1 Install Git in your account

If you want to use Git (and you do!) you need to install it in your computer. Git is not installed in the labs by default, so you will need to install it in your personal account. You have more than enough space. (Hint: your drive is probably H: or I:).

Remember that Git is free open-source software so you can get it easily online. (Hint: searching ”git windows download” will probably get good results)

Once you have installed Git, check that everything works by going to the relevant folder and executing `git`. You should see a listing of all the possible commands. (Hint: the folder you are looking for is called `bin`, for “binary file”; this is a typical name for the folder that stores the executable files of a project).

2 Add git.exe to your path

If you have successfully completed the previous exercise, you can execute git from its own folder. In a real environment, however, you want to execute Git from your project folder, wherever this is. In order to execute a program from a different folder, you need to add it (e.g. `git.exe`) to your PATH.

2.1 What is the PATH?

Every program that you execute in Windows must be either (a) called absolutely, with its fully qualified name (e.g. `c:\windows\notepad.exe`), (b) be in the current folder, or (c) be included in the PATH.

When you type a command that is not a fully qualified name, Windows will look for it in the current folder. If it is not there, it will look for it in all the folders in the PATH. If it cannot find it after all these steps, it will complain:

```
'.....' is not recognized as an internal or external command,
operable program or batch file.
```

2.2 How can I check what my PATH is?

Very easy. Just type ’path’ on the command line.

2.3 How can I edit the PATH?

The right way of doing it requires to have Administrator rights, so you cannot do it in the lab. You can check online how to do it, it is explained in thousands of places (googling ”change path windows 7/Vista/XP/whatever” will probably get you good results).

In the lab, you have to do it manualy by using the command ’path’ that you have just used to check your PATH. You can set your PATH by typing:

```
> path newPath
```

This will overwrite your PATH. This is usually not what you want: you want to add things to your PATH, keeping the old list of folders. This is easy too; the only thing you have to know is that the old value of PATH can be accessed with %PATH%. In other words, if you type...

```
> path %PATH%e
```

...you will have the same path. Now it is easy to add things. You have probably noticed that folders are separated by a semicolon (;), so if you want to add the folder `h:\git\bin` to your PATH, you only need to type:

```
> path "%PATH%;h:\git\bin"
```

The inverted commas are only necessary if you have spaces in your PATH.

2.4 Notes for Unix users

- In Unix (Linux, Mac OS X, etc), entries in the path are separated with a colon (:), not a semicolon.
- In Unix (Linux, Mac OS X, etc), environment variables (like PATH) are accessed using an initial dollar symbol instead of using enclosing percents, i.e. to see your path, type: echo $PATH