FINAL EXAMINATION
DECEMBER 2013
CSC 212 ♦ 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.

Vocabulary (12 points)

Give a brief yet precise definition of each of the following terms.

a.) Instance

b.) Scope (of a variable or other symbol)

c.) Declaration

d.) Allocation

e.) Complexity (of a program)

f.) Abstract class

g.) Reference type

h.) Recursion

i.) Interface

j.) Override (as a method)

k.) Member (of a class)

l.) Overloading (regarding a method)

Graphs (8 points)

Draw the directed graphs represented by the following:

a.) Adjacency graph. Label the nodes A through H. →

0	0	1	0	0	0	0	1
0	1	0	0	0	0	0	0
0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0
0	0	0	0	1	0	0	1
0	0	0	0	0	0	0	0
0	1	1	0	0	1	0	0

b.) Edge matrix. Label the nodes P through T. →

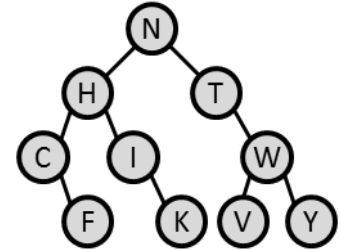| -1	0	1	0	0
-1	0	0	0	1
1	-1	0	0	0
0	-1	1	0	0
0	-1	0	0	1
0	1	0	-1	0
0	0	1	-1	0
0	0	0	1	-1

Trees (12 points)

Consider the tree at right in answering the following questions.

a.) Predict the output of the following algorithm performed on the root:

    procedure: tour(node)
        output data
        if right not null then tour(right) endif
        output data
        if left not null then tour(left) endif
        output data

b.) Suppose that the tree is a binary search tree using alphabetic order on the nodes. Where in the tree would the value M have to be inserted?

c.) Again assuming that the tree is a BST, draw what the tree would look like if H is deleted, with a copy-left protocol to fill in holes.

Recursion (12 points)

Consider the following recursive procedure for determining the winner of a sporting tournament in answering the questions that follow. Note that $\lfloor x \rfloor$ indicates a *floor* operation (round down)

    procedure: champion($team_0, team_1, ..., team_{n-1}$)
        if n == 1 then
            return $team_0$
        else
            p = $\lfloor n/2 \rfloor$
            return winnerOfGame(champion($team_0, ..., team_{p-1}$),champion($team_p, ..., team_{n-1}$))
        endif

a.) In a tournament with 7 teams, how many times will the *winnerOfGame* method be called?

b.) In a tournament with 12 teams, some teams will need to win three games in order to be declared champion, while others will need to win four. Which teams (e.g., *team_0*, etc.) will only have to play three games?

c.) Five teams are playing a tounament: the **Holyhead Harpies**, **Wimbourne Wasps**, **Chudley Cannons**, **Tutshill Tornados**, and **Puddlemere United** (numbered in that order). Assuming that all games are won by the team whose name is first in alphabetical order, list the opponents in every game of the tournament, in the order the games would be played.

Stacks & Queues (8 points)

For each of the following, identify whether their behavior is stack-like, queue-like, or neither.

a.) A bag of candy

b.) Students by class year at Smith (e.g., class of '14, class of '15, etc.)

c.) Nodes already seen but waiting to be processed during breadth-first traversal

d.) Method calls in Java

e.) Coats of paint on a house

f.) Layers of skin

g.) A heap

h.) Cars in a parking lot

Linked List (24 points)

Consider the diagram at right, showing a doubly linked list in memory. For each question below, assume that the starting configuration is as shown.
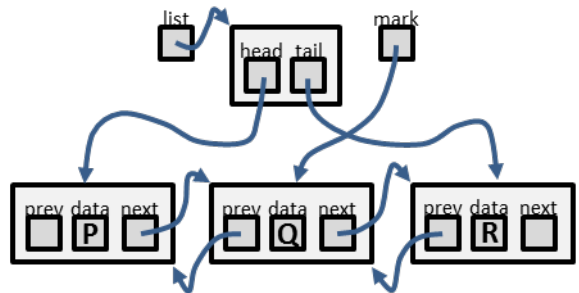
a.) Write a line of code to print out the data of the first element of the list, using mark as the starting point.

b.) Write one or more lines of code to remove the first element from the list, leaving the data structure in a consistent form.

c.) Draw the memory structure that would result from execution of the following code:
```
mark.next.prev = mark.prev;
mark.next.nxt = mark;
mark.prev = mark.next;
mark.next = null;
list.tail = mark;
```

d.) Is the resulting structure from part c a consistent data structure? Describe what has happened to the list in words.

e.) Draw a diagram showing a new variable list2 that is a reference copy or alias of list.

f.) Draw a diagram showing a new variable list3 that is a shallow copy of list.

Programming Practice (8 points)

Consider the program below. Identify any Java errors or instances where the style does not meet the standards expected in thie course.

```java
/** Class for the exam
 *  @author N. Howe
 *  @version December 2013
 */
public class Name {
    /** Parts of the name */
    private String firstName, lastName;

    /** Constructor for first name */
    public Name(String firstName) {
        this.firstName = firstName;
    }

    /** Constructor for last name */
    public Name(String lastName) {
        lastName = lastName;
    }

    /** Accessor for first name */
    public String getFirstName() {
        return this.firstName;
    }

    /** Accessor for last name */
    public String getLastName() {
        return lastName;
    }

    /** Manipulator for first name */
    public void setFirstName(String name) {
        this.firstName = name;
    }

    /** Manipulator for last name */
    public void setLastName(String name) {
        lastName = name;
    }

    /** Convert to full name
     *  @override
     */
    public String toString() {
        return firstName+" "+lastName;
    }
}
```

Program Analysis (8 points)

Consider the method below, which implements a proposed sorting algorithm.

```
public static void cocktail_sort(int[] arr) {
    for (int i = 1; i < arr.length/2; i++) {
        for (int j = i-1; j < arr.length-i; j++) {
            if (arr[j] > arr[j+1]) {
                int tmp = arr[j];
                arr[j] = arr[j+1];
                arr[j+1] = tmp;
            }
        }
        for (int j = arr.length-i-1; j > i-1; j--) {
            if (arr[j] < arr[j-1]) {
                int tmp = arr[j];
                arr[j] = arr[j-1];
                arr[j-1] = tmp;
            }
        }
    }
}
```

a.) Determine the exact number of comparisons this algorithm would make, in terms of the number of elements in the array *n*.

b.) How does this performance relate to other sorting algorithms we have studied?

Programming Principles (8 points)

Write a few paragraphs describing what you know about the `static` modifier in Java. You should consider its use with both fields and methods, in terms of motivation and effects. What are the restrictions imposed, and the benefits gained, from its use? Try to be as specific as possible, preferably with examples, while considering the topic in its entirety.