

CS306 – Introduction to Perl
Fall 2007
Homework Assignment #2
Due: Monday, October 8th, 2007, 4pm

Guidelines for Submitting Homework

All homework should be submitted in the form of a zip file which contains one program for each question. Your zipfile should be named *lastname-hw2.zip* and should contain one text file called *answers.txt* for the written answers, and then files named *hw2p1.pl*, *hw2p2.pl*, *hw2p3.pl*, etc... for the programming assignments. **Make sure your name also appears in all source code files!** Email this zip file to cs306@cis.uab.edu. **When emailing this file email a copy to yourself so you know the email went through ok.**

Homework #2 (200 total points)

Question 1. (8 points) What's the difference between a list and an array?

Question 2. (8 points) Which functions would you use to implement a LIFO, or “Last In First Out”, queue using an array?

Question 3. (14 points) Give an example of looping over an array by calculating the index on each iteration. Then give an example of looping over the same array without needing to calculate the index on each iteration.

Question 4. (24 points) Please indicate if the following operations are in scalar or list context.

- `$foo = 8 + 5;`
- `$string = “Hello and “ . “goodnight”;`
- `@sorted = sort @array;`
- `($a, $b) = localtime()[5,7];`
- `$foo = @array;`
- `print $foo;`

Question 5. (10 points) We've seen the technique where you can assign into an unrelated list of scalars simultaneously like this:

```
($a, $b) = localtime()[5,7];
```

What happens if there are more scalars on the left hand side than there are elements in the list on the right hand side? What happens if there are less scalars on the left hand side?

Question 6. (24 points) Which data structure would be better to use in each example, an array or a hash?

- For each member of the class, you want to record their exam grade.
- You would like to keep a list of numbers in order to calculate their sum.
- You are reading in words one per line from STDIN and you eventually want to concatenate them into a single string.
- You are storing data about various vital statistics for an individual.
- You want to keep track of how many times each word appears in a large stream of text.

f. You give an exam worth 100 points and want to keep track of the distribution of the scores of the class.

Question 7. (12 points) True or false.

- a. Hashes are unordered.
- b. You can always assign an array into a hash.
- c. Setting a hash key's value to undef does not remove the key from the hash.
- d. Hash values must be unique.

Notes on programs: While writing these programs below, remember to comment thoroughly, including your name near the top as well as comments throughout explaining what you are doing in your program.

Also, turn on warnings and use strict; in all of your programs. Here's how:

```
#!/usr/bin/perl -w
use strict;
```

The /usr/bin/perl part may change depending on your platform; it's the -w that is important, as it is what enables warnings.

Read each problem description carefully, and be sure to address everything that is asked for in the problem.

A NOTE ON DEBUGGING: WHEN YOU GET STUCK, PRINT OUT YOUR VARIABLE VALUES AT VARIOUS POINTS IN YOUR PROGRAM. THIS IS OFTEN THE EASIEST WAY TO FIND WHERE THE CODE IS BROKEN.

Program 1. (20 points) TODO list.

Using arrays and the functions which operate on them, you will implement a simple TODO list. Write a program that presents the user with a menu that has three options: 1. add an item to TODO list, 2. take next item from TODO list, and 3. Exit the program. If the user chooses #1, ask them to enter a short string to represent the item on the list. If the user chooses #2, give the user the oldest item on the list, and remove it from the list. If the user chooses #3, exit the program, printing out any remaining items on the TODO list in the order they were added to the list. The program should keep running until the user chooses option #3.

Program 2. (30 points) Deck Shuffler.

Your task is to create a deck of cards and then to shuffle the deck. Start out by populating your deck (an array of card values) with the following values:

Qty	Card Value
4	2
4	3
4	4
4	5
4	6

4 7
4 8
4 9
16 10 (represents the 10s, Jacks, Queens and Kings)
4 11

That gives you a 52-card deck. Then you need to shuffle the cards. To shuffle the deck use the following procedure.

