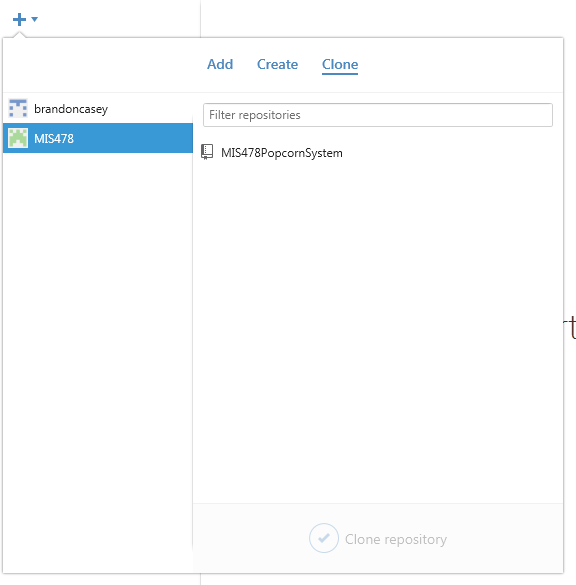
* **Branch -** A branch is a parallel version of a repository. It is contained within the repository, but does not affect the primary or master branch allowing you to work freely without disrupting the "live" version. When you've made the changes you want to make, you can merge your branch back into the master branch to publish your changes.
* **Clone -** A clone is a copy of a repository that lives on your computer instead of on a website's server somewhere, or the act of making that copy. With your clone you can edit the files in your preferred editor and use Git to keep track of your changes without having to be online. It is, however, connected to the remote version so that changes can be synced between the two. You can push your local changes to the [remote](https://help.github.com/articles/github-glossary/#remote) to keep them synced when you're online.
* **Commit -** A commit, or "revision", is an individual change to a file (or set of files). It's like when you *save* a file, except with Git, every time you save it creates a unique ID (a.k.a. the "SHA" or "hash") that allows you to keep record of what changes were made when and by who. Commits usually contain a commit message which is a brief description of what changes were made.
* **Fetch -** Fetching refers to getting the latest changes from an online repository (like GitHub.com) without merging them in. Once these changes are fetched you can compare them to your local branches (the code residing on your local machine).
* **Git -** Git is an open source program for tracking changes in text files. It was written by the author of the Linux operating system, and is the core technology that GitHub, the social and user interface, is built on top of.
* **Issue -** Issues are suggested improvements, tasks or questions related to the repository. Issues can be created by anyone (for public repositories), and are moderated by repository collaborators. Each issue contains its own discussion forum, can be labeled and assigned to a user.
* **Merge -** Merging takes the changes from one branch (in the same repository or from a fork), and applies them into another. This often happens as a Pull Request (which can be thought of as a request to merge), or via the command line. A merge can be done automatically via a Pull Request via the GitHub.com web interface if there are no conflicting changes, or can always be done via the command line. See [Merging a pull request](https://help.github.com/articles/merging-a-pull-request).
* **Pull -** Pull refers to when you are fetching *in* changes *and* merging them. For instance, if someone has edited the remote file you're both working on, you'll want to *pull* in those changes to your local copy so that it's up to date.
* **Pull Request -** Pull requests are proposed changes to a repository submitted by a user and accepted or rejected by a repository's collaborators. Like issues, pull requests each have their own discussion forum. See [Using Pull Requests](https://help.github.com/articles/using-pull-requests).
* **Push -** Pushing refers to sending your committed changes to a remote repository such as GitHub.com. For instance, if you change something locally, you'd want to then *push* those changes so that others may access them.
* **Repository -** A repository is the most basic element of GitHub. They're easiest to imagine as a project's folder. A repository contains all of the project files (including documentation), and stores each file's revision history. Repositories can have multiple collaborators and can be either public or private.

