**Lab 1 - Basic Conditionals**

* Due Apr 14 by 2:50pm
* Points 100
* Submitting a text entry box or a file upload

**Objective**: Prompt the user for a number and check to see if it's larger than 5

1. Create a script called lab1.sh
2. Prompt the user to enter a number
3. Check the number to see if it's larger than 5 and output a message
4. If it's not larger than 5 output a message
5. Focus on proper formatting and readability
6. Comment your code if necessary

**Items to use:**

* conditional
* echo
* read

**What to turn in:**

1. lab1.sh

**Lab 2 - Logic and Math**

* Due Apr 14 by 2:50pm
* Points 100
* Submitting a text entry box or a file upload
* Available after Apr 7 at 12am

**Objective**: Prompt the user for a number, ensure it's a number and check to see if it's an EVEN number.

1. Create a script called lab2.sh
2. Prompt the user to enter a number
3. Make sure the input is a number
4. Check the number to see if the number is even/odd
5. If it's even output a message
6. If it's odd output a message
7. Focus on proper formatting and readability
8. It should handle ANY base 10 integers (even 0 or numbers prefixed by 0)
9. Comment your code if necessary

**Items to use:**

* conditional
* echo
* grep  or pattern matching?
* read

**What to turn in:**

1. lab2.sh

**Lab 3 - More Conditionals**

* Due Apr 19 by 2:50pm
* Points 100
* Submitting a text entry box or a file upload
* Available after Apr 12 at 2:50pm

**Objective**:  Check a users ID and if they're root then display a message saying it's insecure to run this script as root. If they are not root then get their home directory from the /etc/password file and display the size of the contents of it.

1. Get the users ID
2. Exit if the user is root
3. If the user is not root make sure their userid is larger than 500
4. Check to see if /etc/password exists and is readable
5. Get the users home directory from /etc/passwd
6. Display the size of their home directory

**Commands to use:**

1. Conditionals
2. BASH environment variables
3. cat
4. grep/awk?
5. echo
6. du

**What to submit:**

* lab3.sh

**Lab 4 - Logical Conditions**

* Due Apr 19 by 2:50pm
* Points 100
* Submitting a text entry box or a file upload
* Available after Apr 12 at 2:50pm

**Objective**: Use logical ANDs and ORs in conditions to simplify your code. Prompt the user for a number, sanitize the input (eg make sure the input is an integer) and using a logical AND to ensure it's between 50 and 100.  Do this with ONE if conditional.

1. Prompt the user for an integer
2. Ensure it's an integer
3. Check to see if it's in the desired range (50-100)
4. Let the user know if it's a success

**What to submit:**

1. lab4.sh

**Lab 5 - Loops**

* Due Apr 28 by 2:50pm
* Points 100
* Submitting a text entry box or a file upload
* Available after Apr 19 at 2:50pm

**Objective:**  Loop through data and act on each entry

1. Loop through the items in medialist.txt
2. Check to see if each item exists in the medialab directory
3. If it does output the name to a new file called foundfiles.txt
4. if it doesn't output the name to a new file called lostfiles.txt
5. save your script as lab5.sh

**Pre-reqs**:

1. copy /var/share/CS225/cs225-medialab.tar.gz to your home directory
2. extract cs225-medialab.tar.gz

**What to upload:**

1. lab5.sh
2. foundfiles.txt
3. lostfiles.txt

**Lab 6 - More Loops**

* Due Apr 28 by 3pm
* Points 100
* Submitting a text entry box or a file upload
* Available after Apr 19 at 1pm

**Objective:** Create a script called lab6.sh that will check to see if each media file in the medialab directory is in medialab/media.xml.

1. List the filenames in the medialab directory that are not in media.xml
2. Display the number of files in the medialab directory that are not in media.xml
3. List the media files by name in media.xml that do not exist in the medialab directory
4. Display the number of files that are in the media.xml that do not exist in the medialab directory

**Pre-req:** Copy /var/share/CS225/cs225-medialab.tar.gz to your home directory and extract it. If you already did this for Lab 5 then skip it.

**Desired Output:**

Files not in media.xml:  
file1.mpg  
file2.mpg  
file3.mpg

Files not in medialab directory:  
file1.mpg  
file2.mpg  
file3.mpg

3 media files in medialab directory that are NOT listed in media.xml  
3 media files in media.xml that are NOT in medialab directory

**Files to upload:**

1. lab6.sh
2. Output from lab6.sh

**Lab 7 - Basename**

* Due May 5 by 2:50pm
* Points 100
* Submitting a text entry box or a file upload
* Available after Apr 28 at 3pm

**Objective:** Make a shell script that takes one argument and duplicates the functionality of the basename command using pattern matching.  
  
