# Worksheet 1 - Git Basic Commands

1. On the command line, create a new Git repository, called *fall2019*.
2. Change directory to the newly created repo.
3. Create an empty file (you can use *touch syllabus.txt*).
4. Check the status of your repo.
5. Add *syllabus.txt* to the staging area.
6. Check the status of your repo.
7. Commit your changes using message “Initial commit.”
8. Check the status of your repo.
9. Open your project directory using a code editor that supports integration with the command line.
10. Open *syllabus.txt* and add some text to it. Save the file.
11. Check the status of your repo.
12. Check the differences in *syllabus.txt* that were not staged yet.
13. Add *syllabus.txt* to the staging area.
14. Check the status of your repo.
15. Unstage *syllabus.txt* but do not delete its contents.
16. Check the status of your repo.
17. Add *syllabus.txt* to the staging area again.
18. Commit your changes using message “Add learning objectives.”
19. Still in the root of your project, create a file called *lecture01.txt*.
20. Add to staging and commit *lecture01.txt* file to your repo.
21. Add a new folder to your project directory, called *Unit01*.
22. Check the status of your repo. Note that Git does not track empty folders.
23. Move *lecture01.txt* file to *Unit01* folder.
24. Check the status of your repo.
25. Commit this change with a meaningful message.
26. Open *syllabus.txt*, add some text, and save the file.
27. Add and commit these new changes with a message “Add a silly idea that breaks everything.’
28. Now that you accidentally “broke” everything, run a *git log* command and copy the hash from before your last commit that did the damage.
29. Use a command to do a hard reset of your project state to the commit with the hash you’ve copied. The last changes in *syllabus.txt* should be gone.