## Massey University

## 159.251 Software Design and Construction

## Tutorial 1 - Version Control with Git

### Prerequisites (what you are expected to do before you start the tutorial)

1. Study the lecture material on version control
2. Setup the online account on a remote repository site ([BitBucket](https://bitbucket.org/dashboard/overview), [GitHub](https://github.com/), or [GitLab](https://about.gitlab.com/)).

### Tools Required

* Git ([https://git-scm.com](https://git-scm.com/downloads))
* Remote Repository ([BitBucket](https://bitbucket.org/dashboard/overview), [GitHub](https://github.com/), or [GitLab](https://about.gitlab.com/)) online account.

### Objectives

1. Get started with version control
2. Basic understanding of using Git.

**Git- getting started**

Git is very low impact to get started, as you can work initially solely in your work directory.

* Download and install Git (<https://git-scm.com/downloads>)

Just open terminal/Command Prompt or Git Bash (Git for Windows provides a Bash emulation used to run Git from the command line) and run your commands.

* After installing git (as above) –check that git is installed correctly:

**git**

* configure your Git- tell Git who you are (important- used to track commits):git

**git config --global user.name "ABC XYZ"**

**git config --global user.email ABC@home.com**

* display help menu to see the list of commands

**git help**

**Tutorial 1 Instructions**

**Download the zip file from Stream (Tutorial 1- Git.zip) and then unzip the file in one of your home selected directories. Do the following tasks and show it to us:**

1. Create (initiate) a new Git repository of the same folder

**Hint:** To initiate a repository (turn the folder into a repository):

**git init**

this creates the *.git* folder but doesn't affect anything else

1. Add and commit only **.py** files (add a meaningful message when committing the files)

**Hint:** First, add files

**git add leapYear.py** #to add a file

**git add .** #to add ALL files in the folder

Then commit that to your local repository

**git commit -am "Your commit message"** #commit –and add a message

1. Create a new branch (FirstBranch)

Hint: You can create a new branch using the following command

**git branch <branchName>**

Make sure that make the current branch is the working branch

**git checkout <branchName>**

1. Add and commit only **.java** files

Hint: same command as step 2

1. Create a new branch (SecondBranch)

Hint: same command as step 3

1. Add and commit ALL other files
2. Merge both branches with the *master* branch (in the same line! Use only one command)

Hint: You can merge the branch as follows

**git merge <branchName>** #merging branchName with ‘master’

1. Delete both new branches

Hint: A branch be deleted using the following command

**git branch d- <branchName>**

1. Show the your branches using

**Git log # to display the entire commit history**

**git show-branch --all**

**git ls-tree -r master --name-only #to list all files on master branch**

1. Manage conflicts

Open leapYear.py and remove the following line:

**Print(“done”)**

Then create a new branch “feature-2” , and do the following

**git diff leapYear.py**

**git add leapYear.py**

**git commit -m “changed leapYear.py”**

merge this branch with the *master* branch

to display the commit graph, use

**git log --all --oneline --graph –-decorate**

1. Add a new annotated tag for the new version.
2. Host your repository in a remote site, and push all commits to the remote repository. You can use any Git hosting sites such as [BitBucket](https://bitbucket.org/dashboard/overview), [GitHub](https://github.com/), or [GitLab](https://about.gitlab.com/).

Some important command that you will need

**git remote add origin <URL site>** *# The hosting site will generate a unique URL for your newly created repository. URL example:* [*https://XYZ@bitbucket.org/XYZ/test.git*](https://XYZ@bitbucket.org/XYZ/test.git)

**git push -u origin --all** # pushes up the local repository for the first time

**git push origin --tags** # pushes up any tags

**What to Submit**

Submit your remote repository URL. This has to be public (so that we can clone the repository and view history). Make sure that it has the history of the steps you have gone through in the tutorial.

**-Tutorial Ends here-**

**Explore Git Desktop Client (Optional Task)**

Download and use a Git Desktop Client ([SourceTree](https://www.sourcetreeapp.com/))

* + Import your remote repository
  + Repeat steps 1,2,4,5 and 6 (for the single branch only) using the GUI interface (much easier than the command line!!).

**Other useful commands**

You can use Git clean to remove untracked files from the working tree

**git clean –n** # dry run of *git clean* - show you which files are going to be removed without actually perform the clean.

**git clean –f** # to remove all untracked files from the current working directory.

For more information about ***Git clean***, read this: <https://www.atlassian.com/git/tutorials/undoing-changes/git-clean>

|  |  |
| --- | --- |
| To see what's changed | git status |
| To see what's changed since last commit | git diff |
| To see a side-by-side view of changes | git difftool |

Note: When viewing the git log, or git status, press **q** to quit viewing, **space** for next page, **enter** for next line.