Fall ’15 CIS 212 Final Review

You may bring two pages of notes, front and back.

Questions will be in short-answer format with partial credit for partial answers.

You will be asked to both read and write Java code, but only write code using primitive types (including Strings), Math.random(), System.out.println(), and any specific objects/methods below.

Topics:

- All midterm topics
- Searching and sorting: linear/binary search, selection/merge sort
- Complexity: $O(1)$, $O(\lg n)$, $O(n)$, $O(n \lg n)$, $O(n^2)$
- Array lists, linked lists (singly and doubly linked), hash maps: complexity of add, remove, access operations
- Stacks, queues, and sets: basic properties and typical operations
- Generic classes: purpose, defining a generic class, instantiating a generic object
- Runnable interface: run() method, executing with an ExecutorService
- Thread synchronization: LinkedBlockingQueue, synchronized keyword
- Networks: Socket, ServerSocket, ObjectInputStream, ObjectOutputStream, blocking calls

Sample questions:

1. [10] Write a Java implementation for a method that returns the smallest integer in an array of unsorted integers. What is the Big-O complexity of this algorithm?

2. [5] What is the Big-O complexity of a singly-linked list access operation (i.e., get an item at an arbitrary index)? Why?

3. [5] Briefly describe how a set differs from a list in terms of order and uniqueness of elements:

4. [10] Consider the following Java code:

```java
public synchronized void print(int start, int end) {
    for (int i = start; i < end; ++i) {
        System.out.println(i);
    }
}

public void printNumbersThreaded() {
    ExecutorService executor = Executors.newCachedThreadPool();
    executor.execute(new Runnable() { public void run() { print(0, 10); }});
    executor.execute(new Runnable() { public void run() { print(10, 20); }});
}
```

What (if anything) can we guarantee about the order in which the numbers will be printed?

5. [5] Briefly describe the purpose of the ServerSocket accept() method in terms of what it returns and when it returns: