

# ECE653

## Software Testing, Quality Assurance, and Maintenance

### Assignment 0 (0 Points), Version 1

Instructor: Werner Dietl  
Release Date: May 13, 2024

**Due: 22:00, May 17, 2024**  
**Submit: commit on GitLab**

In this course, we will use GitLab<sup>1</sup> to distribute and submit assignments. The goal of this assignment is to test that you have everything necessary for GitLab to work.

- Create a GitLab account. You can use an existing account if you have one. The UW GitLab is at <https://git.uwaterloo.ca/>.
- Locate the git repository that contains the assignment skeleton. This is your **upstream**. It will be updated for every assignment. The URL is

`https://git.uwaterloo.ca/stqam-1245/skeleton`

- Locate the git repository that was created for you in the course group. The URL is

`https://git.uwaterloo.ca/stqam-1245/class/USERID`

where USERID is YOUR quest user id. For example, my repository is at `https://git.uwaterloo.ca/stqam-1245/class/wdietl`.

- Follow the instructions in `README.md` in directory `a0` in the skeleton repository to complete the assignment.

## 1 Detailed Instructions

### 1.1 Clone your personal repository

Set up your local git repository. If you are using the git command line interface, you can do so with the following command:

```
git clone https://git.uwaterloo.ca/stqam-1245/class/<USERID>.git
```

Where you can find the url to your GitLab repository at `https://git.uwaterloo.ca/stqam-1245`, under the `class` sub-group. It should look similar to the following:

```
git clone https://git.uwaterloo.ca/stqam-1245/class/<USERID>.git
```

Notice the `.git` suffix!

---

<sup>1</sup><https://git.uwaterloo.ca/>

## 1.2 Connect your repository to the assignment skeleton

```
git remote add upstream https://git.uwaterloo.ca/stqam-1245/skeleton
```

## 1.3 Get assignment content

Start by fetching the remote repository

```
git fetch upstream
```

If you have not done so before, create a main branch:

```
git checkout main
```

Merge the new content into your local main branch. You will have to do this every time the skeleton repository is updated for new assignment.

```
git merge upstream/main
```

Update your remote GitLab repository with the new content by pushing main there.

```
git push origin main
```

Finally, check that update actually worked by looking at the content of the repository via a web browser.

## 1.4 Complete the assignment

If you followed the instructions above, your local repository should contain a folder `a0` with a file `user.yml`

Replace `FIRST_NAME`, `LAST_NAME`, `WATIAM`, and `STUDENT_ID`, `EMAIL` in `user.yml` by your first and last name, WatIAM id, student number, and email, respectively. Use plain ASCII characters as much as possible.

Do not change the format of `user.yml`. It will be parsed automatically. Only enter the information requested.

Once you have made changes, and have added them with a command such as the following:

```
git add a0/user.yml
```

A “commit” will need to be made. Note that a “commit” contains the changes.

```
git commit -m "ENTER A COMMIT MESSAGE HERE. THIS IS FOR YOUR OWN REFERENCE"
```

Note that the message is optional. It is a message for your own clarity. However, you may find it useful to include meaningful messages in case you need to refer to previous, older commits.

```
git log
```

Once you have a commit ready, it can be pushed onto the GitLab Repository through the following

```
git push origin main
```

We strongly recommend that you go to GitLab and check to make sure your code is visible on GitLab after pushing.

Note that you can see all of the modified files with:

```
git status
```

If you are using git at the command line, it may be useful to configure git on the command line through the following commands.

```
git config --global user.name "FIRSTNAME LASTNAME"
git config --global user.email "USERNAME@uwaterloo.ca"
git config --global push.default simple
git config --global color.ui "true"
```

If you wish to practice using git, you may try the “Learn Git Branching” tutorial on the <https://try.github.io/> web page.

For marking, we will use the main branch of your repository at the time of the deadline. It is advised to commit and push often. This way a version of your work is always backed up into the cloud.