Session 2B Principles of FOSS and Version Control
Exercise 2 – Git Basics

For this exercise, you can work individually or as a group, but either way use your support group to ask questions if you get stuck. Refer back to the lecture slides to help you.

- If you’re using Windows, you’ll need to launch the Git Bash program before starting
- If you’re using Linux or Mac OS, you should just need to open a new terminal (assuming Git is installed)

1. If this is your first time using Git, you should configure your username and email as per the instructions in the slides.

2. Create a new folder somewhere on your machine and switch to it using terminal commands.

3. Create a new Git repository in this new folder.

4. Write a simple piece of code in Spyder, and save the file in your new folder.

5. Check the status of your repository – you should see that Git is aware of your new file, but is not tracking changes.

6. Stage and commit the new file, with an appropriate commit message. Check the status again both post-stage and post-commit.

7. Check the log to see the details of your commit.

8. Make some changes to your code file, and save them.

9. Check the status of your repository again – you should see that Git has identified that the code file has been modified, but notes the changes have not yet been staged for commit.

10. Stage and commit the changes.

11. View the log to look at the commit history. Try using the --oneline option too.

12. Look at the *diff* between the two commits.

13. Make further changes to the code file. Perhaps add a deliberate bug / error.

14. Stage and commit the changes.

15. View the log to look at the commit history.

16. Look at the *diff* between the latest commits.

17. Use a Git Revert to revert the latest commit back to the previous version. Check the log and also flick back to your code in Spyder – you should see it’s flipped back.

18. Try creating a new code file and practice the above on the new file too.