

NAME: _____

NETID: _____

CS2110 Fall 2013 Final

Write your name and Cornell netid. There are 5 questions on 14 numbered pages. Check now that you have all the pages. Write your answers in the boxes provided. Use the back of the pages for workspace. Ambiguous answers will be considered incorrect. The exam is closed book and closed notes. Do not begin until instructed. You have 90 minutes. Good luck! And have a nice winter break!

Note: Please be careful to use correct Java syntax. Many people lost points on prelim2 for omitted (), ;, {}, etc. We deduct when we notice Java syntax errors!

	1	2	3	4	5	Total
Score	/20	/20	/20	/20	/20	
Grader						

1. (20 points) You've taken a job with Facebook and are working on a new app called "AllAboutMe". The purpose of this application is to show you photos of yourself that were posted by other people. You'll solve the problem in three steps. Step (b) uses the method from step (a), and step (c) uses (b). Don't forget to answer part (d) also.

Here are two classes used by Facebook to represent data:

<pre>public class User { public String name; public Set<User> friends; public Set<Photo> photos; }</pre>	<pre>public class Photo { public JPG image; public Set<String> tags; }</pre>
--	--

Each User has a name, a set of friends, and a set of photos. Each Photo has an image (a jpeg object) and a set of tags. These tags could include user names, which is the case that interests us.

(a) [6 points] Write the body of the following method. Assume that who, u, and ps are non-null.

```
/** Search the photos in User u to see if any are tagged with name who. If any
 * photo with that tag is not already in set ps, add that photo to ps.*/
public static void collectPhotosOfName(String who, User u, Set<Photo> ps) {

}
}
```


(c) [7 points] Last, write the body of the method below, which solves the overall problem: Find all pictures tagged with a user name that the user's friends, their friends, etc., have. Use the method of part (b). *Hint: User u's own photos shouldn't be included!*

```
/** Return the set of all photos that friends of u, their friends, their
 * friends, etc. have that are tagged with u's name. */
public static Set<Photo> allAboutMe(User u) {
```

```
}
```

(d) [2 points] Explain briefly why the method you wrote in (c) won't have infinite recursion.

2. (20 points) True or false?

a	T	F	It is important to provide synchronization for any data that might be updated by more than one thread or that might be updated by some thread and read by other threads.
b	T	F	Within a minimum spanning tree for an undirected graph, the path between any pair of nodes is also the shortest path between those two nodes in the underlying graph.
c	T	F	If an undirected graph is disconnected, we can form one minimum spanning tree per component, but we can't form a single minimum spanning tree for the entire graph.
d	T	F	If an undirected connected graph has v nodes, the minimum spanning tree will have $v-1$ edges and no cycles.
e	T	F	If a field in some class is marked with static , it can be initialized but can't later be changed.
f	T	F	In a cloud computing setting, one useful tool is a version of a HashMap spread over the nodes in a data center. Such a hash map is called a Distributed Hash Table, or DHT.
g	T	F	If a computer has C cores and plenty of memory, and you write a correct, deadlock-free program with T properly synchronized threads ($C \geq T$), it should be T times faster than a single-threaded program, no matter what problem it solves.
h	T	F	A min-heap satisfies the invariant that the value in the root node is the minimum among the root value and the values of its child nodes.
i	T	F	During execution of a program using a GUI, if we add a component to a Box object, the layout manager will automatically re-arrange everything on the monitor, without us having to call a method to tell it to re-arrange the GUI.
j	T	F	When we use a try { something } catch (Exception e) { recovery-code }, if an exception occurs within the "something" block, the recovery-code block will run, after which Java re-executes the instruction or line of code that triggered the exception.
k	T	F	Suppose we create an ArrayList<Dog> x, add a single Dog object to x, and then pass x to a method that has a single argument y of type Dog. Java will "auto-unbox" the object we added to x, and when the method runs, y will reference that object.
l	T	F	Suppose that d is an instance of Dog and that the <i>only method</i> defined in class Dog is bark(). If you set String s = d.toString(); method bark will automatically be called and its result will be converted to a string and saved in s.
m	T	F	In Java, if x and y are declared to be of type double[][] and are both square, NxN matrices, then x = x*y saves the matrix product of x times y into x.
n	T	F	If a class includes a recursive method, it is illegal for that method to access static class variables (from the same class), even if they are marked public.
o	T	F	If a class includes a static method that needs to access instance variables from the class, it must qualify those accesses by using an explicit object reference.
p	T	F	If an ADT specifies that some method m should have complexity $O(n \log n)$, and your implementation of that method could have worst-case complexity $O(n^3)$, Java will warn you about this at compile time and it won't be possible to run m until you fix the implementation.
q	T	F	If the first line of a JUnit test is an assert statement and you are running that test, the assertion will be <i>continuously monitored</i> by Java until the test finishes. If it is ever violated (even temporarily) while the unit test is still running, the unit test will fail.
r	T	F	The following is not an infinite loop: <code>int x= 1; while (x > 0) {System.out.print("+"); x= x+1;}</code>
s	T	F	One representation of a hash table represents data in a 1-dimensional array and resolves collisions using "linear probing". True or false: For this implementation, a good way to delete an item is to set the array element to which it was mapped to null.
t	T	F	If thread T1 is a producer and thread T2 is a consumer and they interact through a shared Bounded Buffer object b, neither will ever need to wait when calling the methods in b.

