

CS306 – Introduction to Perl  
Fall 2006  
Homework Assignment #3  
Due: November 20<sup>th</sup>, Noon

Please follow the previous guidelines on how to construct the zip file and submit the homework.

### Homework #3

**Question 1. (10 points)** What are two different ways to run an external program in Perl? What are the key differences between the methods?

**Question 2. (10 points)** Explain the concept of reference counting and how it relates to Perl's memory management.

**Question 3. (10 points)** Given the arrayref:

```
$arrayref = ["fee", "fi", ["fo", "fum", ["eeny", "meeny", "miny", "moe"]]];
```

**Question 4. (10 points)** What are the three different syntaxes for dereferencing the element containing "meeny"?

**Question 5. (10 points)** Define and give an example of autovivification. Be careful here – make sure that what you are doing is really autovivification.

### **Program 1. Tic Tac Toe (50 points)**

Write a program to play Tic Tac Toe. Both players will be human, the program simply takes moves, checks that they are valid, and updates the game board. The game board must be represented and accessed by a single variable (either an array or an arrayref, your preference).

BONUS (10 Points). Make the game check for victory conditions as well.

## **Program 2. Family Tree (100 points)**

You will be given a data file with lines in the following format:

FirstName|LastName|DateOfBirth|Parent

Your goal is to read this input file, construct a multi-level hash of the family tree, and then output the results. To make your job easier, you can assume that all Firstname's will be unique and that a child will never appear in the data file before their parent. We will also not be incorporating spouses into the tree.

You should construct a base hash for each person – this is the person “object”. This object should look like:

```
FirstName => Jane
LastName => Smith
DOB => 06/23/1957
Children => [child1ref, child2ref, etc...]
```

You should then construct a hierarchy of such objects, connecting parents with their children through the Children array.

As soon as you are done constructing the family tree, you should use [Data::Dumper](#) to print out the entire data structure to a file called tree.txt.

Finally, enter a loop where you ask the user for a first name, and then display that person and their children, sorted in order of their birth. For instance:

```
Enter first name: Nancy
Nancy Forrester (5/7/46)
  Jessica Forrester (1/24/74)
  Craig Forrester (7/16/77)
Enter first name: etc....
```

**BONUS.** (15 points) Print -all- descendants of the name entered. This will require a recursive algorithm.