Assessment Pro Forma

1. Module number	CSN08101
2. Module title	Systems & Services
3. Module leader	*Jim Jackson*
4. Tutor with responsibility for this Assessment	*Jim Jackson*
5. Assessment	*Practical Skills Assessment*
6. Weighting	*40%*
7. Size and/or time limits for assessment	*6 hours, 30 minutes.*
8. Deadline of submission	*23:59 Monday 27th November 2017*
9. Arrangements for submission	*Script files must be submitted via Moodle.*
10. Assessment Regulations All assessments are subject to the University Regulations.	*No exemptions.*
11. The requirements for the assessment	*See Attached.*
12. Special instructions	*None.*
13. Return of work	*Feedback is provided via Moodle*
14. Assessment criteria	*See Attached.*

CSN08101 Systems & Services

Linux Practical Skills Assessment

The purpose of this coursework is to allow the student to demonstrate their knowledge of operating system command line usage and the development of shell scripts.

Students will have gained much of the knowledge required to complete the coursework through lectures and labs and some of the previous lab exercises can be utilised in the coursework solution. However, although students will be knowledgeable about the various command line utilities, they should also research the many additional options each command supports in order to best achieve the outcome.

On completion of this coursework you will be able to :

(a) Use the filestore facilities provided by a Linux-based system and particularly, the file attributes and directory structures.

(b) Use the command language associated with a Linux-based system, including some of the commonly used commands and the commonly encountered structures.

(c) Appreciate the strategy associated with the testing of systems software, and the need to build robust products.

(d) Understand and implement 'best practice' housekeeping procedures to allow for user/system misuse or mistakes.

SCENARIO

An on-line news service has amassed a very large number of image files which the business now wishes to archive. At the moment these images are stored on various types of flash memory device.
With no company standard for storing such images, these memory devices have been used by different staff who have created their own sub-directory structures according to their own preference.

Unfortunately they have also used the devices to store other materials including copies and edited versions of the original images.
Therefore, image files may be found in any of the several sub-directories on the flash drive and may even be duplicated in a different folder.
It can be assumed that a duplicate file will always have the same file name as the original.

It is also quite likely that the same file name has been used for two completely different images on occasion.

You have been asked to develop a Linux bash shell script which will reliably copy the specified original images from the flash drive to an archive directory without creating any duplicates whilst also recording the absolute pathnames of any duplicate image files found into a text file called **duplicates.txt** also in the photo archive directory.

It is absolutely imperative for the company's business that no image is ever lost by being omitted or over-written in the photo archive directory.

The script will need to function with flash drives potentially containing tens of thousands of images and should therefore be reasonably efficient in its operation.

SPECIFICATION

The **ph**oto **ar**chiving script must be named “phar” and, should be located in the **~/my-applications/bin** directory in the Cubix1 file system.

It will be used as:

phar flash_pathname archive_dir_path

The locations of the flash drive (images source) and the archive (images destination) can be given as relative or absolute paths.
Examples:

Absolute source and destination.
```
# phar   /mnt/sdb1   /root/my-documents/photoarch
```

Absolute source, relative destination.
```
# phar   /mnt/sdb1   ../imagesarc
```

Relative source, absolute destination.
```
# phar   sdb1   /root/images/archive
```

The examples above cause the script to copy all the relevant image files from **/mnt/sdb1,** and any sub-directories within this directory, to the destination directory. Running the script should not leave the user in a different working directory from where they started.

The script should be efficient, robust and cope with possible user error.

- If the photo archive directory does not already exist, the script must create it.
- If the image source directory does not exist, the script should report this error and terminate.
- If the user omits either of the two arguments then the script should provide a helpful usage message and terminate.
  Example:-
  “Usage : phar image_path archive_path”

2. Only original image files with file names in the form IMG_dddd.JPG are to be archived. (where “dddd” can be any 4 digit number).

3. No sub-directories are allowed inside the destination photo archive directory, only files i.e. it is a flat one-level structure.

4. When copying an image file to the archive directory (e.g. IMG_0001.JPG), if there is already a file with that name in the archive then the second file should be called IMG_0001.JPG.JPG to avoid overwriting the first file, unless the two files are identical, in which case the absolute pathname of one is recorded in the *duplicates.txt* file and the image file is not copied.

IMPLEMENTATION

The script must be documented internally using #comment text. This should include a brief description of the script's purpose and identification of the author (by matriculation number), current version together with a note of recent modifications at the start of the code.
There should also be enough comment in the code to enable an administrator to understand and follow the intended logical operation.

Regardless of how the script(s) are constructed, the user must still be able to perform the archive using:-

```
     # phar image_path archive_path
```

TESTING

The script functionality should be tested thoroughly before submission.
The script should work with any appropriate source and destination directories and any reasonable quantity of image files.

Example test image files will be provided for help with testing.
Students can extract an example set of directory and image files from a zipped archive, available from the file repository using a web browser.

SUBMISSION

The submission is in the form of an electronic copy of the shell script, documented with internal #comments, renamed with the student's matriculation number (e.g. 40345678.sh) and submitted to the Moodle assignment.

MARKING SCHEME

Script documentation in the form of embedded #comments (15)

Exception handling of incorrect/missing user input arguments (20)

Creation of the target directory and copying files according to the file name specification. (15)

Renaming non-identical files having the same original file name; ensuring identical files are not copied. (20)

Creation of the duplicates.txt file containing absolute path references. (20)

General robustness of the script operation. (10)

Total 100

SUGGESTED SCRIPT SKELETON

```
Check user supplied arguments.
        If there are any issues, display the usage message and terminate.

    Obtain a list of paths to all IMG_dddd.JPG files within the source
    directory.
        For every file in this list
              Check to see if there is already a file in the target directory
              with this name.
                         If there is not, copy the file to the target directory as
                         is.
                         If there is and it is not a duplicate:-
                                 • Copy the file to the target directory but
                                   append “.JPG” to the filename.
                         If there is and it is a duplicate:-
                                 • Do not copy the file but record its absolute
                                   path in the file "duplicates.txt".
```