Initialising the Git repository

1. Create a new folder at any location on your machine. Let’s call this folder (or directory) `GitFiles`

2. Open the `GitFiles` folder (or directory) and right-click anywhere inside the folder (or directory) and click Git Bash Here. This should open a command-line interface, ready for typing Git commands.

3. Run the command `git init`

4. You should see a message similar to the following
   `Initialized empty Git repository in C:/GitFiles/.git/`

Creating your first Git-tracked file

5. Create a text file called `Welcome.txt` and add the following lines into it.

   ```
   Welcome to 2019.

   One of my goals this year is to learn to use
   the Git version control software to manage my
   documents and source code.

   This is the first step towards achieving this
   goal.
   ```

6. Save the text file in the `GitFiles` folder (or directory).

7. Now let’s look at the status of our Git repository. Run the command `git status`

8. You should see a message similar to the following

```
On branch master

No commits yet

Untracked files:

  (use "git add <file>..." to include in what
will be committed)

        Welcome.txt

nothing added to commit but untracked files
present (use "git add" to track)
```

9. This is because Git has not yet been told to keep track of this new file.

Adding and committing files to Git

10. Run the command `git add Welcome.txt`

11. Now look at Git's status by running the command `git status`

```
On branch master

No commits yet

Changes to be committed:

  (use "git rm --cached <file>..." to unstage)

        new file:   Welcome.txt
```

12. Git has now become aware of this new file. We now need to commit this change to Git. This saves the file in Git's repository (remember that hidden folder called `.git`)

13. Run the command `git commit -m "Created and committed my first Git file"` and then view Git's status again (see step 7)

Viewing the log

14. Git keeps a record of all the commits that we make. To view this record, run the command `git log`. You should see a message similar to the one below.

```
commit 5609ff3a7d054e837423f2fcf1356dc45c289050
(HEAD -> master)

Author: Chi Onyekaba <c.onyekaba@dundee.ac.uk>

Date:   Sun Jan 6 20:13:44 2019 +0000

    Created and committed my first Git file
```

Each log contains the following information

Commit ID:	The unique identifier of this commit.
Branch info:	The branch* this commit belongs to
Author:	Who made the commit
Date:	The date the commit was made
Message:	The commit message
	* more on branches later

15. Now visit the lecture slides to get more information about some of the options available with the git log command and try them out.

Additional tasks

16. Create two or more new files, add lines of texts to them, save, add to Git using the git add command (see step 10) and commit with a commit message. Then take a look at Git's status messages.

17. Modify the text in one or more of your files, add the modification to Git using the git add command (step) and commit the change. Repeat this process, each time, making changes to one or more files. Once done, view the log to see a history of all the changes.