CS 201 Exam 3 – Spring 2005

Last Name, First Name (print clearly) ____________________________
Lab Section _________________
UVA Email Address _________________
Pledge:

Solution key

You must pledge the exam. I assume that your work is pledged, even if you do not do so. Please write your email on the scantron page. By signing the pledge or by taking the test you promise not to discuss this test with anyone until Professor Milner gives the class permission to do so. There are several students who will take the exam next week.

This exam is closed note, closed book. You are not to speak with anyone except the Instructor or a teaching assistant for any reason except an emergency during the exam. Turn off all pagers, PDAs, media players, cell phones, etc.

Use the scantron to answer the questions. You must fill in your name, the test (Exam 3), the date and the course CS201.

You must write in and bubble in your ID number or you will be penalized 50 points on this exam. Write and bubble in the test number: 0003.

You must fill in this page and return the entire exam. Do not keep this exam paper.

NOTE: the exam is deceptively short. Please use all the exam time allotted to thoroughly check your answers.

The exam is a total of 100 points. Good Luck!
The height of a tree is defined in the following way: a tree with no children has a height of 0 if it consists of one node with no data and has a height of 1 if it consists of one node with data; otherwise, the height of a tree is defined as one plus the maximum of the two subtrees.

For the following questions we build a binary search tree by inserting integer values into a binary search tree in the following order:

17 33 25 10 13 12 11

1. The height of the tree is:
   A) 3
   B) 4
   C) 5
   D) 6
   E) 7

2. Which binary tree traversal is defined as “first visit left subtree, then visit right subtree, then visit this node:”
   A) Postorder
   B) Inorder
   C) Preorder
   D) Level by level
   E) None of the above

3. Which traversal will result in the following output?
   17 10 13 12 11 33 25
   A) Postorder
   B) Inorder
   C) Preorder
   D) Level by level
   E) None of the above

4. Preorder traversal of the tree will result in which of the following output?
   A) 10 11 12 13 17 25 33
   B) 13 11 10 12 25 17 33
   C) 25 33 11 12 13 10 17
   D) 17 10 13 12 11 33 25
   E) None of the above

5. If the desired goal is to create a tree with the smallest height, which insertion order should be used?
   A) 25 13 11 10 12 17 33
   B) 33 17 10 25 13 11 12
   C) 13 11 10 12 17 25 33
   D) 13 11 25 10 12 17 33
   E) 13 11 10 12 17 33 25
6. Assume I have the following methods for the BinarySearchTreeNode class: getLeft(),
getRight() and getData(). Further, assume I have inserted integers into the binary search tree in
the following order:

3 2 5 1 7 4

Which statement in a unit test will pass?

A. assertEquals(new Integer(1), t.getRight().getRight().getData());
B. assertEquals(new Integer(7), t.getRight().getLeft().getData());
C. assertEquals(new Integer(4), t.getRight().getLeft().getData());
D. assertEquals(new Integer(1), t.getLeft().getRight().getData());
E. assertEquals(new Integer(7), t.getLeft().getLeft().getLeft().getData());

Assume the following code for a binary search tree:

```java
public void testList() {
    BinarySearchTreeNode t = new BinarySearchTreeNode(new Integer(3),null,null);
    t.insert(new Integer(2));
    t.insert(new Integer(1));
    t.insert(new Integer(5));
    t.insert(new Integer(4));
    t.insert(new Integer(7));

    LinkedList ll = new LinkedList();
    ll.add(t);
    while(!ll.isEmpty() ){
        BinarySearchTreeNode tn = (BinarySearchTreeNode) ll.remove(0);
        System.out.print(tn.getData().toString());
        if(tn.getLeft() != null) {
            ll.add(tn.getLeft());
        }
        if(tn.getRight() != null) {
            ll.add(tn.getRight());
        }
    }
}
```

7. What is the last digit this routine will print?

A. 4
B. 7
C. 1
D. 5
E. 2
Consider the following method mystery():

```java
public static int mystery(int num) {
    Stack st = new Stack();
    st.push(new Integer(1));
    st.push(new Integer(4));

    while(true) {
        Integer nm = (Integer) st.pop();
        Integer cnt = (Integer) st.pop();
        if(nm.intValue() <= 0) {
            return cnt.intValue();
        } else {
            st.push(new Integer(nm.intValue() * cnt.intValue()));
            st.push(new Integer(nm.intValue() - 1));
        }
    }
}
```

8. What value does mystery(5) return? (full credit for 8-D and half credit for 8-E)
   A. 4
   B. 12
   C. 48
   D. 24 (parameter num is never used – this was not intentionally written to fool you, I just wrote it this way (wrong) and chose to keep it after discovering the error) full marks if you CAREFULLY traced the code and ½ marks for getting the idea but not being quite so careful)
   E. 120

9. What is the value of the following expression:
   `new Integer(3).compareTo(new Integer(4))`
   A. -1
   B. 0
   C. 1
   D. (Integer) 1
   E. not a legal expression
Consider the following method `mystery2()`:

```java
1. public static int mystery2(int a, int b) {
2.     if(a == 0) {
3.         return b;
4.     } else {
5.         return mystery2(a-1, b+1);
6.     }
7. }
```

10. Which line/lines of mystery2 is the base case or terminating condition:
   A. 1
   B. 2 - 3
   C. 4 - 5
   D. 6-7

11. Which line of mystery2 guarantees forward progress:
   A. 1
   B. 2
   C. 3
   D. 4
   E. 5

12. The method mystery2() : (full for 12-E, half for 12-B)
   A. Calculates multiplication over all the ints
   B. Calculates addition over all the ints
   C. Calculates exponentiation over all the ints
   D. Calculates subtraction over all the ints
   E. None of the above (full marks if you realized this code would not work for all the ints, just those >= 0. Half marks if you realized what the code does but were not so careful)
Consider the following method `mystery3`:

```java
public static int mystery3(int num, int continuation) {
    if(num <= 0 ){
        return continuation;
    } else {
        System.out.print(continuation + " "+num + " ");
        return mystery3(num-1, num*continuation);
    }
}
```

13. What is the output of `mystery3(4,1)`

A. 4 1 4 2 12 2 24 1  
B. 4 1 4 3 12 2 24 1  
C. 1 4 4 3 12 2 24 1  
D. 4 1 3 4 2 12 1 24  
E. 1 4 3 4 12 2 24 1