Dordt University
Engineering 304, Embedded Microcontroller Systems, Spring 2023
Problem Set 2, Software Version Control using Git

1   **Establish an account on github.com**.
    If you already have an account on github.com you may use that account.
    Visit https://github.com/ and follow the on-screen dialogs to set up an account.   Choose a businesslike username just in case you end up keeping this account!   Choose a long password since this account will be exposed to the world and you do not want it to get hacked.

2. **Request permission to push**
    Send your Github username to Professor De Boer via e-mail.   Professor De Boer will use your username to give you permission to push to his repository on Github.   You will need that permission to complete step 10 below.   In the meantime you can proceed with steps 3 through 9.

3. **Establish a working folder on your computer**.
    You may use your own computer and create a folder on your own hard drive or on your G: drive,
    or you may use a computer in the EE lab, in which case it would be best to place the working folder on your G: drive.
    Using a Dordt-owned computer has the advantage of not needing an installation of Git on your own computer.   Name the working folder PS-2.   (Yes, other names can work, but for now, make it easy on yourself, use the suggested name!)
    Do not initialize this folder as a repository.   You will do that later using a "`clone`" command.

4. **Install Git SCM on a computer that has access to your working folder.   (If it is not already installed).**
    The recommended URL for the download is https://git-scm.com/downloads
    Once it is installed, start it and set the core editor to something you like.
    For example, choose *Windows Notepad* with this command in Git Bash:
    `git config --global core.editor "%windir%\system32\notepad.exe"`

5. **Point Git Bash to your working folder.**
    Start Git Bash with a right-click on your working folder name or, within the Bash shell, use Unix/Linux commands such as `pwd`, `cd`, `ls -l`, etc. to make the working directory point to your local PS-2 working folder directory, as reported by `pwd`.

6. **Clone Professor De Boer's PS-2 Repository to your working folder, PS-2**.
    All the files in Professor De Boer's Repository will be entered into your working repository.   The command is
    `git clone https://github.com/dfdeboer/PS-2.git .`
    Notice the space and dot (period) on the end of the command.   That causes the clone to be created in Git's current working directory.   All the files in the cloned set of files arrive in tracked-unmodified condition in your local repository.   The clone command will also initialize the local `PS-2` directory as a repository.

7. **Write something to describe one of the git commands**.
    In your local working repository find file `Git_Commands_to_Document.txt`.   Open the file using a text editor and follow the directions that are already in that file.   Notice, a text editor is a program such as Windows Notepad, Notepad++, vi, vim, Sublime Text, etc. (Please: Don't use a word processor like MS Word, Pages, etc! It will mess up the later merge operation.)   After defining just one of the terms in the file, save your working file and exit the editor. The file is now in the tracked-modified state.

8. **Stage the file.**
    Open your local copy of Git Bash (if it is not already open).   Point it to your working directory (if it is not already pointed there).   For good measure, run `git status` and observe the file that needs staging.
    Then perform a `git add *` command.   The file is now staged.

9. **Take a snapshot**.
    For good measure, perform a git status command and observe the staged file.   Then perform a `git commit` command.   Write a comment that includes your name in the editor window that opens.   Save the comment file and close the editor. The commit will then finish execution.   (By including your name you can later identify your commit on Github.)

10. **Push your commit to Professor De Boer's repository**.
    You will need permission from professor De Boer to do this.   See step 2.   The command you need is
    `push origin main -u`
    If you get a message indicating success, your work on this assignment is done.
    If your push is rejected, somebody else has pushed the file ahead of you.   You will have to pull their edits into your file, then add, commit, and push.   Use the `git pull origin Git_Commands_to_Document.txt` command.

11. **Recommended: Watch the file change as others push their files.**
    As others push their files the various files will all get merged into professor De Boer's repository.
    To watch, open a web browser and go to https://github.com/dfdeboer/PS-2 which is the public repository.
    Click on a filename to view the file.