Git Assignment

1. Objectives:

- To introduce to the fundamental concepts of git
- To provide hands-on experience with git commands

2. Tasks to complete:

For this assignment, you need to have an account on GitHub.com and set up a git repository. For the duration of the semester, you will need to update the repository regularly with the practical codes which will be covered in the live demo sessions on Tuesdays.

Follow the steps below which will also be demonstrated in the Q&A session in Week 3. You can access the recording on Brightspace/Canvas at any time.

1. Create a GitHub account.
2. If you haven't already, generate SSH keys on your local machine and add your public key to GitHub.
3. Set up a git repository on GitHub. Name your repository as “sciprog_24”.
4. Create a new directory on your computer and initialise it as local git repository.
5. Configure your git user information.
6. Connect your local repository to the remote repository.
7. Create a new file: “README.md” in your new directory and update your file describing your repository.
8. Check the current status of your git repository, Upload the README file using the git add, commit and push commands.
9. For each practicalx (x=2, 3, ..., 10), locally:
   - 9.1. Create a new branch named practicalx, switch to branch and push it to remote repository.
   - 9.2. Create a folder called practicalx, copy your practicalx source codes from sciprog and create a README file explaining how to compile and run your codes.
   - 9.3. Check status, Upload files using the git add, commit and push commands.
   - 9.4. Merge practicalx branch to master

3. Assessment Criteria:

This assignment will contribute 10% of your total marks to this module, so to ensure that you get full marks.

- All practicals from Week 2 onwards should be included in the repo and the code from each practical should be able to be run and compiled (on sciprog) once cloned by another user. You can use your own solution, or the material covered in the demos. Upload only the source codes, not executables.

- Each practical is clearly commented, and the repository should be well structured and laid out (i.e. have a folder for each practical). This includes having sufficient README explaining the contents of each folder, and instructions on how to use/run the programs (these do not need to be exhaustive, enough information for the reader).

- You should update this repo weekly following each practical with clear and concise commit messages. We are looking for stepwise development by committing code (and comments) on a regular basis.

- At the end, capture the output of the git status and git log commands (e.g. $ git log > gitlog_output.txt will write the output to a file called gitlog_output.txt),

- IMPORTANT: This repository should only be used for the practical sessions covered in the live demo sessions ONLY. **Assignments 1 and 2 should not be added to the repo.** If the assignments are uploaded to your repository, then you will receive 0% for this assignment.

4. Submission Guidelines:

- Provide the URL of your GitHub repository
- Make sure you upload the following on Brightspace/Canvas:
  - output of the git status
  - output of the git log

5. Main References:

- Git documentation: https://git-scm.com/doc
- GitHub documentation: https://docs.github.com/en
- Basic formatting syntax: https://docs.github.com/en/get-started/writing-on-github/getting-started-with-writing-and-formatting-on-github/basic-writing-and-formatting-syntax

Good Luck.