**Desired Output:**

lab7.sh /home/bob/test.txt   
test.txt

**What to submit:**

* lab7.sh

**Lab 8 - Pattern Substitution**

* Due May 5 by 2:50pm
* Points 100
* Submitting a text entry box or a file upload
* Available after Apr 28 at 3pm

**Objective:** Using the last line of /var/share/CS225/addresses.csv gather each field into the following variables using bash pattern substitution. Do NOT use awk for this!

* FNAME
* LNAME
* COMPANY
* STREET
* CITY
* STATE
* ZIP
* HPHONE
* WPHONE
* EMAIL
* WEBURL

**What to submit:**

* lab8.sh

**Lab 9 - User input**

* Due May 10 by 2:50pm
* Points 100
* Submitting a text entry box or a file upload
* Available after May 3 at 4pm

**Objective:** Create a script that parses commandline arguments and options.

1. Create a script named lab9.sh
2. The script should take these options
   1. -d - turns on debugging (set -x)
   2. -v - sets the VERBOSE variable to true
   3. -h - prints out a help message with the name of the script (and no path information) plus the options
   4. -n <name> - takes a name argument and prints a message on the screen
   5. -l <dir> - have it list the directory using a for loop or ls (ask yes/no before proceeding)
   6. any other options - prints the help message and exits
3. Puts all additional arguments in $1 $2 etc.

**Needed:**

* while
* case
* getopts
* parameter expansion
* shift
* set

**What to submit:**

* lab9.sh

**Lab 10 - Script Control**

* Due May 17 by 2:50pm
* Points 100
* Submitting a text entry box or a file upload
* Available after May 10 at 4pm

**Objective:** Create a script that does work and cleans up if forced to exit

1. Script should list all files one per line in the entire OS
2. The filelist should be stored in a temporary file
3. If a user presses control-c a function will clean up the temporary file and exit

**Output:**

     File 1:  testfile.sh

     File 2:  testfile2.sh

     ....

**Items to use:**

* traps
* functions
* conditionals
* loops
* mktemp

**What to upload:**

* lab10.sh

**Lab 11 - Function Library**

* Due May 24 by 6pm
* Points 100
* Submitting a text entry box or a file upload
* Available after May 10 at 12am

**Objective:** Start building a function library

**Pre-req:**

copy /var/share/CS225/timefiles.tar.gz to your home directory and extract it

**Create a script that:**

1. looks at the modified timestamp of each file in timefiles
2. creates a directory structure for the month and day in your home directory
3. copies each file to the appropriate directory
4. moves each file to the appropriate directory if -f (force) is set

Script has to include the following

* syntax/help function
* getopts options
* trap for control-c
* function library for generic functions (check timestamp, create directory, print syntax etc..)
* uses pattern substitution

**What to turn in:**

1. lab11.sh
2. functionLibrary.sh

**Lab 12 - Arrays**

* Due May 31 by 6pm
* Points 100
* Submitting a text entry box or a file upload
* Available after May 24 at 12am

**Objective:** Create a script with a pseudo multi-dimensional array

**Create a script that:**

* Reads in /var/share/CS225/addresses.csv and parses the various parts into separate arrays
* Loops through resulting arrays and prints data back onto the screen in the same format as addresses.csv

For example:

"Byron","Fortuna","Jackson & Collins Pa","700 Sw Higgins Ave","Missoula","Missoula"

LNAME[1] should be Byron

FNAME[1] should be Fortuna

.....

When you get done you should have an array for each column and each item number should correspond to a row. Print the results back out by iterating through the NUMBER of rows (get from one of the arrays ${#LNAME[@]}).

Script has to include the following

* trap for control-c
* uses pattern substitution
* uses arrays

**What to turn in:**

1. lab12.sh

**Lab 13 - Regular Expressions**

* Due Tuesday by 6:30pm
* Points 100
* Submitting a text entry box or a file upload
* Available after May 31 at 12am

**Objective**: Create a script with four functions using BASH's in process regular expressions to check the validity of Social Security numbers, telephone numbers, IP addresses and credit card numbers based on each particular data format.

1. Take a piece of data as an argument and echo back it's type based on the results of these functions.
2. Each function will be named check\_ip, check\_ssn, check\_pn and check\_ccn.
3. Each function will take one argument - the data in question
4. Each function will return a message and a 1 or 0 depending on the outcome of the comparison
5. Make sure the script traps control-c and exits
6. Make sure the script prints out help if no argument is given.

**What to submit:**

* lab13.sh