**Object-Oriented Programming in Java**
**MISM/MSIT 95-712**

**Assignment 3**

***This Assignment is worth 3.5% of the overall assessment for this course.***

1. (40 points) Remember the concentration game that you might have played as a kid with some cards. The idea of the game is to find identical pairs among a shuffled pack of cards laid out. For example, let us assume that you are given 10 cards in the deck, with two Aces, two Queens, two 5's, two Jacks and two labeled 9. The cards are shuffled and placed face down on the table.

A player then selects two cards that are face down, turns them face up, and if the cards match they are left face up. If the two cards do not match they are returned to their original face down position. The game continues until all cards are face up.

Write a program that plays this game of concentration. Use 16 cards that are *shuffled* and laid out in a 4 by 4 square. These cards should be labeled with pairs of card numbers (A, Q, K, J, 2, 5, 6, 9).

Your program should allow the player to specify the cards that she would like to select through a coordinate system.

For example, in the following layout:

```
   1  2  3  4
1  A  S  $  $
2  $  $  $  $
3  $  A  $  $
4  $  $  $  $
```

All of the cards are face down indicated by $. The pairs of A which are face up and at coordinates (1,1) and (2,3). To hide the cards that have been temporarily placed face up, output a large number of newlines to force the old board off the screen (or find something better in the JAVA API).

2. (40 points) Once upon a time in the land of Asgard there lived 3 wizards Gandalf, Merlin and Dumbledore. These 3 wizards often had an argument over which one of them was the greatest software developer of all time. To end the argument once and for all, they agreed on a fight to the death. Gandalf was a poor shooter and only hit his target with a probability of 1/3. Merlin was a bit better and hit his target with a probability of 1/2. Dumbledore was an expert marksman and never missed. A hit means a kill and the person hit drops out of the duel.
To compensate for the inequities in their marksmanship skills, the three decided that they would fire in turns, starting with Gandalf, followed by Merlin, and then by Dumbledore. The cycle would repeat until there was one man or creature standing, and that man or creature would be the greatest software developer of All Time.

An obvious and reasonable strategy is for each wizard to shoot at the most accurate shooter still alive, on the grounds that this shooter is the deadliest and has the best chance of hitting back.

Write a program to simulate the duel using this strategy. Your program should use random numbers and the probabilities given in the problem to determine whether a shooter hits the target.

Create a class named **Fighter** that contains the wizard's name and shooting accuracy, a Boolean indicating whether the fighter is still alive, and a method **shootAtTarget(Fighter target)** that sets the target to dead if the fighter hits his target (using a random number and the shooting accuracy) and does nothing otherwise.
Once you can simulate a single fight, add a loop to your program that simulates 10,000 fights. Count the number of times that each contestant wins and print the probability of winning for each contestant in a tabular format. For example:

| **LEADERBOARD – AFTER 10000 DUELS** | | |
|---|---|---|
| **Contestant** | **Number of Wins** | **Winning Percentag** |
| **Gandalf** | **3595** | **35.95%** |
| **Merlin** | **1355** | **13.55** |
| **Dumbledore** | **5050** | **50.50%** |

*An alternate strategy is for Gandalf to intentionally miss on his first shot.* Modify the program to accommodate this new strategy and output the probability of winning for each contestant after 10000 fights.

What strategy is better for Gandalf, to intentionally miss on the first shot or to try and hit the best shooter? Who has the best chance of winning, the best shooter or the worst shooter?

3. (20 points)

One common security method requires that the amount on a cheque be written in numbers and spelled out in words as well. Even if someone is able to alter the numerical amount of the check, it is extremely difficult to change the amount in words.

Write an application that asks the user to input a numeric check amount and writes the word equivalent of the amount. For example, the amount 112.43 should be written as ONE hundred TWELVE and 43/100. A sample run:

**Please enter a dollar amount:**

**19.50**

**Dollar amount in words: NINETEEN and 50/100**

java.util.StringTokenizer will be useful in developing the above application.

**For every homework assignment, Blackboard will allow only one file to be submitted. Also, once you have submitted a file to Blackboard, you are not allowed to remove it and re-submit another file. Because of this, you should be *absolutely sure* that you are finished with the homework and that it is correct before you submit it.**