UNIVERSITY OF THE PHILIPPINES VISAYAS
COLLEGE OF ARTS AND SCIENCES
DIVISION OF PHYSICAL SCIENCES AND MATHEMATICS

CMSC 126: WEB PROGRAMMING
2nd Semester AY 2024-2025

Prepared by:
Nikko Gabriel J. Hismaña
Instructor, Faculty of Computer Science

LAB EXERCISE 1: Git and GitHub

ACADEMIC INTEGRITY

As a student of the University of the Philippines, I pledge to act ethically and uphold the value of honor and excellence. I understand that suspected misconduct on given assignments or examinations will be reported to the appropriate office and if established, will result in disciplinary action in accordance with University rules, policies and procedures. I may work with others only to the extent allowed by the Instructor.

Final Project and Grouping:

In this course, you are also divided into groups for a **final project**, a web application or website, to be passed at the end of the semester. Each group should be comprised of 3 members. When you start developing your final project, it is advisable for each member to take up a specific role to streamline the development: Frontend Dev, Backend Dev, Project Manager (for leading, documentation, and testing).

Instructions:

1. Submit your group members (3 members each) and group name
2. Submit your top 2 ideas for your final project
3. **Deadline**: After the first long exam

Laboratory Exercise #1: Git & GitHub

As a software developer, you will be tasked to collaborate on a specific project. For your first individual lab exercise, you should be able to do all the tasks below.

DOWNLOADING AND INSTALLING GIT

- *To install Git on your computer, go to https://git-scm.com/downloads to download the Git installer for your specific computing platform.*
- *Then, follow the installation steps as you install Git using the installer.*
- *You can find more details about installing Git at https://git-scm.com/book/en/v2/Getting-Started-Installing-Git . This document lists several ways of installing Git on various platforms.*
- *Installing some of the GUI tools like GitHub Desktop will also install Git on your computer.*
- *On a Mac, setting up XCode command-line tools also will set up Git on your computer.*
- *You can choose any of the methods that is most convenient for you*

GLOBAL CONFIGURATIONS

1. Open a cmd window or terminal on your computer.
2. Check to make sure that Git is installed and available on the command line, by typing the following at the command prompt:

```
git --version
```

3. To configure your username to be used by Git, type the following in the prompt:

```
git config --global user.name "Your_Name"
```

4. To configure your email to be used by Git, type the following:

```
git config --global user.email your_email_address
```

5. `You can check default Git global configuration using:`

```
git config --list
```

BASIC GIT COMMANDS

In this exercise, you will get familiar with some basic Git commands. At the end of this exercise you will be able to:

- Set up a folder as a Git repository
- Perform basic Git operations on your Git repository

1. At a convenient location on your computer, create a folder named **git-test**.
2. Open this git-test folder in your favorite editor (in my case, I used Visual Code).

3. Add a file named **index.html** to this folder, and add the following HTML code to this file:

```
<!DOCTYPE html>
<html>
       <head></head>
       <body>
            <h1>Hello, World!</h1>
       </body>
</html>
```

4. Initialize the folder as a Git repository. Go to the **git-test** folder in your terminal, and type the following at the prompt to initialize the **git-test** folder as a local Git repository:

```
git init
```

5. Check the Git repo status:

```
git status
```

6. Adding files to the staging area. To add files to the “staging area” of your Git repository, type:

```
git add .
```

Think of this as “selecting”/”highlighting” all the files in the folder.

7. Commit the current staging area to your Git repository, type:

```
git commit -m “first commit"
```

8. To check the log of the commits to your Git repo, type:

```
git log --oneline
```

9. Now modify the **index.html:**

```
<!DOCTYPE html>
<html>
       <head></head>
       <body>
            <h1>Hello, World!</h1>
            <p1>This is the first paragraph</p>
       </body>
</html>
```

10. Add a **sub-folder** named **templates** to your git-test folder, and then add a file named **temp.html** to the templates folder. Then set the contents of this file to be the same as the index.html file above.
11. Then, check the status and **add all the files to staging area**.
12. Now, do a **second commit** to your repository.

13. Next, modify the **index.html** file to:

```
<!DOCTYPE html>
<html>
       <head></head>
       <body>
            <h1>Hello, World!</h1>
            <p1>This is the first paragraph</p>
            <p1>This is the second paragraph</p>
            </body>
</html>
```

14. Now, add the modified index.html file to the staging area and do a **third commit**.

CHECKING OUT A FILE FROM AN EARLIER COMMIT

1. To "check out" the index.html from the second commit, find the number of the second commit using the **git log**, then type the following:

```
git checkout <second commit' number> index.html
```

2. Now, check if your file has reverted back to the second commit version (without the second paragraph).

RESETTING THE GIT REPOSITORY

1. To discard the effect of the previous operation and restore **index.html** to its state at the end of the **third commit**, type:

```
git reset HEAD.index.html
```

2. Then type the following at the prompt:

```
git checkout -- index.html
```

3. You can also use **git reset** to reset the staging area to the last commit without disturbing the working directory.

ONLINE GIT REPOSITORIES

In this exercise you will learn about how to set up and use an online Git repository and synchronize your local Git repository with your online repository. At the end of this exercise, you will be able to:

- Set up the online repository as a remote repository for your local Git repository
- Push your commits to the online repository
- Clone an online Git repository to your computer

SETTING UP AN ONLINE GIT REPO

- Sign up for an account either at Bitbucket (https://bitbucket.org) or GitHub

- (https://github.com). Note that private repositories on GitHub requires a paid (or student) account, and is not available for free accounts.
- Then set up an online Git repository named git-test. Copy and save the URL of your online Git repository

SET THE LOCAL GIT REPOSITORY TO SET ITS REMOTE ORIGIN

- At the prompt, type the following to set up your local repository to link to your online Git repository:

```
git remote add origin <repository URL>
```

PUSHING COMMITS TO THE ONLINE REPO

- At the prompt, type the following to push the commits to the online repository:

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

- Depending on your online repo setup, you may have to change “master” to “main”

CLONING AN ONLINE REPOSITORY

- To clone an online repo to your computer (or any other device), type the following:

```
git clone <repository URL>
```