

## Code and data management with Git

### Git Basics

#### Connect to the education server

Open <http://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html> in your browser. It may be faster to google for putty.

Download and run **putty.exe**, the standalone executable and not the installer (Alternative binary files, the SSH and Telnet client itself).

Introduce the education server IP address **145.88.35.11** as destination.

Make sure that the **port number** is set to **22** and that **SSH** is selected as **connection type** and press “**Open**” to start the connection.

Press “Yes” if the security alert pops up.

Introduce your provided **username** and press enter.

Introduce your provided **password** and press enter.

## 1 General questions

1. How can you check your local git repository state?

- |                    |                |              |
|--------------------|----------------|--------------|
| (a) git state      | (c) git commit | (e) git show |
| (b) git repository | (d) git status | (f) git log  |

2. What command is used to stage a file?

- |                  |                |              |
|------------------|----------------|--------------|
| (a) git stage    | (c) git add    | (e) git diff |
| (b) git checkout | (d) git status | (f) git log  |

3. How do you tell git what files to ignore?

- |   |                               |
|---|-------------------------------|
| (a) git ignore {file-name}                          | (c) it is not possible        |
| (b) add the file paths to a file named “.gitignore” | (d) git not track {file-name} |

4. The **git commit** command:

- |   |   |
|---|---|
| (a) stages all the modified files                                 | (c) retrieves old files into the working directory                            |
| (b) saves the staged files changes into the repository “database” | (d) shows a repository history and lets the user pick what commits to discard |

5. A file is brought into the modified state when:

- |   |   |
|---|---|
| <input type="checkbox"/> it is staged                   | <input type="checkbox"/> the user checks out a previous version of it |
| <input type="checkbox"/> it has been edited by the user | <input type="checkbox"/> the file is committed                        |

## 2 Practical exercises

### Local configuration

First, let Git know what your name and e-mail address is:

```
$ git config --global user.name "Your Name"
$ git config --global user.email "your@email.address"
$ cat ~/.gitconfig
```

### Create a repository

First, create an empty directory inside your home directory and use “`git init`” in that newly-created directory to make a new repository.

- *Hint:* Use a short and descriptive name for the directory.
- *Question:* How can you see that you are working in a Git repository?

Check the status of your repository.

### Your first commit

Create a new file named “README” (with an editor, or by using the “`touch`” command).

- *Question:* What is the status of this new file?

Stage this file for the next commit using “`git add`”.

- *Question:* What is the status of this file now?

Commit your changes.

- *Question:* What happened to the status?

### Manipulation in the staging area

Edit your file. Check the repository status.

Stage your modified file. Check the repository status.

Now remove the file from the staging area. Check the repository status.

Discard the changes performed to the file. Check the repository status.

Edit again the file, stage it, and commit.

### Working with versions

Edit your file again and commit one more time.

Now you change your mind and you want to restore the version stored in the previous commit.

- *Hint:* Use “`git log`” to see a list of all your versions.

Edit your file again and check the differences with your last commit.

- *Question:* What is the difference with your first commit?

Make a file that you do not want to track.