Part I – Installing Git (individual)

- As a prerequisite, each student should have git installed on their machine. Git is cross-platform and can be downloaded from here http://git-scm.com/downloads

- The command line interface is the minimum requirement, but GUI-based front ends can also be downloaded and installed.

- If you use an IDE (Eclipse, IntelliJ, NetBeans, …), latest editions will support Git natively unless you downloaded a "bare minimum" version. In that case, Git can usually be downloaded as a plugin: http://eclipse.org/egit/ (Eclipse), https://plugins.jetbrains.com/plugin/3033?pr=idea (IntelliJ), https://netbeans.org/kb/docs/ide/git.html (NetBeans)

At this point, you should be able to work with Git locally

Part II – Working with Git and Github (individual)

An organization has been created on GitHub, called BMI-591-CAD-2015, which is dedicated to the 2015 edition of the course. All students have been invited to become members. Now do the following:

- Go to the organization's master page (https://github.com/BMI-591-CAD-2015) and create a new repository using the "Create new..." in the top right part of the screen.
  - The repository should be named "NameSurname".
  - Remember to set the Organization as the owner, rather than yourself as per default!
  - Initialize the repository with a .gitignore file
    (this will also create a master branch on the project, which otherwise you would have to add manually)

For example, a test repository should look like : https://github.com/BMI-591-CAD-2015/MockStudent

- Now fork the organization's "blessed" repository into your personal workspace. Click on the "Fork" button in the top right part of the screen and wait a few seconds.

Now you should be taken to a URL like https://github.com/bmi591/MockStudent (bmi591 will be replaced by your github ID)

- Now create a local working copy somewhere in your machine. Run "git clone" with the SSH/HTTPS link you'll find in the repository. Make sure you use the one from the organization:
  - git clone https://github.com/BMI-591-CAD-2015/MockStudent.git
- This creates a "MockStudent" folder. Go in there and check that the local working copy is correctly linked to the remote one using "git remote show". Notice that the remote repository has been assigned a local alias, "origin" by default:
  - git remote show origin
- In order to link your working copy to your personal copy, use "git remote add" and give it a local name such as "personal" (any would work as long as it is not yet in use). This command requires the URL of your personal repository:
  - git remote add personal https://github.com/[bmi591]/MockStudent.git

- If you run “git remote show”, you will see a list of two remote locations. Running “git remote show personal” will confirm that the newly added link points to your personal remote repository.
- Now you can populate the repository, for example with a maven structure.
  - Copy the POM file and the folders from your last week's assignment into the git-enabled folder. See that it builds successfully
    - mvn clean install
  - Run “git add .” to ensure that git will manage all the files
    - git add .
  - Commit the changes. Notice that the remote repositories will not be affected (yet)
    - git commit -m “Put some comment here”
  - Push your changes to your personal repository. Notice that in this example you would be able to push to the organization's repository as well, but this would not normally be the case.
    - pit push -f personal
- At this point your personal repository will be updated. Go to the github website and check it out. Also see that the organization's repository has NOT been updated yet.
- Open a “pull request” (PR) from your personal repository to the organization's one. There is a “Pull Request” button in the central part of the screen which may be a bit hard to spot initially. Confirm the pull request in the next two screens.
- You will be taken to a page – owned by the organization – that shows that the pull request has been received. You normally won't be able to “Merge pull request”, but you can leave comments. This page can also be accessed later from the “Pull Requests” link (on the right side of the screen) from the organization's repository. Notice that the PR is accessible from the target's repository, not the source one. If you try to open another PR from the source repository, you will be taken to the existing one.
  Accept the PR and notice that the organization's repository has been updated.
- Finally, ensure that your local repository is in sync with all the remotes. In particular, make the local repository aware of what happened to the organization's repository:
  - git fetch origin
- At this point all repositories are synchronized and up-to-date.

Part III – working in teams (team)

- Each team should create a repository for their project, fork it and clone it.
  - The repository should be created by a team member **under the organization**, as per the previous exercise.
  - The creator will invite the other team members (who will accept the invite).
  - Each team member will fork the repository to their personal account and clone it on their machines.
- Now <u>each team member</u> will repeat the following steps:
  - First and foremost, incorporate any change incoming from the “blessed” organization's repo:
    - git fetch origin
    - git rebase origin/master
  - Each member will make some changes to the repository (add a file, edit an existing one, …), add, commit and push to their personal repo:
    - (edit something)

        - git add .
        - git commit -m “this is what I did...”
        - git push -f personal
    - Now go to github, open a pull request and accept it
    - Resynchronize your local copy
        - git fetch origin
        - git rebase origin/master

If you apply these steps consistently, you will always be up to date with your colleagues' work, the project will have a linear commit history and conflicts will be detected as soon as possible.

Over the next weeks, we will use these repositories for both individual and team assignments and milestones.