CST0006 – Computer Programming Foundations

INTRODUCTION to WEEK01

In this lab section we are going to learn how to use the Git Software Control Management System to track the changes of a software project and how to use a Workflow Board to manage the addition of software features . Learning these tools is essential either if you are working alone or collaborate with others on software projects.

Git

Last week we installed git and the git software tools that we will need for this lab. You can use any operating system for this lab, but if you are using Windows, you will need to use the git-bash shell.

Part I – Try Git

We are going to go through the "Try Git" web site. If you've already gone though it, please go though it all again. The more you do this, the more it will sink in.

Go here to do that now: https://try.github.io

Also, please pay attention to what you are doing. Pay attention to the entire screen and read all the notes that are being said. They all matter!

Take a screen shot of the completed tutorial, so that I know you've gone though each step.

Part II – Create a remote repository

We are going to work with a partner in this portion. You can have groups of two or three people, but keep the groups small for now. It's going to be difficult enough with just two or three for this portion.

Create a new git repository on Gitlab:

Project name: "2018W_CST0006_LAB01_<*your_Algonquin_username*>"

Visibility Level: Private

The username is important, because you will be downloading other people's repository, and if you only use "2018W_CST0006_LAB01" you will overwrite your own work when you download their repo.

Follow the instructions under: "**Create a new repository"**, so that you have a local copy.

Once you've done that, add the members of your team into your repo. Go to "Settings/Members", then search for their username and set them as a developer. You don't need to set any expiration date. Then click "Add to project". Once you've done that, they will get an email notification.

Part III – Creating a local copy of the remote repository

When you get the email inviting you to your teammate's repo, open "Git Bash" (or the terminal if you're not in Windows). Copy the repo location, and on the command line type the following:

If you haven't yet created a git folder and a term folder (note: the '~' is a tilde, not a dash)

```
cd ~
mkdir git
cd git
mkdir 2018W
cd 2018W
```

If you have, just simply type (note: the '~' is a tilde, not a dash)

```
cd ~/git/2018W
```

Clone the remote repository

```
git clone <remote_repo_url>
```

Change into the repo directory

```
cd <repo_folder>
```

Part IV - Branching

This part can be a fun and exciting part of the lab. IF you follow these instructions properly.

I want you to create a branch and checkout that branch, add some content (files), maybe even a picture… refer to the git cheat sheet if you can't remember how to do this.

Hint: To easily create a new file in "Git Bash", you can use the touch command.

```
touch <new_filename>
```

You can then view the files in the directory by using the ls command (ls is short for list)

```
ls
```

You can also use the ll command (which is an alias of ls -l)

```
ll
```

Once you've finished adding a branch, checking out that branch and adding content, push your changes. **Pay attention to any error messages.** I want you to read these. There is a step here that will require you to add your branch to the repository so that others can see it.

Follow these instructions carefully.

Get used to reading these error messages. They will help you troubleshoot any errors you've made.

By the way, it is important to mess up now and not later when you're working on something important.

Since you don't have write access to the main branch (master), you will need to create a merge request.

To do this, you need to go to Gitlab, and click on the merge request button. Choose the branch name you created and where it needs to merge into. This will send a notification to your team mate asking them to accept your awesome changes.

Take a screenshot of this merge request, you will need to submit it for lab marks.

Workflow Boards

For this section, we are going to play with the workflow board that is a recently new feature added to Gitlab. Before this feature was added, you had to use other boards and try to integrate them into Gitlab or just use them separately.

On **your own** repository in Gitlab, Click on the "Issues" button.

A dropdown menu will open and there will be a series of workflow tools you can use with your project.

I would like you to create at least three issues, change the board to add a To Do and Doing and assign some issues to the other members in your team.

Again, play around with this. The more you try and the more you try to do, the more you will learn. All of this stuff will make any team-related projects in the future easier to manage. Git works with any type of file, so presentations or essays will work too. It will also protect you from any member that isn't contributing. Tracking who does what and who isn't keeping up their end of things can be taken to the prof and you can show them who isn't doing their work. Trust me when I say this, there are team members that will do nothing and they will put you at risk of failing the whole course because of their laziness. It is wise to safe-guard yourself from people like this by using a tool like Git and Gitlab.

Take a screenshot of the board with tasks and issues.

Submission

When you are done, submit all three screenshots to blackboard.