CS/ENGRD2110: Prelim 1

10th of March, 2011

NAME : _____

NETID: _____

- The exam is **closed book and closed notes**. Do not begin until instructed. You have **90 minutes**. Good luck!

- Start by writing your name and Cornell netid on top! There are **10 numbered pages**. Check now that you have all the pages.

- Web, email, etc. may not be used. Calculator with programming capabilities are not permitted. This exam is **individual work**.

- We have **scrap paper** available, so you if you are the kind of programmer who does a lot of crossing out and rewriting, you might want to write code on scrap paper first and then copy it to the exam, just so that we can make sense of what you handed in!

- Write your answers in the space provided. Ambiguous answers will be considered incorrect. You should be able to **fit your answers easily into the space we provided**. Answers that are not concise might not receive full points.

- In some places, we have abbreviated or condensed code to reduce the number of pages that must be printed for the exam. In others, code has been obfuscated to make the problem more difficult. This does not mean that its good style.

POINTS:

Classes, Interfaces, Types, and Stuff	_____	/ 19
Recursion	_____	/ 14
Trees	_____	/ 22
Asymptotic Complexity	_____	/ 18
Sorting	_____	/ 13
Lists	_____	/ 14
	==========	
Total	_____	/100

1 Classes, Interfaces, and Types

1. Answer the following questions with either true or false. No explanation necessary.

   7 pts.

   - Both interfaces and classes define the type hierarchy in Java.
   - Every type in Java has exactly one or zero supertypes.
   - Every type in Java has exactly one or zero subtypes.
   - A cast changes the dynamic type of an object.
   - Upcasts can produce runtime errors.
   - Abstract classes that contain no implementations provide exactly the same functionality as interfaces.
   - The static type of an argument to an overloaded method determines which of the methods is selected.

2. Given the interface definitions from above, what are the methods that need to be present in the following class `CA`?

   3 pts.

   ```
   interface IA {
           public IA mA(int a);
   }

   interface IB extends IA {
           public int mB(int a);
           public Object mBB(int a);
   }

   interface IC extends IA {
           public IA mC(int a);
   }

   class CA implements IB {
   ...
   }
   ```

3. Given the interface definitions from above, does the following class definition contain any errors? If yes, what are the errors?

   2 pts.

   ```
   class CB implements IA {
           public IC mA(int a) { return this; }
   }
   ```

4. What output does the following program produce?

```java
class CC {
        int x=1;
        public float mD(int a) { return a; }
}

class CD extends CC {
        Integer x=2;
        public float mD(int a) { return x*a;}
}

public class TypeMania {
        public static void main(String[] args) {
                CD x = new CD();
                System.out.println(x.mD(1));
                CC y = (CC)x;
                System.out.println(y.mD(1));
                y.x=3;
                System.out.println(y.mD(1));
        }
}
```

5. The following program does not compile correctly. Please identify the problem in one sentence.

```java
public class Test {
        public static void main(String[] args) {
                System.out.println(max(3,5));
        }

        public int max(int a, int b) {
                if(a > b) {
                        return(a);
                }
                else {
                        return(b);
                }
        }
}
```

2 Recursion

1. What is the output of rec(14,4)?

```java
public static void rec(int a, int b) {
        if (a == 0 && b == 0) return;
        if (a >= b) {
                System.out.print (a + " ");
                rec((a/2)-1, b);
                System.out.print (b + " ");
        }
        else {
                System.out.print (a + " ");
                rec(a, b-2);
                System.out.print (b + " ");
        }
        return;
}
```

2. Write a method with the signature `public int max(int[] array)` that returns the maximum value of all the elements in the array. You can assume that the array contains at least one element. Do this without using any loops but use recursion instead.

   Hint: Feel free to create a helper methods.

3 Trees

1. Give the preorder, inorder, and postorder traversal of the following tree.

   9 pts.

   Preorder:

   Inorder:

   Postorder:
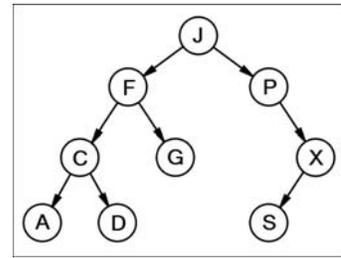
2. Show the binary search tree (BST) after inserting the following sequence of elements. The lexicographical ordering should be used to construct the tree.

   7 pts.

   M, G, A, Q, N, B, Z

3. Derive an efficient method for deleting a given element (denoted by X) from a BST, so that the resulting tree after deletion is again a BST. There are three cases to be considered depending on where in the tree the element is located. Explain in English (or through a diagram showing how the tree is modified) what needs to be done in each case. Do NOT write Java code.

6 pts.

4   Asymptotic Complexity

1. Answer the following questions with either true or false. No explanation necessary.

   6 pts.

   - Binary search in a sorted array with n elements has average-case time complexity $O(log(n))$.
   - Binary search in a sorted array with n elements has worst-case time complexity $O(log(n))$.
   - Finding an element in a BST with n elements has worst-case time complexity $O(log(n))$.
   - There might be an algorithm for sorting an array of integers with n elements in time $O(n)$.
   - Inserting n elements into an unsorted linked list can be done in worst-case time $O(n)$.
   - Checking whether an element is contained in a general tree can always be done in average-case time $O(h)$, where h is the height of the tree.

2. Give the mathematical definition of "$f(n)$ is $O(2^n)$".

   5 pts.

3. Prove that $(3 * n^2 + 15 * n * log(n) - n)$ is $O(n^2)$.

   4 pts.

4. Prove that 3^n is NOT $O(2^n)$.

   3 pts.

5 Sorting

1. The following algorithm (in abbreviated pseudo code) sorts an array of n integers. In Big-O notation, what are the numbers of pair-wise comparisons that this algorithm makes in the best case, the expected case, and in the worst case? Give the tightest Big-O statement you can make.

6 pts.

```
void lameSort(int[] arr) {
  for (int i = 1; i < arr.length; ++i) {
    for(int j = 0; j < arr.length - i; ++j) {
      if (arr[j] > arr[j+1]) {
        'swap arr[j] and arr[j+1])'
      }
    }
  }
}
```

Best-case:

Expected-case:

Worst-case:

2. Explain (in English) how you could improve the sorting algorithm from above so that it has best-case time complexity of $O(n)$? What is this best case?

5 pts.

3. Yes or No: Is the sorting algorithm from above stable? No explanation necessary.

2 pts.

6 Lists

1. In the following code, what does the method "mystery" do? Also explain briefly when such a method would be useful?

8 pts.

```java
class ListNode {
        ListNode next;
        String value;
}

public class List {
        ListNode head;

        boolean mystery(String s){
                if(head == null) return false;
                if(head.value.equals(s)) return true;
                ListNode p = head;
                while(p.next != null){
                        if(p.next.value.equals(s)){
                                ListNode q = p.next.next;
                                p.next.next = head;
                                head = p.next;
                                p.next = q;
                                return true;
                        }
                        else
                                p = p.next;
                }
                return false;
        }

        //other list methods not shown
}
```

2. Add a method `deleteMax()` to the class `List` from above that finds the maximal element in the list and deletes all its occurences from the list.

<div align="right">6 pts.</div>

Reminder: A string `s` is greater than a string `t` if `s.compareTo(String t) > 0`.