

# Submitting Lab Code

**Quick Links:** [Home \(https://q.utoronto.ca/courses/419441/assignments/syllabus\)](https://q.utoronto.ca/courses/419441/assignments/syllabus) [Lecture material \(https://q.utoronto.ca/courses/419441/pages/lecture-material\)](https://q.utoronto.ca/courses/419441/pages/lecture-material) [Lab assignments \(https://q.utoronto.ca/courses/419441/pages/lab-assignments\)](https://q.utoronto.ca/courses/419441/pages/lab-assignments) [Piazza Discussion !\[\]\(c8d96c8885d3000a912c2582004aed63\_img.jpg\) \(https://piazza.com/utoronto.ca/winter2026/ece353/home\)](https://piazza.com/utoronto.ca/winter2026/ece353/home)

---

## Submitting Lab Code

Please follow the instructions for submitting your lab code carefully.

If you make *any mistakes* with these instructions, you will find that you are unable to submit your code correctly, so make sure to check and type each command correctly. Look out for "0" (zero), "O" (big O) or "o" (small o). The domain name is `utoronto.ca`, not `toronto.ca`, etc.

It is a good idea to try doing a submission much before the deadline so that you know that the submission process works. You can submit code as many times as you want before the deadline.

## Setup Remote Repository

You will need to do this step once.

Recall that you created a Git repository in `~/ece353` (see [Setup Instruction 2 in Lab 1 \(https://q.utoronto.ca/courses/419441/pages/lab-1-review-of-c-data-structures-using-system-calls#setup\)](https://q.utoronto.ca/courses/419441/pages/lab-1-review-of-c-data-structures-using-system-calls#setup)). We (i.e., the instructors and the TAs) cannot access this repository because it is within your home directory (that should not be accessible to others). Instead, you will submit your lab code by creating a clone (copy) of your repository to a remote repository location that is accessible to you and to us.

You need to run these instructions **once** to setup your remote repository. After that, you can update the remote repository by following the instructions in the next step.

1. Each of you has been assigned a [user number \(https://q.utoronto.ca/courses/419441/pages/lab-assignments#started\)](https://q.utoronto.ca/courses/419441/pages/lab-assignments#started) for this course. Below, we use XXX to denote your 3-digit user number.
2. The remote repository will be located on the `ug250.eecg.utoronto.ca` machine. Log in this machine.

```
ssh ug250.eecg.utoronto.ca
```

Next check that your file and directory create permissions will allow us to access your remote repository. To do so, run the following command.

```
umask
```

The output of this command should be 022 or 027 (or 0022 or 0027). If it is 077, please ask a TA how it should be changed.

Now create your repository as follows:

```
cd /srv/ece353s/os-XXX
mkdir ece353
chmod 750 ece353
cd ece353
git init --bare
```

Make sure to replace XXX with your 3-digit (not 2-digit) user number. The chmod command above ensures that the repository will be accessible to the instructor and the TAs but not to other students. The git init command creates an empty Git repository.

3. Now, you are done with this machine, so exit out of it.
4. Back on your local machine, you need to tell the local repository about the location of the remote repository.

```
cd ~/ece353
git remote add origin ssh://ug250.eecg.toronto.edu/srv/ece353s/os-XXX/ece353
```

The command shown above is run in your local repository. It associates the name origin with the remote repository. You can see that by using the git remote command.

```
git remote -v
```

It will show the following:

```
origin  ssh://ug250.eecg.toronto.edu/srv/ece353s/os-XXX/ece353 (fetch)
origin  ssh://ug250.eecg.toronto.edu/srv/ece353s/os-XXX/ece353 (push)
```

Now you are ready to update your remote repository.

## Update Remote Repository

1. Make sure that you are in your local repository.

```
cd ~/ece353
```

2. You can update your remote repository by running the git push command shown below. This command will only push the files that you have committed to your local repository, so make sure to

commit all the relevant files to your local repository that you would like to push to the remote repository.

```
git push -u origin master --tags
```

When you run the command above, you will see:

```
Branch master set up to track remote branch master from origin.
```

Recall that your local repository branch is called "master" (see [Setup Instruction 3 in Lab 1 \(https://q.utoronto.ca/courses/419441/pages/lab-1-review-of-c-data-structures-using-system-calls\)](https://q.utoronto.ca/courses/419441/pages/lab-1-review-of-c-data-structures-using-system-calls)). The command above pushes the branch called "master" in your local repository to a new branch, also called "master", in the "origin" (or the remote) repository. The --tags option pushes your local tags to the remote repository. We will be using these tags to mark your labs, so don't forget to add the tags option. If you run `git status`, you should see the following:

```
On branch master
Your branch is up-to-date with 'origin/master'.

nothing to commit, working directory clean
```

3. You need to only run the push command with all the options shown above once. After that, you can push your commits in the local repository to the remote repository by typing the following commands in your local repository:

```
git push
git push --tags
```

When you add a new tag (e.g., Lab1-end) and run the push command shown above, it will show you:

```
To ssh://ug250.eecg.utoronto.ca/srv/ece353s/os-XXX/ece353
* [new tag]          Lab1-end -> Lab1-end
```

You may resubmit your code any number of times. For example, you may modify your code and update the end tag in the local repository (see [Using Git in Lab 1 \(https://q.utoronto.ca/courses/419441/pages/lab-1-review-of-c-data-structures-using-system-calls\)](https://q.utoronto.ca/courses/419441/pages/lab-1-review-of-c-data-structures-using-system-calls)). To resubmit this updated code, you will need to force an update to the tag on the remote repository by using the -f option. Here is the full sequence of operations:

```
git tag -d Lab1-end // Deletes tag from previous submission for Lab 1
git tag Lab1-end    // Creates tag for current submission for Lab 1
git push --tags -f  // The "-f" option forces the tag to be pushed
                   // to the remote repository, even if the tag exist there
```

You can update the remote repository at any time as shown above.

## Submitting Code For Marking

You submit your code by simply tagging your code with the "end" tag for that lab and updating your remote repository. Then, you can test whether your submission succeeded by simulating our [automated marking](https://q.utoronto.ca/courses/419441/pages/testing-your-lab-code) (<https://q.utoronto.ca/courses/419441/pages/testing-your-lab-code>) functionality.