3. (20 points) Suppose we store N distinct strings (e.g. no two strings have equal values) into a binary search tree (BST) with the following fields:

```
public class BSTNode {
    public String value;
    public BSTNode left;
    public BSTNode right;
}
```

(a) [5 points] Write the body of the method shown below

```
/** Return the number of nodes in BST x (return 0 if x is null). */
public static int numberOfNodes(BSTNode x) {

}

}
```

b. [6 points] Write the body of the method below. Your solution should call the method of (a).

```
/** Using numberOfNodes from (a), return the number of BST nodes in the subtree rooted at x
that have values smaller than v, using standard string comparison. */
private static int numberOfSmaller(BSTNode x, string v){

}

}
```

(c) [9 points] For each of the following questions, circle the best answer

In a BST of height h , the minimum number of nodes would be ...

- i. $h+1$
- ii. 2^h
- iii. h^2

In a BST of height h , the maximum number of nodes would be ...

- i. $2(h+1)$
- ii. $2^{h+1}-1$
- iii. h^2

In a BST of height h , a worst-case search for a value that is present requires

- i. $h+1$ comparisons
- ii. $\log(h)$ comparisons
- iii. A constant number of comparisons, but the constant depends on the order in which the elements were inserted

4. (20 points) Suppose we are given a directed graph representing the highway system in some country. Vertices are locations, edges represent road segments between them, and each edge has a weight corresponding to the average travel time for that road segment.

Consider the following graph algorithms from class: BFS, DFS, Dijkstra's, and MST (Prim's method).

(a) [5 points] Which method would be the most suitable for recommending the fastest way to get from some starting location to a specific destination (e.g. from Ithaca to Miami South Beach)? Why?

(b) [5 points] What is the best way to find an alternative route if you are following the fastest route (as computed using the method you recommended in (a)) but discover that the recommended route is blocked because of road construction? E.g. you are driving from Ithaca to Miami South Beach, but the recommended route that runs from Philadelphia to Washington DC is incredibly slow. Don't write code, but do explain exactly how this problem can be efficiently solved (i.e. what algorithm to use, how to modify the graph, if you think it needs to be modified, etc). **Your solution should stick to the original route except on this section, avoid all the road segments in impassable section, and still seek to minimize the driving time.**

(c) [5 points] Suppose that two friends are coming with you. One lives in Syracuse (northeast of Ithaca), and one lives in New York City (southeast of Ithaca and Syracuse). Can an unmodified, standard implementation of Dijkstra's algorithm be used to compute a shortest path that starts in Ithaca but that passes through Syracuse and New York, and then ends in Miami? Explain.