# First Setup (Only needs to be done first)

1. Prerequisites:
   1. Android Studio installed w/ 2.3.3 (Gingerbread) SDK installed
2. Go to <https://github.com/>
3. Create an account
   1. Message Brandon your username that you created so that he can add you to the project
   2. You may need to go to your email to verify your account
4. While you wait to be added to the group, download GitHub for Windows <https://windows.github.com/>
   1. It may need to install .NET 4.5 which takes a little bit of time
5. Once it gets installed then you will be taken to a login screen.
6. Log in with your credentials that you created above
7. The next step is to configure Git. Simply type your name in the first box and leave the 2nd box filled in with your email address.



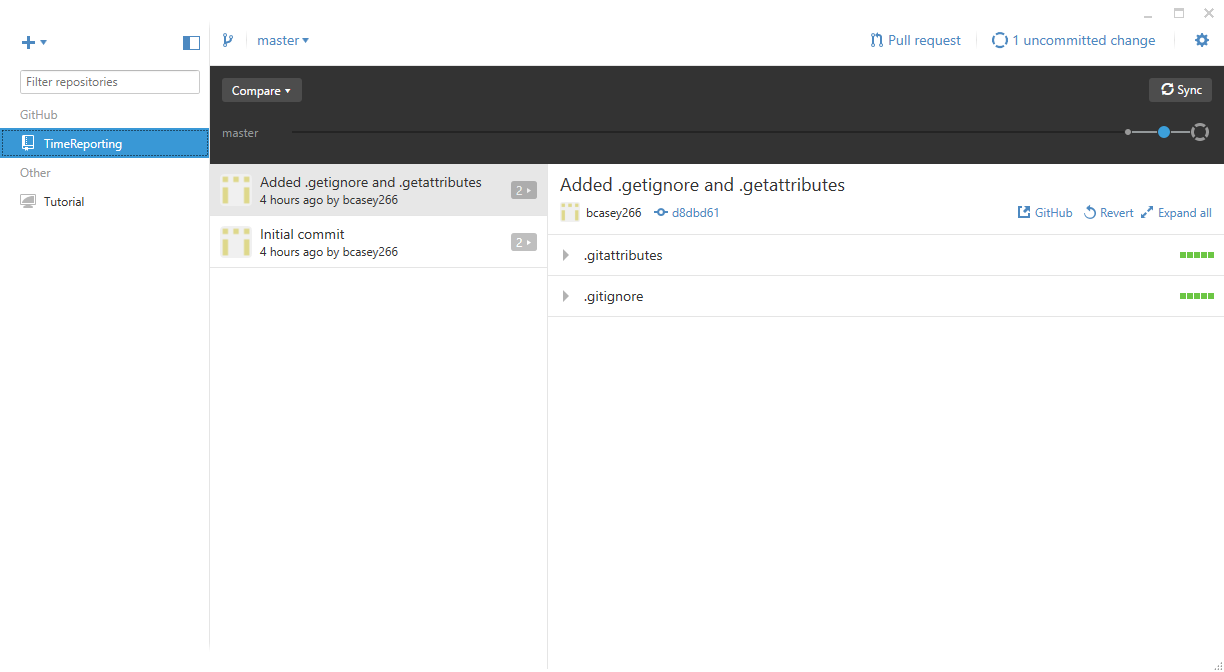
1. Click the Continue button at the bottom
2. (wait until you have been added to the project by Brandon before continuing)
3. Click on the plus sign in the top left of the window
4. Now click on the clone option
5. “MIS499” should be listed
6. Click that and then select “TimeReporting”



1. Select “Clone TimeReporting” at the bottom
2. Select the location where you want the project to be stored locally.
   1. For this example we will assume it is under C:\Users\Brandon\Documents\GitHub
3. Everything should be set up now for the first time use

# Updating the Project

1. Open the GitHub for Windows
2. At the top right of the screen, select “Sync”
3. Once it is complete, Make any changes that are needed and save
4. Go back into GitHub for Windows
5. At the top you should see “uncommitted change”, select it



1. Give the change a meaningful summary and description to use as a changelog, and then click “Commit to master”

