

CS306 – Introduction to Perl
Summer 2005
Homework Assignment #5
Due: July 6th, 2005, 5pm

The homework should be submitted in the form of a zip file which contains one program for each question. Your zip file should be named *firstname-lastname-hw5.zip* and should contain files like hw5q1, hw5q2, hw5q3, etc... Email this zip file to cs306@cis.uab.edu.

You should follow good Perl design principles. Avoid global variables, use subroutines, comment the code well and indent the source code appropriately. You should run under 'use strict' and 'use warnings'.

1. Grade Calculator (5 pts). You will be provided with a sample hash of hashes that holds data about fictional students in a class. You are to traverse the hash of hashes and calculate an average grade for each student **and** for each quiz. Print the results to the screen. Some of the quizzes were optional, so each student took a different number of quizzes. The sample file will be available on the course web site.

2. Team Lister (5 pts). You will be given a text file where each line contains a team name, some white space, and a l-separated list of player names. You are to build a hash of arrays. The keys should be the team names and each key should reference an array of team member names. After the data is loaded, print the data back to the screen with the teams in alphabetical order and the players in alphabetical order within each team:

```
ATeamName  
  AplayerName  
  BPlayerName  
  . . .
```

```
BTeamName  
  APlayerName  
  . . .
```

Finally, write out an ordered.txt file that has all the data in this alphabetical order. The input text file will be available on the course web site.

3. Tic-Tac-Toe (10 pts). Write a tic tac toe game. Store the board as a two-dimensional array. Display the board like this:

```
  0  1  2
0 X  -  -
1 -  -  -
2 O  -  -
X's Turn.  Move?  (User enters something like "1,1" here)
```

```
  0  1  2
0 X  -  -
1 -  X  -
2 O  -  -
O's Turn.  Move?
```

There are 8 combinations of three array elements that could lead to winning conditions (the three rows, the three columns and the two diagonals). Write a function to check for a winning condition after each move. It's probably easiest to simply hardcode the winning scenarios since there are a limited number of them.