Exercise – Bash Environment

1. Review questions
2. Which command is used to display all variables exported in the current shell ?

set

1. What is the name of the variable that contains the list of directories used to search for programs?

env

1. If the following files existed in a Bash user's home directory, which would be executed on login, and in what order?

**.profile**

**.bashrc**

**.bash\_profile**

**.bash\_login**

.profile

.bash\_login

.bash\_profile

.bashrc

1. Which two builtin commands allow to manipulate and interrogate Bash environmental definitions and options ?

$ Alias

$ set

1. How would you switch on spell checking when changing directories for a Bash user?

$ Set –o spellcheck

1. Control the primary prompt: the PS1 variable
2. Interactively (on the command line) set the **PS1** variable, your primary prompt, to contain your user name (**\u**) and the current location on the system (**$PWD**).

Your future system navigation, i.e. when you use the **cd** command, should be reflected correctly by the prompt: it should always show which directory you moved into.

The prompt can include any separators you choose but should end with the dollar character (to signify the shell belonging to an ordinary user).

The following command line shows the correct assignment string.

**PS1=’\u:$PWD $’**

1. Test you new prompt: use **cd** to move into the **/tmp** directory, and verify that your prompt is showing your new location correctly.

If not, go back to the previous exercise, and set the **PS1** again.

1. Explain why single quotes (rather than double quotes) should be used when constructing the **PS1** assignment line.

*Single quote ignores special characters double quotes would protect them causing the code to execute causing an error*

1. Enter exit – this will close your terminal. Open another terminal session.
2. Write a simple shell script
3. Using an editor, create a file called **myscr1** containing just the three following lines:

#!/bin/bash

hello=“Hello my friend“

cal

echo $hello

1. Now give the file the ***execute*** access permission, and then test it. You can use GUI for the purpose: in the file manager identify the icon for **myscr1**, right-click on it and tick the Execute box in the Properties.   
   Alternatively, the below shows the command line achieving the same.

chmod +x myscr1

1. Test the script - run it, providing the location of the script (the current directory).

*./myscr1 to call the script from terminal*