1. Start with a deck of 52 unshuffled cards and an empty array to hold the shuffled cards.
2. Pick a random number between 0-51. Choose that card from the unshuffled deck and add it to the shuffled deck.
3. Remove the chosen card from the unshuffled deck, leaving an unshuffled deck of 51 cards and a shuffled deck with only one card in it. Hint: we learned a technique for selecting out multiple elements of an array into a new array. You'll want to select all of the UNchosen cards back into the unshuffled deck.
4. Repeat steps 2 and 3, remember this time you have one fewer card in the unshuffled deck, which affects the range of random numbers you want to generate.

You can do this with only two variables: @deck representing the original, unsorted deck, and @shuffled representing the sorted deck. You do not need any temporary variables or scalars, but if you find it easier to use them, you may do so.

This program should first print out the contents of the unshuffled deck and then print out the contents of the shuffled deck.

BONUS: (20 pts) Modify your simple Blackjack program from HW1 to have a deck of cards which are shuffled in this manner, and then use that deck to hand out cards during the game. Turn this in as hw2p2BONUS.pl in your homework submission – turn in a standalone deck shuffler as hw2p2.pl.

Program 3. (20 points) Car Configurator.

Using a hash structure called %car, write a program to allow a user to configure a custom-built car. You will ask the user a series of questions, and store their answers as name=>value pairs in the %car hash. At the end of the program, print out a summary of their custom car. Here is the list of questions you must ask:

What is the make and model of your car? Pentium Flash GTX
What is the color of your car? Blue
What is the general body shape of your car? Sports coupe
What is the horsepower of your car? 250
What is the seating capacity of your car? 4
What is the cargo capacity of your car? 12 cubic feet
How many miles per gallon does the car get? 20
How many gallons of gas does the tank hold? 10

And here is the output:

Make: Pentium Flash GTK

Color: Blue
Body: Sports coupe
Horsepower: 250
Seating Capacity: 4
Cargo Capacity: 12 cubic feet
MPG: 20
Tank Capacity: 10
Fuel Range: 200 miles

Notice that the program calculated the fuel range for the car, your program must do the same.

All data must be stored in the %car hash.

Program 4. (30 points) Hashercise.

The goal of this program is to keep a data structure that represents specific pieces of data, or facts, about a movie. The program will present a menu or prompt to the user like this:

(E)nter a fact, (R)emove a fact, (L)ookup a fact, (P)rint all facts:

You will store all of the movie facts in a hash. A sample run looks like this:

(E)nter a fact, (R)emove a fact, (L)ookup a fact, (P)rint all facts: E

What's the name of this data point: title

What's the value of title: The Princess Bride

(E)nter a fact, (R)emove a fact, (L)ookup a fact, (P)rint all facts: E

What's the name of this data point: Release Date

What's the value of Release Date: 9/25/1987

(E)nter a fact, (R)emove a fact, (L)ookup a fact, (P)rint all facts: E

What's the name of this data point: Main Actor

What's the value of Main Actor: Cary Elwes

(E)nter a fact, (R)emove a fact, (L)ookup a fact, (P)rint all facts: L

What would you like to know? Release Date

The value for Release Date is 9/25/1987

(E)nter a fact, (R)emove a fact, (L)ookup a fact, (P)rint all facts: L

What would you like to know? Main Actress

I'm sorry, I have no value for Main Actress

(E)nter a fact, (R)emove a fact, (L)ookup a fact, (P)rint all facts: P

title: The Princess Bride

Release Date: 9/25/1987

Main Actor: Cary Elwes

(E)nter a fact, (R)emove a fact, (L)ookup a fact, (P)rint all facts: R

Which data point would you like to remove? Main Actor

OK.

(E)nter a fact, (R)emove a fact, (L)ookup a fact, (P)rint all facts: R
Which data point would you like to remove? Genre
I'm sorry, there is no entry for Genre

(E)nter a fact, (R)emove a fact, (L)ookup a fact, (P)rint all facts: P
title: The Princess Bride
Release Date: 9/25/1987

All of your data about the movie must be stored in a single hash.