CS211	Computers and Programming	Handout 2
	ASSIGNMENT 1: Introduction to Java Programming	
Fall 1999		**September 2, 1999.**

DUE DATE: September 9.

Exercise 1

The table below shows the result of voting in a contest. The actual contest is not important for this exercise. A row in the table indicates the votes given by the jury of a particular state. A column in the table indicates the votes received by the contestant of a particular state.

		Contestant							
		IOWA	UTAH	HAWAII	MAINE	IDAHO	TEXAS	OHIO	OREGON
Jury	IOWA		1	2			3	4	6
	UTAH	2			6	3	1	4	
	HAWAII	6	3		4		2		1
	MAINE		3	4		2	6	1	
	IDAHO	1	6	2				4	3
	TEXAS	4	1		2	6		3	
	OHIO		2	6	3	4			1
	OREGON	2			4	3	6	1	

Rules for voting by a jury are as follows:
- 6 votes to the best contestant.
- 4 votes to the next best contestant.
- 3 votes to the third best contestant.
- 2 votes to the fourth best contestant.
- 1 vote to the fifth best contestant.

The jury of a state cannot vote for its own contestant, and it must vote by the above rules. A contestant cannot receive votes from the same jury more than once. A jury has to vote for exactly five contestants.

The winner of the contest is of course the contestant(s) that receives the highest number of votes. Note that in ranking the contestants by votes, there might be a tie among several contestants. In such a case, the appropriate positions in the ranking are skipped. For example, if two states tied for the third place, the fourth position will be skipped and ranking continues from ranking position number five.

The votes which contestants receive from the juries can be read from a single file as triplets (state of the jury, state of the contestant, non-zero number of votes). Each triplet is interpreted as the state jury giving the state contestant the specified number of votes. Each line in the file specifies the state of the jury and the state of the contestant as words, and the number of votes as an integer. For example, the voting of the jury from Iowa would be specified as follows in the vote file:

```
IOWA   UTAH     1
IOWA   HAWAII   2
IOWA   TEXAS    3
IOWA   OHIO     4
IOWA   OREGON   6
```

(See the example for file processing at the end of this assignment.)

Write a program that, based on the voting, prints out the results of the voting. The name of the vote file should be specified to the program on the command line. A sample report is given below.

```
Voting Report
Voting was legal, no rules were violated.
All the juries voted for 5 contestants.
No jury gave its own contestant any votes.

The winner of the contest is MAINE.
The winner contestant from MAINE received a total of 19 votes.
The winner contestant from MAINE received votes as follows:
UTAH            6 votes
HAWAII          4 votes
TEXAS           2 votes
OHIO            3 votes
OREGON          4 votes

Ranking:
1.  MAINE       19 votes
2.  IDAHO       18 votes
    TEXAS       18 votes
4.  OHIO        17 votes
5.  UTAH        16 votes
End of voting report.
```

Exercise 2

Write a class called `SingleBuffer` which implements the following abstraction: it stores a single item (of type `Object`) and has a data member that indicates whether the buffer is logically empty. It provides two methods: `put()` and `get()`. The `put()` method may be only applied to an empty buffer, and it inserts an item into the buffer. The `get()` method may be applied only to a non-empty buffer, and it deletes and returns the contents of the buffer.
Write a client to test the buffer.

Exercise 3

The program below has several errors. Identify, explain and correct the errors. No code lines should be removed, and no new code lines should be added. The program when compiled and run successfully should print the following result on the terminal:

```
74.3F = 23.5C
```

```
// Filename: Temperature.java
import java.util.*;
package com.acme;

interface IFtoC {
    double f2c(double fahr) {};
}

class TemperatureConverter extends IFtoC {}

PUBLIC CLASS temperature implements TemperatureConverter {

    PUBLIC void main(string argsList) {
        double fahrenheit = 74.3;
        */ Convert /*
        Temperature tmp;
        double celsius = tmp.f2c(fahrenheit);
        System.out.println(fahrenheit + ’F = ’ +
                                   celsius + ’C’);
    }

    double f2c(float fahr) {
        RETURN (fahr - 32) * 5 / 9;
    }
}
```

Exercise 4

a) Write a program that prints all the combinations for a 3-digit combination lock. In a combination, all the digits might not be unique. How many combinations do we have for a N-digit combination lock?

b) Write a program that prints all the permutations for a 3-digit combination lock. In a permutation, all the digits are unique. How many permutations do we have for a 10-digit combination lock?

File Processing using CS211 I/O Package

The following files comprise the CS211 I/O package:

```
CS211InInterface.java
CS211In.java
CS211OutInterface.java
CS211Out.java
```

Your should compile the file below with the files in the CS211 I/O package.

```
//  Filename: TestIo.java
/*
    This code shows a simple example of the use of the CS211 I/O
    routines. It expects an input file name as a command line
    argument. It reads two words and an integer from that file,
    and prints something out on a file named Puzzle.output.

    If the input file contains
     James Bond 007

    the output file contains
    ***** Eyes only *****
    The code name for James Bond is 7

    Appropriate error messages are printed if the file does not
    begin with two words and an integer.
*/

import java.io.*;

class TestIO {

    public static void main (String[] args) {
        if (args.length == 0) {
            System.out.println(
                "Error: you must specify an input file name");
            return;
        }
        // Get the file name
        String inFileName = args[0];

        // Reading from a file
        CS211In fIn = new CS211In(inFileName);
        String firstName = fIn.getWord();
        String lastName  = fIn.getWord();
        int codeName = fIn.getInt();
        fIn.close();
```

```java
        // Writing to a file
        CS211Out fOut = new CS211Out("Puzzle.output");
        fOut.println("***** Eyes only *****");
        fOut.print("The code name for " + firstName + " " +
                   lastName + " is ");
        fOut.println(codeName);
        fOut.close();
    }
}
```