CS 150 Programming Assignment 7 Due Tuesday 11/1/11 @ 11:11pm

There are 5 questions, worth a total of 50 points.

All of your files must have exactly the filename listed in the assignment. Each file should have a comment at the beginning with your name and the date.

grep.py 10 points

This is Exercise 8.11. Write a program grep.py that asks the user for a search term, then asks the user for a filename, and then proceeds to print all lines of the file which contain the search term. For example:

```
Search term? Great Britain
Document? declaration.txt
The history of the present King of Great Britain is a history of repeated
of Great Britain, is and ought to be totally dissolved;
```

youngest.py 10 points

Write a program that prints the names of the youngest people in a file famous.txt listing names and birth years. See Exercise 8.9 for details.

randompeople.py 10 points

Write a program that creates a file famous.txt with 512 lines. Each line should have a randomly generated person's name, followed by a random year in the range 1600-2000.

stock.py 10 points

Write a program stock.py that reads a file of stock prices and creates a graphical image of the daily values. This is Exercise 8.20 except our data files contain one line per day with the date, opening price, high price, low price, and closing price, in that order, separated by tab characters '\t'.

Your program will need to adjust the width, height, and spacing of the bars to fill most of a 512x256 graphics Canvas.

HighScores.py 10 points

Write a class HighScores that manages a high score list for a game. The class should keep track of the top ten scores and the associated player names (which can be arbitrary strings). The class should implement the following methods:

```
__init__(self)
    Create an empty high score list with room for 10 entries.
addScore(self, score, playername)
    Add a new score to the high scores list, if it qualifies.
display(self)
    Display the score list as text, sorted from highest to lowest score.
isHighScore(self, score)
    Returns True if <score> would make it onto the high score list, False otherwise
load(self, filename)
    Load high scores list from file given by <filename>.
    Destroys any existing score information.
save(self, filename)
    Save high scores list to file given by <filename>
```