CS 211	Computers and Programming	
	Fall 2003	
	Prelim I	10/16/2003

NAME:_____

CU ID:_____ Net ID:_____

Section instructor_____

You have one and a half hours to do this exam.

All programs in this exam must be written in Java.

The last page of this exam has a few questions about the course. If you wish to preserve anonimity, you can tear off that page and hand it to TA after filling it in.

Problem	Score	Grader
1 (10 pts)		
2 (20 pts)		
3 (10 pts)		
4 (30 pts)		
5 (15 pts)		
6 (15 pts)		
Total (100 pts)		

1. (Types, 10 points)

   The following definitions and declarations are part of a Java program.

   ```
   interface I1{}
   interface I2{}
   class C1 implements I1 {}
   class C2 implements I2 {}
   class C3 extends C1 implements I2 {}
   .....
   C1 obj1 = ...;
   C2 obj2 = ...;
   C3 obj3 = ...;
   .....
   ```

   Say whether each of the following assignments is legal or illegal. Consider each assignment by itself. Justify each answer in one or two sentences.

   (a) obj2 = obj1;
   (b) obj3 = obj1;
   (c) obj3 = obj2;
   (d) I1 b = obj3;
   (e) I2 c = obj1;

2. (Induction, 20 points)

Use induction to prove the following results. Your answers must state clearly (i) the base case or cases, (ii) the inductive hypothesis, (iii) the inductive step, and (iv) the conclusion.

(a) (10 points)
$(1 - \frac{1}{4})(1 - \frac{1}{9})...(1 - \frac{1}{n^2}) = \frac{n+1}{2n}$ (for $n \geq 2$)

(b) (10 points)

Consider the infinite sequence of numbers $a_1, a_2, a_3, ...$ defined as follows:

- $a_1 = 3$
- $a_2 = 5$
- .....
- $a_{n+1} = 3a_n - 2a_{n-1}$

Prove that $a_n = 2^n + 1$.

This page intentionally left blank.

3. (Recursion, 10 points)

Consider the following function definition.

$$
A(i, j) = \begin{cases} 2^j & i = 1 \text{ and } j \geq 1 \\ A(i - 1, 2) & i \geq 2 \text{ and } j = 1 \\ A(i - 1, A(i, j - 1)) & i, j \geq 2 \end{cases}
$$

(a) (7 points) Write a public class method that takes two integers i and j as parameters, and returns the value of $A(i, j)$. You may assume that integers i and j are greater than zero. You may find the Java method `Math.pow` useful: it takes two doubles `a` and `b` as parameters, and returns the value of a^b as a double.

(b) (3 points) To compute $A(2, 2)$, your method must invoke itself recursively many times. Write down the sequence of these recursive calls and returns, showing the parameters that are passed to each call and the values returned from each one.

For example, to compute $A(2, 1)$, the sequence is

- invoke $A(2, 1)$
- invoke $A(1, 2)$
- return 4 from invocation of $A(1, 2)$
- return 4 from invocation of $A(2, 1)$.

This page intentionally left blank.

4. (Lists, 30 points)

Write a public class method to reverse a list *in-place*. That is, given a reference to the first cell of a list constructed using the ListCell class discussed in lecture, your method must (i) modify the list so that its cells are in the reverse order and (ii) return a reference to the last cell of the original list, as shown in the figure below.

Your method must have the following signature.

```
public static ListCell reverse(ListCell f);
```

The ListCell class is reproduced at the end of the exam for your convenience. You may NOT use the List class discussed in lecture. *No credit will be given if you create a new list; that is, if you allocate any new ListCell objects.*
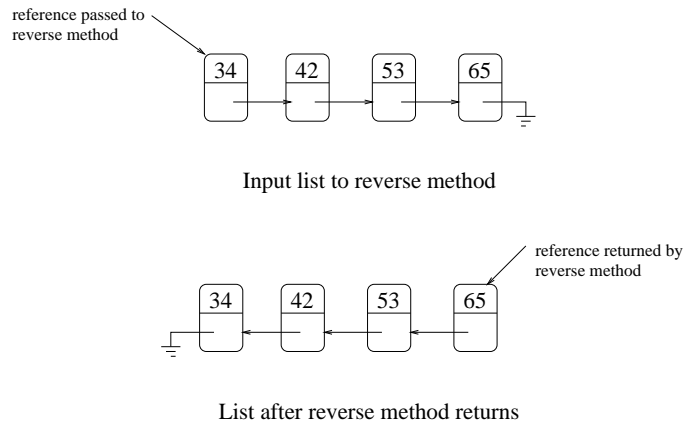

Input list to reverse method

List after reverse method returns

Figure 1: Reversing a list in-place

7

This page intentionally left blank.

5. (Trees, 15 points)

You are given a general tree that is implemented using the GTree-Cell class discussed in lecture. The nodes of the tree contains Integer objects. Write a public class method called sumTree which takes a reference to the root of this tree and returns the sum of all the values stored in this tree.

The GTreeCell class is reproduced at the end of the exam of your convenience. Also, if i is a reference to an Integer object, the method invocation i.intValue() returns the int value that is boxed inside that Integer object.

This page intentionally left blank.

6. (Trees, 15 points)

Given a tree containing a single character in each node, Java Nagila outputs the characters encountered in a preorder traversal of nodes and produces the sequence [a b c d e f g h], while Java de Hutt performs a postorder traversal *of the same tree* and produces the sequence [d c b g f h e a].

   (a) (10 points) Draw a binary tree that might have been the input to their traversals.

   (b) (5 points) Is your answer unique? If not, show one other tree that would produce the same result.

This page intentionally left blank.

```java
class ListCell {

 protected Object datum;
 protected ListCell next;

  public ListCell(Object o, ListCell n){
    datum = o;
    next = n;
  }

  public Object getDatum() {
    return datum;
  }

  public ListCell getNext(){
    return next;
  }

  public void setDatum(Object o) {
    datum = o;
  }

  public void setNext(ListCell l){
    next = l;
  }

 public String toString(){
    String rString = datum.toString();
    if (next == null) return rString;
    else return rString + " " + next.toString();
 }
}
```

```
class GTreeCell{

 protected Object datum;
 protected GTreeCell left;
 protected GTreeCell sibling;

 public GTreeCell(Object o) {
  datum = o;
  left = null;
  sibling = null;
 }

 public Object getDatum() {
  return datum;
 }

 public void setDatum(Object o) {
  datum = o;
 }

 public GTreeCell getLeft() {
  return left;
 }

 public void setLeft(GTreeCell l){
  left = l;
 }

 public GTreeCell getSibling() {
  return sibling;
 }

 public void setSibling(GTreeCell s) {
  sibling = s;
 }

}
```

Course Feedback

You can detach this sheet from the exam and hand it to a TA to maintain confidentiality.

1. How many hours per week are you spending on the homework for this course?

2. Do you attend lecture regularly? What can we do to improve the lectures?

3. Do you attend section regularly? What can we do to improve section?

4. Do you see the consulants for assistance? What can we do to improve consulting?