Prelim 1 - Solutions

CS2110 Spring 2014

March 11, 2014

	1	2	3	4	5	Total
Question	TrueFalse	Multiple	String	LinkedList	Trees	
Max	20	30	20	5	25	100
Score						
Grader						

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 5 questions on 8 numbered pages, front and back. Check now that you have all the pages. When you hand in your exam, make sure your booklet is still stapled together. If not, please use our stapler to reattach all your pages!

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, so if something seems to need more space, you might want to skip that question, then come back to your answer and see if you really have it right.

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.

1. True / False (20 points)

a)	T	F	In a Stack, the first element to be put in is the first one to come out.
b)	T	F	After the statement, `char c= 2014;`, the type of `c` becomes `int`.
c)	T	F	It is not possible to write a search method in a tree using iteration.
d)	T	F	In Java, we cannot change the value of a String. In other words, Strings are immutable.
e)	T	F	In Java, there are two ways to get the first character of String `s`, which are `s.charAt(0)` and `s[0]`.
f)	T	F	If class `C` is defined as `public class C implements C1, C2`, and both `C1` and `C2` have method `run()`, if we invoke `x.run()` on an object `x` of type `C`, the version of `run` defined in `C1` is executed, due to inside out rule.
g)	T	F	If non-abstract class `A` extends abstract class `B`, the following expression is valid: `B b= new A();`
h)	T	F	All fields of an interface are `final`.
i)	T	F	If class `A` has two methods with signatures `m(int)` and `m(String)`, we say that `m` is overridden and put annotation `@Override` before its second definition.
j)	T	F	In a non-static method we can call a static method.
k)	T	F	If class `A` extends abstract class `B`, but doesn't implement a particular non-abstract method defined in `B`, the code won't compile, because we have to override that method in `A`.
l)	T	F	Within a constructor, we can call another constructor of the same class using `this`.
m)	T	F	A method with return type `void` has to end with the `return;` statement.
n)	T	F	Within the same package, we can access any `private` member of all classes defined.
o)	T	F	In a JUnit testing class, we have access to `private` members of the classes we are testing.
p)	T	F	In an `ArrayList`, we have random access to elements using methods `get` and `set`.
q)	T	F	If we have two `catch` blocks corresponding to the same `try` block, we can catch a particular exception and all of its subclasses with the first one and any other exception with the second one.
r)	T	F	The expression `new Integer[3][]` creates a one-dimensional array object with 3 slots, all of which are set to `null`.
s)	T	F	Consider the class hierarchy for a class C that extends another class and may implement interfaces. This class hiearchy is a tree with C at the root and class Object is one of its leaves.
t)	T	F	If class `A` extends class `B`, and `A x= new B();`, then the static type of `x` is `A`.

2. Multiple Choice (30 points)

In each part, circle ONE of the possible answers A, B, C and D.

(a) You need to store and manipulate the information of n students. Each student has a unique ID in the range $1..n$. You will need to frequently access some random students by their IDs. Which of the following data structures would you use?

A. Tree B. Array C. Linked List D. Set

(b) Given a binary tree of depth d. What is the maximum number of nodes it can have? What is the minimum?

A. $2^{d+1}-1$, d+1 B. d+1, $\log(d+1)$ C. 2^{d+1}, d D. $2^{d+1}-1$, 2^d

(c) Given the definition of function shown below, what is the value of `fun(15, 9)` ?

```
public static int fun(int x, int y) {
   if (x == y)
     return x;
   else
     return x > y ? fun(x-y, y) : fun(x, y-x);
}
```

A. 1 B. 3 C. 5 D. 7

(d) Which of the following is NOT true:

A. A local variable is a variable declared in the body of a method.
B. A local variable declared at the beginning of a method maintains its value from one call of the method to the next.
C. The expressions within the () of method calls are called arguments.
D. An instance method in one class can refer to a public static variable in any public class.

(e) Keyword **static**

A. Tells Java that something will never change.
B. Says that there is one copy of a field or method and that it is associated with the class.
C. Forces Java to interpret the associated field or method as if it were defined in the parent class.
D. Forbids a variable to be initialized in a constructor.

(f) Which of the following is not a primitive type in Java?

A. `int` B. `bool` C. `byte` D. `char`

(g) Given the grammar below, which of the following NetID's is legal? {} means zero or more occurences, {}1 means 0 or 1 occurence.

```
<NetID>    ::= <Initials> <Number>
<Initials> ::= <Letter> { <Letter> }1
<Letter>   ::= a | b | c
<Number>   ::= <Nonzero> { <Digit> }
<Digit>    ::= 0 | <Nonzero>
<Nonzero>  ::= 1 | 2 | 3 | 4 | 5 | 6| 7 | 8 | 9
```

A. `ad1240` B. `cb1204` C. `ab0124` D. `cba012`

(h) Given the class definitions below, `A a= new B(); System.out.println(a.x * a.m());` prints:

A. 2 B. 4 C. 6 D. 12

```
public class A {
  public static final int x = 1;
  public int m() { return 2; }
}

public class B extends A {
  public static final int x = 3;
  public int m() { return 4; }
}
```

(i) In writing a class `Number`, you have a method `toRoman` that should return the Roman equivalent of its input as a String. In the JUnit testing class, which of the following is the best way of expressing your assertion that 2014 corresponds to "MMXIV"?

