Computing & Algorithms II

CPSC-102 **Summer 1999**
Midterm Exam **12 August 1999**

Instructions: This exam is closed-book, closed-notes, and is to be completed individually. Complete all problems; each problem indicates the number of points possible. Show your work; partial credit may be awarded.

1. *5 points each.* Describe each of the following concepts *briefly.*

   (a) Exceptions

   (b) Circular linked lists

   (c) Top-down design

   (d) References

2. *10 points.* By definition, a recursive method is any method which calls itself (either directly or indirectly). Every useful recursive method must satisfy two conditions. Name those two conditions, and explain why each condition is necessary.

3. *10 points each.* Consider the following class definitions (virtually identical to those used in class):

```
class IntNode {
     int datum;        // datum to be stored
     IntNode next;    // next item in list

     public int getDatum()            { return datum; }
     public void setDatum(int which)  { datum = which; }

     public IntNode getNext()         { return next; }
     public IntNode setNext(IntNode which)
                                      { next = which; }
}

class LinkedList {
     IntNode head;   // first item in the list;

     public LinkedList() { head = null; }
}
```

Each of the following methods is a member of the `LinkedList` class. All the methods compile correctly; however, they do not function as described in their headers. What is wrong? How would you fix them?

(a)
```
// This method counts the number of items in the list.

public int length () {

    int count = 1;
    IntNode current = head;
    while (current.getNext() != null) {
        current = current.getNext();
        count++;
    }
    return count;
}
```

(b)
```
// This method places "target" at the front of the list

public void prepend (int target) {

    IntNode newNode = new IntNode();
    newNode.setDatum(target);
    head = newNode;
    newNode.setNext(head);
}
```

(c)
```
// This method searches a sorted list for "target"
// Assumption:  all numbers in the list are sorted
// from small to large

public boolean search (int target) {

    if (head == null) return false;
    IntNode current = head;
    while (current.getDatum() < target)
        current = current.getNext();
    if (current.getDatum() == target)
        return true;
    else return false;
}
```

4. *20 points.* Using the declarations given in the last problem, write a Java method which accepts a linked list as input and returns the average of the elements in that list. (If the list is empty, return `0`.)

5. *20 points.* Write a *recursive* Java method which takes a `String` as input and returns `true` if it is a palindrome. A palindrome is a string which reads the same forward and backward (such as "radar" or "deed").

   Hint 1: A palindrome must start and end with the same letter. What about the letters in-between?

   Hint 2: The following methods may be useful:

   - `str.length()` returns the number of characters in string `str`.
   - `str.substring(begin,end)` returns the substring of `str` starting at `begin` and ending just before `end`.
   - `str.charAt(index)` returns the character at position `index` of string `str`.