FINAL EXAMINATION KEY ♦ DECEMBER 2009

CSC 112 ♦ SECTION 01

INSTRUCTOR: NICHOLAS R. HOWE

YOU MAY USE TWO 8.5"x11" SHEETS OF NOTES ON THIS EXAM.

YOU MAY NOT USE THE TEXTBOOK, A COMPUTER, OR ANY OTHER INFORMATION SOURCE BESIDES YOUR TWO PAGES OF NOTES.

All work should be written in the exam booklet. Partial credit will be granted where appropriate if intermediate steps are shown. When you are done, please sign the statement below. Good luck!

1. **Data Structure Selection** (16 points)

Sometimes more than one data structure can serve in a given application. However, they are not necessarily equally efficient or well suited. Please describe the considerations you would use in order to choose between the following data structures (assuming both will work for a given application). You should cite items such as memory requirements, time required for common operations, etc., and explain which data structure choice has the advantage in each area.

a.) ArrayList and HashMap for a dictionary application

b.) ArrayList and LinkedList to store elements in sequence

c.) Binary Search Tree and Array (used as a simple map or lookup table) for storing items in a sorted order

d.) ArrayDeque and LinkedList for a queue

2. **Recursion** (12 points)

You have spent the day baking mince pies with your friend. Unfortunately, once the pies are in the oven she discovers that she has lost her gold ring, which she had taken off before starting. You suspect it has accidentally been baked into one of the pies. Your friend is distraught, but you tell her not to worry. Fortunately, you have studied recursion!

Assume that the n pies are all identical in weight except for the one with the ring, and that you have a balance available that can compare two sets of pies to each other and indicate which is heavier. Please formulate the search for the ring as a recursive problem, by giving (a.) a problem statement, (b.) a stop condition, and (c.) the location of the ring based on the answer to a simplified version of the problem. You do not need to write this as code, although you may find it helpful to do so. Please write an efficient solution for full credit.

3. **Binary Tree Traversal** (12 points)

The three traversals below come from the same tree. From the traversals, infer the position of the nodes in the tree, and draw a diagram of the three structure.

Preorder: G,I,B,F,K,D,C,E,J,H,A
Inorder: I,F,B,D,K,G,E,J,C,A,H
Postorder: F,D,K,B,I,J,E,A,H,C,G

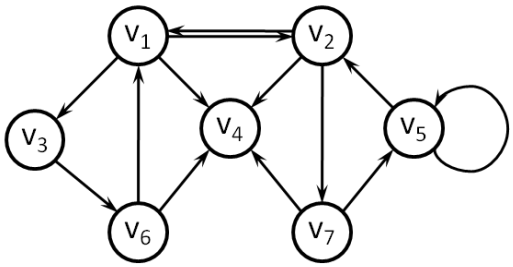
4. **Program Reading** (12 points)

Consider the program below with a critical eye. It is intended to print six random numbers from 1 to 40. Identify the flaws in the program, both functional and stylistic.

```
/**
 *  A class to choose lotto numbers
 *
 * @author N. Howe
 * @version 10 December 2009
 */
public class LottoPicks {
    private int picks[];

    /** maximum number value */
    public static final MAX_VALUE = 40;

    /** pick six numbers for the ticket */
    public static void pickNumbers() {
        for (int i = 0; i <= 6; i++) {
            picks[i] = (int)Math.ceil(Math.random()*40);
        }
    }

    /** Picks numbers and prints them */
    public static void main() {
        pickNumbers();
        System.out.print("Quick lotto picks:  ");
        for (i = 0; i < picks.length; i++) {
            System.out.print(i+" ");
        }
        System.out.println("");
    }
}
```

5. Graphs (8 points)

The diagram below shows a single disconnected directed graph. Given the picture, write down the requested equivalent representation.



a.) Adjacency matrix

b.) Edge matrix (assume head entries are positive, and tail entries are negative.)

6. Java Memory Structures (12 points);

Consider the class below. Draw a diagram showing the state of memory just before the call to `return` in the method `messy()` below. Your diagram should show the stack frame for `messy()` and all the structures it references on the heap, using the style we have employed in class.

```
public class Ref {
    public Ref a;
    public Ref b;
    public Ref() {
        a = b = null;
    }
    public Ref(Ref a, Ref b) {
        this.a = a;
        this.b = b;
    }
    public static Ref messy() {
        Ref f = null;
        Ref g = new Ref();
        Ref h = new Ref(f,g);
        f = new Ref(f,f);
        g.a = g;
        g.b = f;
        return h;
    }
    public static void main(String[] args) {
        messy();
    }
}
```

7. Class Design (12 points)

Write a short essay describing the best practices for class design when planning out a programming project. (This is an open-ended question with no specific answer in mind. Responses will be graded subjectively, based in part on the cogency of the response and in part upon how much they demonstrate a grasp of accepted class design principles.)

8. Sorting (8 points)

Consider the array of numbers below, which are to be sorted in increasing order from left to right. Simulate the array version of the algorithms specified, and show the state of the array after each swap performed.

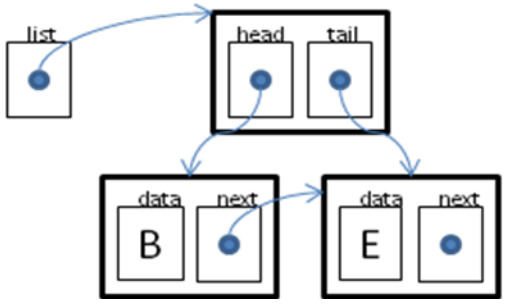
8, 3, 11, 9, 2, 7

a.) Insertion sort, array implementation, growing the sorted region <u>from right to left</u>.

b.) Heap sort, growing the sorted region from left to right.

9. Lists, Stacks, and Queues (8 points)

Consider the singly-linked list shown at right. Draw the new state of the list under the following conditions (both questions start with the state shown):



a.) The list is operating as a queue, and the program performs: `list.out(); list.in('D');`

b.) The list is operating as a stack, and the program performs: `list.pop(); list.push('D');`