A. `assertEquals("MMXIV", Number.toRoman(2014));`
B. `assertEquals(Number.toRoman(2014), "MMXIV" );`
C. `assertEquals(Number.toRoman("2014"), MMXIV);`
D. `assertTrue(Number.toRoman(2014) == new String("MMXIV"));`

(j) `1 + 2 + "xyz"` evaluates to

A. `"3xyz"` B. `"12xyz"` C. `"xyzxyzxyz"` D. `6`

3. String and ArrayList (20 points)

Pig Latin is a system for altering words according to a simple set of rules. Thomas Jefferson, principal author of the US Declaration of Independence and third president of the US is supposed to have used Pig Latin in letters to his friends. This question concerns a simplified version of Pig Latin with two rules:

1. If the word starts with a group of consonants, move them to the end of the word and add "ay". Examples: "duck" becomes "uckday". "scram" becomes "amscray".

2. If the word starts with a vowel, move it to the end and add "way". Examples: "eight" becomes "ighteway". "a" becomes "away"

Complete the two functions whose headers are given below. You can make use of the third function, which you do not have to write: We use the name "word" for a string that consists of 1 or more lowercase letters (no digits, punctuation, blanks, etc.). Assume parameters are non-null. Do not use assert statements.

```
/** s is a sequence of words with each pair of words separated by one or more blanks.
*  Return a list of the Pig-Latin translations of the words, with no duplicates */
public static ArrayList<String> m(String s) {
        ArrayList<String> list= new ArrayList();
        s= s.trim();
        // inv: s contains unprocessed words, with no surrounding blanks.
        //      list contains the piglatin of processed words, with no duplicates
        while (s.length() > 0) {
            int k= s.indexOf(" ");
            if (k < 0) k= s.length();
            String w= s.substring(0, k);
            s= s.substring(k).trim();

            String pigw= pigLatin(w);
            if (!(list.contains(pigw))) {
                list.add(pigw);
            }
        }

        return list;
}

/** Return the Pig Latin translation of word w. */
public static String pigLatin(String w) {
        int c= nConsonants(w);
        if (c > 0) {
            String init= w.substring(0,c);
            return w.substring(c) + init + "ay";
        }
        // The first character is a vowel
        return w.substring(1) + w.charAt(0) + "way";
}


/** Return the number of consonants at the beginning of w. */
public static int nConsonants(String w){}
```

4. Linked List (5 points)

Consider the following implementation of `LinkedList`.
Tell us in **one sentence** what $foo()$ does. Assume the head of the list is numbered as node number 0, and the remaining elements are numbered as $1, 2, 3$, and so on.

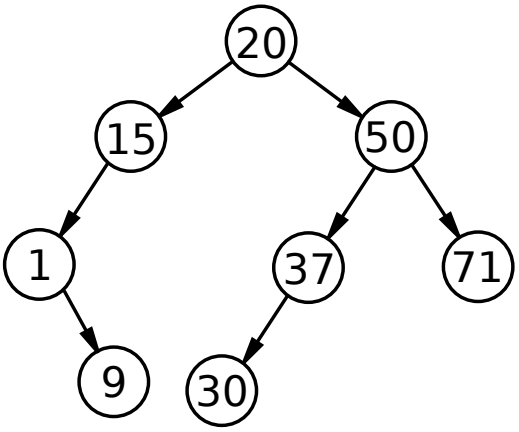
```
public class LinkedList {
    // The first node of the list (null if list is empty)
    public ListNode head;

    private static class ListNode {
        int data;
        ListNode succ; // Next node of the list (null if no more)
    }

    public void foo(int n) {
        if (n < 0) return;
        if (n == 0) {
           if (head==null) return;
           head = head.succ;
           return;
        }
        ListNode p= head;
        for (int i= 0; i < n-1 && p != null; i++)
            p = p.succ;
        if (p == null || p.succ == null)
            return;

        p.succ = p.succ.succ;
    }
}
```

Answer: It deletes element number n if exists, does nothing otherwise.

5. Binary Search Tree (25 points)

(a)[4 points] Draw the Binary Search Tree (BST) obtained by inserting these integers in the following order:

20, 15, 50, 37, 71, 1, 9, 30



For parts (b) and (c), we use the following class TreeNode. Of course, it has various methods, but they are not necessary for this question. You will write two functions, which go in class TreeNode so they can refer directly to the fields. Our sample solution to each is a single return statement, perhaps with conditional expressions. Yours do not have to be.

```
/** An instance is a node of a binary tree. */
public class TreeNode {
    private int value;           // The value associated with this node.
    private TreeNode left;   // The left child of this node --null if none.
    private TreeNode right; // The right child of this node --null if none.

}
```

(b)[8 points] Write the following (recursive) method size.

```
/** Return the number of nodes in the tree whose root is this node. */
public int size() {
    return 1 + (left == null ? 0 : left.size()) +
                   (right == null ? 0 : right.size());
}
```

(c)[13 points] Complete recursive function isBST, below, which checks whether a tree is a BST. That is, each node N in the tree satisfies this: the values in N's left subtree are less than N's value and the values in N's right subtree are greater than N's value. Parameters min and max are there to help in writing the function; without them, it would be hard to write it. Suppose TreeNode variable r contains (a pointer to) the root of a tree. Then the following call will tell whether tree r is a BST.

$$r.isBST(Integer.\text{MIN\_VALUE}, Integer.\text{MAX\_VALUE});$$

```
/** Return true if the following 3 properties hold and false otherwise:
* (1) All values in the tree with this node as root are >= min.
* (2) All values in the tree with this node as root are <= max.
* (3) The tree with this node as its root is a BST.  */
public boolean isBST(int min, int max) {
    return  min <= value  &&  value <= max  &&
        (left == null  ||  left.isBST(min, value-1))  &&
        (right == null  ||  right.isBST(value+1, max));
}
```