HOMEWORK 0

- **DUE TUESDAY, 1/20/2017, 11:59 PM**

Do your own work for this assignment; do not work with others. Consult the resources provided and your professor for help if you need it. **This assignment will be submitted in two ways, electronically on GitHub, and printed at the beginning of class.**
Total points: 80.

In this homework assignment you will do mainly four things:

- Setup Java in your computer
- Create a Kattis account
- Create a GitHub account
- Add, commit, and push you first homework files on GitHub

You will be collecting several screenshots as evidence that you did your homework. You should put them all into a document (landscape format) that you will submit as a PDF file.

Setup Java

Your first mission is to install Java in your personal computer. Now, here is the thing, your computer might already have Java installed. To find out use the shell to run the following commands:

Test if you have the Java Virtual Machine runtime environment installed as follows:

```
java -version
```

If you do have installed the Java VM, you should receive details of the version installed in your system. Take a screenshot of that and put in a document along with all the next screenshots you will take. Take it so that it is legible please.

Now, test if you have the Java compiler installed as follows:

```
javac -version
```

If you do have the Java compiler installed, you should receive details of the version installed in your system, although it would be less information than with the Java VM above, but that is normal. Take a screenshot of that and put in a document along with all your screenshots. Take it so that it is legible please.

If you **do not have the compiler installed** you should follow instructions below for your system. However, if you are a Windows user (not judging anyone here) there is a chance that you have the Java compiler but it is not in your system path. If that is the case, follow instructions below but skip until the part that says how to add it to the "system environment" or to the system path. Everyone

else, follow all instructions for your system, then come back and run the commands above to make sure you installed it correctly.

- "Oracle Solaris Operating System"
- "Microsoft Windows"
- "Linux"
- "OS X"

Create an account on Kattis

Using your marist email account go ahead and create an account in Kattis.

- Go to https://open.kattis.com/register
- Choose a good username
- Select your country as "United States"
- Select "Marist College" as your university
- Choose your default programming language as "Java"

The rest you are free to fill up at your discretion, e.g., profile picture, privacy, email preferences, etc. But usually the default options work just fine.

Once you are done, go to:

```
https://open.kattis.com/users/YOURUSERNAME
```

where `YOURUSERNAME` is the user name you specified, and take a screenshot of your profile as proof, and add it to the document where you are collecting all your screenshots.

In Project 0 you will actually submit your first assignment to Kattis, but for now, you are done with this part. Move on to the next one.

Create an account on GitHub

PROFILE

Click this link to create an educational GitHub account. It is very important that you use your @marist.edu account here too. Otherwise it will be nearly impossible to validate your student status. Follow instructions for creating your account.
Once you create your account go to your profile overview page:

```
https://github.com/YOURUSERNAME
```

where `YOURUSERNAME` is the user name you specified, and take a screenshot of your profile as proof, and add it to the document where you are collecting all your screenshots.

REPOSITORY

Now it is time to create your class repository. Before you begin, think of a short meaningful name for your class repository, e.g. cmpt220lastname, where "lastname" would be your lastname.

Remember, **only one** repository is needed for the class, thus, once you create it, you must use only that one for the class. All right, follow this instructions to create your class repository. **Do not add a Read Me** to your new repository, otherwise it may cause conflicts down the road.

Now, it is time to add your professor as a collaborator to your repository. Why, you ask? Well, this is the best way for your professor to see how often you pushed/committed to your repository, it allows your professor to see when you worked, what progress you were making, what changes you made to different versions, and more importantly, it allows your professor to post feedback into your repository. Please follow this instructions about how to add your professor to your repository. Note, your professor's GitHub user name is `jfac65-marist`.

If everything went well, congratulations you created your class repository. Now, it is time to produce the evidence. Go to

```
https://github.com/username/cmpt220lastname
```

where `username` is your username, and `cmpt220lastname` is the name of the repository you chose for the class. Once you are in that page, click on the green button that says "Clone or download" and take a screenshot of that page and save it into your document with your other screenshots. Please, copy the link under "Clone with HTTPS" that appears after clicking the green button, and paste it in the same document under the screenshot. This link will enable your professor to setup access to your repository.

