University of Toronto
CSC148 – Introduction to Computer Science, Summer 2003

Mid Term Test – Section L0101

Duration: 50 minutes
Aids allowed: 1 8.5” x 11” piece of paper with information written on one or both sides.

Make sure that your examination has 6 pages (including this one). Write your answers in the spaces provided. Write legibly. You may use page 6 for rough work (tear it off, if you like). If you require more space to answer a question, write on the back of the previous page, and indicate in the answer space where your answer is.

Personal Information:

Surname:	
Given name(s):	
Student #:	
Circle section of registration:	L0101 or L5101

Please note that if you write the midterm in pencil, you will ***not*** be allowed to submit a remark request.

Do not write anything on this page below this line:

1.	/ 12
2.	/ 12
3.	/ 6
Bonus.	/ 0
Total:	/ 30

Question 1: Quick Concept Questions

[3 sub-questions, 4 marks each]

a. Briefly explain the difference between a `null` String (eg: `String s = null;`) and an empty String (eg: `String s = "";`). Include a description of the differences seen in the Object Space of computer memory.

b. Below is the skeleton of a try/catch block:

```
try {
   // block 1: code that throws an exception
} catch (ExceptionClass1 e) {
   // block 2: code that deals with this type of exception
} catch (ExceptionClass2 e) {
   // block 3: code that deals with this type of exception
}
```

An exception thrown in block 1 might match either ExcpetionClass1 or ExceptionClass2. Under what circumstance would it match both? Which block of code (2 or 3) would run if this were to happen?

c. What special name have we given to trees with a branching factor of 1?

Question 2: Linked List

[12 marks]

Here is a node class:

```
class IntNode {
   int data;
   IntNode next;

   IntNode(int data, IntNode next) {
      this.data = data;
      this.next = next;
   }
}
```

Complete the method body for "getMax" below. ***Be sure to include all internal comments.***

```
/* Returns the maximum value stored in the list rooted at head.
 *
 * Requires: the list contains at least 1 element. */

public static int getMax(IntNode head) {
```

```
} // did you remember to return something, and include comments?
```

Complete the method body for "insertBefore" below (using the IntNode class from the previous page). ***Be sure to include all internal comments.***

```
/* Inserts “n” into the list rooted at “head” before the node
 * containing the value “i” in the list.
 * Returns the head of the list containing n.
 *
 * Requires: there is a node in the list rooted at head
 * containing i. */

public static IntNode insertBefore(IntNode head, int i, int n) {
```

```
} // did you remember to return something, and include comments?
```

Question 3: Recursion

[6 marks]

Here is a recursive method:

```
/* Finds and returns the sum of the series [0,1,…,n]
 * Requires: n>= 0
 */

int sumLess(int n) {

   if (n == 0) {
      return 0;
   } else {
      // A: return sumLess(n);
      // B: return n + sumLess(n);
      // C: return n + sumLess(--n);
      // D: return n + sumLess(n--);
   }
}
```

In order for sumLess to work as specified, which line of code should be uncommented? Circle the correct answer:

A ***B*** ***C*** ***D***

Bonus question

[+3 marks]

Would the line:

```
return (n--) + sumLess(n);
```

work in place of the commented lines above? Circle one of:

YES ***NO***

Explain your answer:

This page intentionally left blank. Tear it out and use it for scratch paper.