(d) [5 points] Suppose we label some of the location nodes with restaurants. Suppose you are given a route from Ithaca to Miami, a departure time, and a time when you would like to stop for food.

How could you write a program that would recommend 10 options for stopping, ordered by their distance from where you will be at the desired stopping time? Don't give code, but make sure to tell us precisely how a route will be represented and precisely what your solution would do. Assume there are thousands of candidate restaurants on the route you'll be following.

5. (20 points) We are building a multi-threaded program that maps the locations of animals in a new kind of “open” zoo where many of animals co-exist in a huge shared exhibit space, within which they can wander about as they wish.

Each animal wears a small tracking device. For each animal, the program creates a thread, which loops: every 30 seconds it reads the animal’s location and updates the map (in variable *theMap*) by deleting a pin that represents the previous location and placing a new pin representing the new location. An additional thread reads the map once every few seconds and redisplay it on monitors throughout the park. Visitors use the map to find the herd of zebras, the monkeys, etc.



(a) [5 points] You have been hired to use your cs2110 skills to help fix a bug: animals keep flickering on the displayed maps. For example, there are moments when all the zebras simply vanish. Looking at the display thread and the location update thread, you see code like the following:

<pre>/** Called every 5 seconds to redraw the map */ public static void redisplayMap() { synchronized (theMap) { repaintMapGUI(theMap); refreshGUI(); } }</pre>	<pre>/** Called every 30 seconds to update location pin */ public static void updateLoc(Animal what) { synchronized (what) { oldLocation= what.where; what.where= what.RFID.readLocation(); } synchronized (theMap) erasePin(oldLocation); synchronized(theMap) paintPin(what.animalSpecies, what.where); }</pre>
---	---

Briefly explain why the animals sometimes vanish.

b. [5 points] Rewrite the code for updateLoc to correct the problem. Keep it simple!

```
/** Called every 30 seconds to update location pin */  
public static void updateLoc(Animal what) {
```

```
}
```

c. [10 points] A year has passed and you've graduated from Cornell with a PhD in Romance Languages, but there aren't any jobs for people with PhDs in English, so you are back at the zoo. The zoo's software director (who also has a PhD, in Medieval History) explains that there were too many threads, so he modified the program so that each location thread could update locations for hundreds of animals. Now he only has four location-update threads.

He also added new functionality: now there are threads that track the animal feeding schedules (by keeping the carnivores well fed, they can share the exhibit with the grass-eating animals), threads that track medication schedules, etc. But the program suffers from deadlocks! Reviewing the code you see a great number of synchronized code blocks, often nested one inside another. The director explains that while the program often use synchronized it never uses wait/notify, Locks, or Semaphores.

i. [2 points] Define the term *deadlock*. *Hint: You do not need to list the three conditions for deadlock. We are only asking for a definition.*

ii. [2 points] If the Map refreshing thread became deadlocked, what would happen —what would visitors see on the monitors around the zoo?

- iii. [3 points] The director has heard of deadlocks and knows his program has deadlocks but has difficulty visualizing the basic idea. When he learned Java, his instructor taught him that a deadlock was impossible in a program with no wait statements. Give a very simple example (give the actual Java code) of how a deadlock could arise in Java, using two threads T1 and T2, two objects x and y, and synchronization statements (but no wait or notify statements).

What thread T1 does	What thread T2 does

- iv. [3 points] Give an example of a rule the zoo program could follow that would guarantee that no deadlocks can arise. Assume that the code still will need to use synchronized keywords and that the synchronization blocks still will need to nest but that no use of wait/notify occurs and that there are no Lock or Semaphore objects.