Submitting this homework to GitHub using Git

So far you have been using a document collecting screenshots. Now it is time to wrap things up and submit this homework. So, make sure your document has your name and ID, you know, in a nice cover page. In summary your document should contain the following screenshots and information:

- Java VM version
- Java compiler version
- Kattis account
- GitHub account
- GitHub class repository
- GitHub class repository HTTPS link

Somehow make evident that you did all this in YOUR computer. Ask yourself the question, "how can the professor be sure that I did all this work in my own computer and not in any other machine?" There are ways you can do that, be creative!

Save your document as a PDF file. **Only a PDF will be accepted.** A suggested name is `hw0lastname.pdf` where "lastname" is your last name. Once you saved your homework as a PDF, upload your homework using Git. Lets pretend that your repository is called `cmpt220lastname`. Then all your files for homework should be in a folder `hw` like this:

```
cmpt220lastname/
  hw/
    0/
      files for homework 0 here
      ...
    1/
      files for homework 1 here
      ...
  prj/
    0/
    1/
    ...
  labs/
    1/
    2/
    ...
```

where the folder `cmpt220lastname/hw/0/` is where you would put all the files you want to submit for Homework 0, namely, your PDF file (and your source .doc file if you wish, but the **PDF is a must**).

First, let us assume that you have created somewhere in your file system (hopefully somewhere you will remember) the folder explained above called `cmpt220lastname`, using the shell change directories until you are inside that folder; note that Windows users (not judging here) might need to use something called "Terminal Git Bash". Anyway, inside that folder issue the following command:

```
git init
```

which initializes the directory to be a Git repository. **You will do this step only once, at the beginning of the semester**. There is no need to initialize the repository over and over again.

Second, change directories until you are inside a directory called `cmpt220lastname/hw/0/` and inside that folder is a file called `hw0lastname.pdf` that contains your homework, you will indicate Git to start tracking this file (and all inside the directory) by running:

```
git add .
```

which is a process called "staging" if you are curious. **In further assignments you will repeat this step every time you create new files that you want to add**.

Third, you have to commit your files officially as part of your current version. Committing your files does not upload them to GitHub yet, and the files being committed are only those that were "staged" (added) in the previous step. Committing files allows you to write a reason why you made changes to your repository, this is specified in the form of a meaningful message. Here is how you do it

```
git commit -m "This is my first commit and my long meaningful message."
```

This command commits the tracked changes and prepares them to be pushed to your remote repository on GitHub (coming up next). **In further assignments you will repeat this step every time you staged (added) new files that you want to commit**.

Fourth, now you configure your GitHub (a.k.a. remote) repository by telling Git that you have setup a cloud repository on GitHub. For this you will need that HTTPS link that can be retrieved pressing the green button in your repository, but, fear not, you have this saved in your homework PDF file, so have that ready and issue the following command:

```
git remote add origin https://github.com/username/cmpt220lastname.git
```

where `username` is your username, and `cmpt220lastname` is the name of the repository you chose for the class. This command sets the new remote repository. Then, to verify the repository do:

```
git remote -v
```

and if everything went well you should get no errors there. **You will do this step only once, at the beginning of the semester**. There is no need to link the local repository with the remote (GitHub) repository over and over again.

Fifth, you will finally push (upload) your local version to the cloud as follows:

```
git push origin master
```

which pushes the changes in your local repository up to the remote repository you specified as the origin (see the Fourth step above). And that is it, if you go and refresh your web browser and go and check your repository, you should see your "pushed" files in the cloud now. Yay! **In further assignments you will repeat this step every time you want to push (upload) committed files to the GitHub cloud**.

NOTE: GitHub doesn't like empty folders, so do not be surprised if you proactively created some empty folders and they do not show up on GitHub, that is OK. Once you put stuff inside them, they will be created and their content uploaded if you correctly added them and committed them.

Finally, **print your homework and bring it to class** as well. This concludes your Homework 0. If your homework PDF file does not show up on GitHub, something went wrong, and you won't receive a grade because **your work is not officially submitted if it is not on GitHub**, so go back and review what you did, until you know what went wrong and correct it. If you had any errors along the way, do your research, investigate, use GitHub help and google to get help for your specific operating system. It is hard for your professor to foresee all OS scenarios and variants; this was just a general idea, and **you are responsible if you did not allow sufficient time to make sure you have the correct setup and followed the necessary steps for your system**.

A final note on dektop applications: some companies look down on developers that do not know how to use the shell (terminal or command line) Git commands, and it is not hard to learn, really; there is a small number of commands that you need to learn, `pull`, `add`, `commit`, and `push` are the ones you will probably use the most. Invest time right now in learning this source control tool and you will be rewarded in the future.