



## Faculty of Science

**Course:** CSCI 2020U: Software Systems Development and Integration

**Lab:** #1

**Topic:** Version control

### Overview

In this lab, you'll create a repository, add files to it, and sync it with a remote repository. You'll also create a new branch, and merge the changes with your main branch. We can use this repository for all future coursework.

### Part 1: The Basics

- Boot into your Linux partition (or open Linux in a VirtualBox VM).
  - You may use other operation systems, but be aware that your TAs might not be able to help you.
  - If using Windows, it is recommended to use GitBash as your terminal.

You will create a new directory in your home directory called `csci2020u`. We'll make this folder a Git repository. Use the following steps to complete this lab:

1. Create a new directory called `csci2020u`
2. Change into this directory
3. Initialize the current directory as a new Git repository
4. Make a new directory, called `lab01`
5. Add the `lab01` directory to Git
6. Commit all changes to the local repository (for all commits, include a useful message)
7. Sign-up for a new GitHub account (or log into an existing one)
8. Create a new repository on your GitHub account, called `csci2020u_<your name>`
9. Set this GitHub repository as a remote origin linked with the local Git repository
10. Push all changes to the remote repository
11. In the `lab01` directory, create a new Java source file
12. View the changes in the files
13. Commit the changes to the repository
14. Compare your local repository contents with the contents of the remote repository on GitHub
15. Create a `.gitignore` file to ignore files that are normally not committed to a repository (e.g. `.class` files)
16. Make changes to the cloned repository
17. Compare the two repositories (remote and local cloned)
18. Commit the changes to the local (cloned) repository
19. Compare the two repositories again
20. Push the changes to the remote repository
21. Pull the changes you've just made

This was designed to replicate a typical session when working with a version control system. Multiple team members are often working on the same project simultaneously, with different goals in mind. They often work independently on this code, and the version control software assists this process by allowing them to see other teammates' changes, and incorporate those changes into their own.

**Note:** *Do not commit error-ridden code to the repository. It is bad for team morale when one member is working on another problem and must stop due to bugs from another team member's work. Get it working, then commit/push the changes.*

## Part 2: Branching and Merging

In this part, you'll create a new branch, called `beta`, and make some changes to your source file. Follow these instructions to finish this part of the lab:

1. Create a new branch, called `beta`
2. Switch to the new branch (`beta`)
3. Make some changes to the `beta` branch
6. Check in the `master` branch (observe the changes)
4. Make some changes to the `master` branch
5. Compare the two branches
7. Merge the changes together into the original (`master`) branch

Branches should be used when adding significant new behaviour, but other team members are still working on the old functionality. The new functionality will not affect the other team members if they continue to work on the old branch.

## How to Submit

### In session

#### *(Preferably)*

- Show your local and remote repositories to the TA to prove that you have finished this lab.
  - This can happen by your sharing your screen to the TA or direct messaging them with screenshots.

### After lab hours

#### *(1 week to submit - before your next lab session)*

- In one PDF documents attach the following:
- Screenshot your command line terminal with the steps
- Screenshot your local repository where relevant
- Screenshot your remote repository showing creation, commit, branching, and merging history
- Link of your GitHub repository (if it is a public repository)

The TA can provide oral feedback if you do not receive full marks for any lab assignment, but it is most appropriate to ask the TA for this feedback in a timely fashion (i.e. ask now, not at the end of the term).