# Shell Programming Assignment 1

**Deadline:** **Thurs. Nov. 12th 2015, 5pm**

**Submit:** **1.** a fully and meaningfully *commented* version of your program

**2.** an uncommented plain text version.

**Send both as attachments to an email to** paul.rothwell@cit.ie I will send a receipt email. If you do not receive an email in reply confirming that I have received your assignment by **Thursday 12th November 2015, 6pm** then you should **send it again.** Be careful to actually attach the file.

**Notes:** **The commented program should be commented fully and meaningfully.** There is room for plenty of difference, assignments that look exactly similar will be assumed not to be original work and will be marked as 0%. The assignment is worth 15% of the overall module marks. Award of marks will be subject to a satisfactory ability to explain your code. You are expected to complete an attempt at doing this assignment with help from the notes, the web, books, other people, etc. However, having presented a program for marking you are also expected to understand how it works.

# Assignment

Write an interactive script that allows a user to copy an existing file from an existing directory to another named new directory. The script should ask the user for:

- the name of a source directory from which to copy a file,
- the name of the file to copy, and
- the name of a destination directory to create and into which to place the copied file.

The script should report errors (and abort the program) where:

1. no source directory name is given;
2. no source file name is given;
3. no destination directory name is given;
4. the source directory name given does not name an existing directory;
5. the source file name given does not name an existing file;
6. the destination directory name given names an already existing directory;
7. the attempt to create the new directory fails;
8. the attempt to copy the file fails.

**Examples**

Here's how some example runs would look (assume the program is called 'ass1' and **user input in bold**):

A successful run would look something like this (**user input in bold**):

```
ass1
Please enter the name of the source directory: source
Please enter the name of the file to copy: myfile
Please enter the name of the new destination directory: destdir
ass1: File myfile from directory source has been copied to destdir.
```

An unsuccessful run would look something like this (**user input in bold**):

```
ass1
Please enter the name of the source directory: source
Please enter the name of the file to copy: myfile
Please enter the name of the new destination directory:
ass1: ERROR: Destination directory not supplied.
ass1: USAGE: Enter the name of a new directory to create.
```

Another unsuccessful run would look something like this (**user input in bold**):

```
ass1
Please enter the name of the source directory: source
Please enter the name of the file to copy: myfile
Please enter the name of the new destination directory: source
ass1: ERROR: Destination directory already exists.
ass1: USAGE: Enter the name of a new directory to create.
```

Another unsuccessful run would look something like this (**user input in bold**):

```
ass1
Please enter the name of the source directory: source
Please enter the name of the file to copy: fileMisnamed
ass1: ERROR: File fileMisnamed does not exist.
ass1: USAGE: Enter the name of an existing file.
```

# Notes:

Errors should be echoed to the standard error channel and adhere to the standard syntax shown in the notes. For example:

```
echo    "${0}: ERROR: Destination directory already exists." 1>&2
echo    "${0}: USAGE: Enter the name of a new directory to create." 1>&2
exit    6       #exit with status error code 6
```

Which, in an example run would display on the error channel as:

```
ass1: ERROR: Destination directory already exists.
ass1: USAGE: Enter the name of a new directory to create.
```

All your exits for errors should be given a unique exit status number 1-255.

# Pseudocode for assignment 1:

```
1. Get name of source directory.
2. Check if name of source directory is blank
3. If so exit with error message.
4. Check that source directory exists.
5. If not then exit with error.
6. Get name of source file.
7. Check if name of source file is blank
8. If so exit with error message.
9. Check that source file exists.
10. If not then exit with error.
11. Get name for destination directory.
12. Check if name of destination directory is blank.
13. If so exit with error message.
14. Check if name given for destination directory names an existing directory.
15. If so exit with error message.
16. Try to create the destination directory.
17. If fail send error message and exit.
18. Try to copy source file from source directory to destination directory.
19. If fail send error message and exit.
20. If successful send confirmation message.
```

**Example Program**

The following example program may help:

A program to accept a filename and some text to look for in the file named. The results of the search are stored in a file called Results in a named directory. If the results directory does not already exist it is created.

```
#!/bin/bash
echo -n "Please enter name of file to search: "
read filename #Get filename
if test -z "${filename}" #if filename not entered
then
    echo "${0}:ERROR:No file name entered." 1>&2
    echo "${0}:USAGE:Enter the name of an existing file to search." 1>&2
    exit 7
fi
if test ! -f "${filename}" #if file does not exist or is not ordinary
then
    echo "${0}:ERROR:Either ${filename} does not exist or it is not an ordinary file." 1>&2
    echo "${0}:USAGE:Enter the name of an existing ordinary file to search." 1>&2
    exit 6
fi
echo -n "Please enter text to search for: "
read searchterm #get search term
if test -z "${searchterm}" #if searchterm not entered
then
    echo "${0}:ERROR:No search term entered." 1>&2
    echo "${0}:USAGE:Enter a search term." 1>&2
    exit 5
fi
echo -n "Please enter name of results directory: "
read resultsdir #get directory
if test -z "${resultsdir}" #if directory not entered
then
    echo "${0}:ERROR:No directory entered." 1>&2
    echo "${0}:USAGE:Enter a name for the results directory." 1>&2
    exit 4
fi
if test ! -d "${resultsdir}"
then #if directory entered is not a directory
  if  test ! -e "${resultsdir}"
```

```
 then  #if directory entered does not exist as a file
  if mkdir  "${resultsdir}" 2>/dev/null
  then  #try to create directory & dump any error messages
   echo "${0}:INFO:${resultsdir} created." 1>&2  #warn
  else  # if not created OK then exit with error
   echo "${0}:ERROR:Failed to create ${resultsdir}" 1>&2
   echo "${0}:INFO:Check permissions or disk space." 1>&2
   exit 3  #kill program with status 1
  fi
 else # (directory entered does exist as a file but is not a directory)
  echo "${0}:ERROR:Failed to create ${resultsdir}." 1>&2
  echo "${0}:INFO:${resultsdir} already exists as a file and is not a directory." 1>&2
  exit 2  #kill program
 fi
fi
if  grep  "${searchterm}" "${filename}" > "${resultsdir}/Results" 2>/dev/null
then  #search for searchterm in filename and output results to screen and Results file in directory
named.
 exit 0  # exit with success
else  #otherwise exit with error
 echo "${0}:INFO:Failed to find ${searchterm} in ${filename}." 1>&2
 exit 1
fi
```