Mod 06 BASH Arrays & Advanced Flow Control Homework

SCRIPTING ESSENTIALS

DR. BURKMAN

Read through the entire homework problem before starting.

Cards!

We want to make our card deck script again but BASH doesn't support nesting arrays. We can force that, but it's ridiculously hard. Instead we will thing in terms of suits.

Download the script starter "Mod_06_hwstarter.sh" from the sftp server. Do chmod 750 " from the sftp server. Do chmod 750 Mod06Homework_starter.sh to give it the correct set of permissions.

Cards!

We will make this script cleaner than we did in Python. The menu will be in a while true loop. Option 1 only calls the function to get a card. Option 2 only calls the function to make a new deck. Option 3 clears the screen then breaks out of the loop. Otherwise it catches an invalid selction, pauses for user entry, then returns to the menu. The menu does nothing else. Use echo like you would a Python multistring to set up the menu.

Flow

The big picture flow:

- Declare and build the four suits for the deck
- Declare the four play suits
- Function to get a new deck
- Function to draw cards
- Call new deck function
- Menu loop

Declare and build the four suits for the deck

I’ve done this for you with the starter

Declare the four play suits

These are just four lines declaring empty play suits. We will have the regular suits and a copy suit for playing.

Function to get a new deck

This has four lines to copy each original suit to a play suit. One line each for spades, diamonds, clubs, hearts. Also initialize four variables to zero. Name these like “spades_gone”, etc.

Function to draw cards

This is obviously the bulk of the code. The big picture flow for this function:

- Get the number of cards from the user
- Check input validity
- See if there are enough cards remaining to meet the user request
- Loop and randomly get the cards, keeping track of pulled cards

Function to draw cards
Get the number of cards from the user and check input validity

Get the number of cards from the user per my demo output. Check to see if the input is a valid integer.

Here's something a bit new. To check for the opposite condition we put the ! outside the brackets. Like this:

- If ! [[$request =~ ^[+]?[0-9]+$]]; then

If that is true tell the user the option is invalid (per my demo output), pause for input then use 'return' to leave the function. That will put the user back at the main menu. Just if/fi, no elif or else needed.

Function to draw cards
See if there are enough cards remaining to meet the user request

Create a variable to hold the sum of the length of each of the four play suits. That's the number of cards remaining

If the user request is greater than the number of cards remaining inform them (see my demo) and use return to leave the function.

After the fi you can echo "Your cards are:"

Function to draw cards
Loop and randomly get the cards, keeping track of pulled cards

This all goes in a while true loop that uses the number of cards requested as a counter. After the 'do':

- Assign a random value between 0 and 3 to a variable. I'll call that variable suit. Now do an if/elif/elif/else/fi. Each of those is a suit. Here is the flow for one suit (the other three are basically the same)
  - If suit = 0 then #or 1, 2 or 3 (I'm doing spades in this one)
  - Suit_length = get the length of that suit
  - If that length is zero, set spades_gone to 1 then continue (or diamonds_gone, etc)
  - Else a new variable (I'll use card) equals a random number between zero and one less than the length of that suit.
    - RANDOM%length of the suit
  - Echo that element of the play suit
    - Echo ${play_spades[$card]}
  - Look at slide 6 of the lecture. Remove the card that was picked. Be sure to remove the element, not just empty it
  - Reduce the number of cards requested by one.
- fi

Function to draw cards
Loop and randomly get the cards, keeping track of pulled cards

After the while statement ‘done’, echo and do a user read “Press enter to continue”

So this function has a big while, with if/elif/elif/else/fi. Inside each of those is an if/fi.

Menu loop

In between your function to draw the cards and the loop for your menu be sure to call the function to initialize your first new deck.

Slide 4 and the demo should give you enough info to make the menu.