# Assignment 2 - Git and Object Oriented Programming warm up

**DEADLINE:** Thursday, October 12, 2017, 9:29 AM.

**Objectives**:

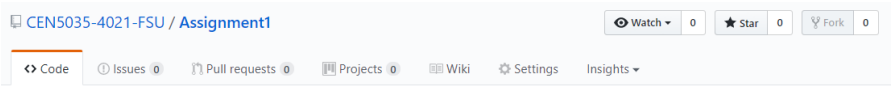
- Familiarize yourself with GitHub commands
- Refresh your memory on important Object Oriented Programming concepts

## 1 Overview

You will use a Git client to add files and change existing files in a repository. You will also complete the implementation of a small RoleGame prototype written in C++, constantly committing and pushing your changes to the Git repository. You can use any IDE, source code editor, or text editor to complete the implementation.

## 2 Set up

1. Install a Git client (if you don't have one already). You can get one from `https://git-scm.com/downloads`
2. Create a GitHub account at `https://github.com` (if you don't have one already)
3. Fork the following repository `https://github.com/CEN5035-4021-FSU/Assignment1` using the GitHub fork option. This will result in you obtaining your own copy of the repository and all its contents in your GitHub account. You will find the GitHub's fork option at the top right corner of the screen:




4. Clone the forked repository (i.e., the one that is in your GitHub account) to your local machine
5. Read and familiarize yourself with the existing code

## 3 Assignment

The following steps are done in your local copy of the repository.

1. Modify the file README.md to include your name and your FSUID in the first two lines. (5 points)

   **Example:**
   Name: Javier Escobar-Avila
   FSUID: jre14c

2. Commit the changes from item 3.1 to the repository (5 points). Write an appropriate commit message. Follow the guidelines presented at: `http://chris.beams.io/posts/git-commit/`

   **Example:**

   *Updated information files*

   *The information files have been updated to describe the current state of the repository.*
   *Repository owner information added to README.md*

3. Use the command "git status" on a console. Copy and paste the output of this command to the docs/status.txt file. ***Using the same git console***, commit and push your changes to the repository. Copy and save (i.e., paste in a temporary file) the output of the console after you have pushed your changes. (5 points)

4. Create a new file named "answers.txt" inside the "docs" folder. Add, commit, and push this empty file to the repository. (5 points)

5. Answer the following questions in the answers.txt file, and then commit and push the changed file to the repository:

   (a) Please paste the console output you saved at the end of step 3.

   (b) How many commits have been done to the repository (not only by you, but by anyone) so far? Please write the git command you used to get this information. (5 points)

   (c) When was the file .gitignore modified last? Please write the git command you used to get this information. (5 points)

   (d) Please list the IDs of the commits used to merge the changes from the branch "development" to the branch "master". (5 points)

   (e) Mention two reasons for using branches in a Git repository. (6 points)

(f) What is the difference between `git log` and `git status`? (5 points)

(g) What command would you use to see the commits where "Weapon.h" was one of the committed files? (5 points)

(h) What command would you use to see the commits whose commit message contains the word "file"? (5 points)

(i) In the context of object oriented programming (I) What is inheritance? (II) What is polymorphism? (III) What is encapsulation? (9 points)

6. Use the content of the forked repository as a basis to finish the implementation of a simple RoleGame prototype. Commit and push your changes to the repository frequently (e.g., after you create a class) using a meaningful commit message. Make sure you commit all your valid changes, and double check that everything was properly pushed. If it is not in your GitHub repository, it does not count for the grade.

**RoleGame prototype**

This small prototype of a role game has the initial code for the Weapons module. The main components of this module are the abstract class "Weapon", the class "WeaponFactory", and a set of implemented weapons. We provide the classes "CommonSword" and "CommonSpear" as examples of weapons, each one with the following features:

- CommonSword: hitpoints = 50. It does not ignore armor points.
- CommonSpear: hitpoints = 40. Ignores 20% of the armor points.

You are required to implement 3 more weapons as separate classes:

- SimpleAxe (5 points): hitpoints = 60. If armor is greater than 0 and less than 20, the axe will ignore all the armor points.
- CrazyRandomSword (10 points): hitpoints = random integer number between 10 and 100. Ignores a random amount of integer armor points, ranging from to 0 to half of the armor the weapon hits.
- Propose your own weapon (as a new class). The weapon must have hitpoints greater than 0 and must interact in some non-trivial way with the armor it hits (10 points).

**HINT:** If you generate double or float random numbers, trim the decimal part of the number and keep only the integer. In other words, use a floor function.

**EXTRA POINTS:** Use the "development" branch to implement all your changes. Then, when you consider that your prototype is ready for submission, use a merge commit to merge the changes in the "development" branch into the "master" branch. Therefore, the "master branch" can only have one modification from you, but the development branch can have as many as you need. (10 points)

7. Answer the following question in the answers.txt file, and then commit and push the changed file to the repository (10 points):

Is the RoleGame prototype using polymorphism or encapsulation? If you find that the prototype is using any of these two OOP principles, please discuss in which way the prototype is using it.

**RULES**: All weapons must inherit from the abstract class "Weapon" and the creation of their instances must be delegated to the class "WeaponFactory" (i.e., you should not use the "new" keyword in the "main()" method). Use the implementation of the existing classes as guide.

Print the behavior of the new weapons using the main method. Note that the new weapons will simulate the interaction with a character with no armor, and a character with 20 armor points (the existing implementation of the main method will guide you).

CEN4020 is not a programming class, so it is perfectly fine if you go to the Internet to get some advice on how to implement a certain requirement (e.g., going to StackOverflow.com to get some code about how to generate random numbers in C++, etc.). You still need to write some code on your own, though. If you get the source code (even partial implementations) from some web page or Internet resource, please indicate the source in your source code comments (you will not be deducted points for doing this). **You are not allowed to collaborate with your classmates or other students/developers**. If you have any doubt or question, please contact the TAs as soon as possible.

**Deliverables:**

The contents of the Git repository in your GitHub account by the deadline (remember to push your changes to your GitHub repository!!). We will access your GitHub repository to grade the assignment; therefore, you do not need to submit any document or zipped folder to Blackboard. Any change committed or pushed after the deadline will be considered as a late submission.