HW#1 - Recycle Script

Due Date: Feb 07 2016

CIS4361 Spring 2016

This assignment can be worked with a **group of 3 people**. I advise that you find a group (let me know if you need assistance with that) because that way you will be able to develop the team before the harder assignments later in the course come.

**Instructions:**

Write a script (file with a sequence of commands) which you can invoke with the bash shell, either by executing “sh recycle filename” or “./recycle filename” or “recycle filename”. This scripts accepts a filename as first parameter (don’t worry about anything else than a file, I will teach you later how you can handle such scenario).

The filename parameter will be stored in a special variable called $1; if the script were to receive more parameters they would be inside of $2, $3, $4, etc.

The recycle command will move the filename to the Trash folder and create the TrashInfo file. You can see the uploaded video for an example of what and where the Trash folder is located (most likely in ~/.local/share/Trash/[files | info]. Be aware that this folder is not created until you delete your first file using the Ubuntu GUI (Nautilus). You may handle this special case if you feel like learning the if instruction as we will need it for further homework, but it sufficient to assume that the folder is already there.

**Hints:**

The following commands will be of use for this homework (use man <command> or <command> --help to check more usage details):

* **mv src dst:** moves a file from src (original file) to dst (destination path/file)
* **pwd:** shows/returns the current path of the terminal (shell)
* **echo someText:** presents someText at the screen
* **[someCommand] > outputFile:** any output of a command can be written to an external file in this case called outputFile
* **someVar=$(command-name-here)** or **someVar=`command-name-here`**: returns the output of a command into a variable.
  + Notice that these are back single quotes `
    - Single quote ‘text’ is used to interpret the text literally (e.g., echo ‘$var’ would show $var on the screen)
    - Double quote resolves the variables inside of the string (e.g., echo “$var” would show the value of the command on the screen)

**References:**

* <http://ss64.com>
* <http://www.cyberciti.biz/faq/unix-linux-bsd-appleosx-bash-assign-variable-command-output/>