University of Toronto
csc148— Introduction to Computer Science Fall 1999

Midterm Test

Duration: 50 minutes

Aids allowed: None

Make sure that your examination booklet has 4 pages (not including this one). Write your answers in the spaces provided. Write legibly.

Surname: ________________ **First name:** ________________

Student #: ________________

Tutor (circle one): Kaytek Przbylski Petter Wiberg Jade Rubick

1.	________	/ 8
2.	________	/ 7
3.	________	/ 6
4.	________	/ 10
Total	________	/ 31

Question 1 [8 marks in total]

Consider the `removeDuplicates` method shown below. It uses the following node class:

```
class Node {
    public Object contents;       public Node link;
    // Constructor.
    public Node (Object o) {
        contents = o;
        link = null;
    }
}

// Removes all but the first occurrence of 'o' in the linked list referred
// to by 'front'.  Uses '.equals' to check for equality.

// Preconditions:

public static void removeDuplicates (Node front, Object o) {

     Node temp = front;
     while (temp != null && !temp.contents.equals(o)) {
          temp = temp.link;
     }
     if (temp != null) {

          // Assertion:  temp is

          Node temp2 = temp;
          while (temp2 != null && temp2.contents.equals(o)) {
               temp2 = temp2.link;
          }

          // Assertion:  temp2 is

          temp.link = temp2;
     }
}
```

(a) [2 marks]
Under some circumstances, this method does not work as described in the comments. Add an appropriate precondition to the method comments.

(b) [4 marks]
Where marked in the comments, write an appropriate assertion about `temp` or `temp2`, as indicated.

(c) [2 marks]
`temp` and `temp2` are terrible variable names. Rename them.

New name for `temp`:
New name for `temp2`:

Question 2 [7 marks in total]

Complete the `insert` and `appendList` methods below. (They use the same `Node` class as question 1.) Note that `appendList` should not construct any new nodes; it should simply hook together the existing lists.

```
    // Inserts 'o' at the front of 'list' and returns the new front node.
    // Preconditions: none.
    public static Node insert (Node list, Object o) {








    }

    // Appends 'list2' to the end of the 'list1' and returns a reference to
    // its new front node.
    // Example: If 'list1' contains 1, 2, 3 (in that order)
    //          and 'list2' contains 7, 8, 9, 10 (in that order)
    // returns a reference to a list containing 1, 2, 3, 7, 8, 9, 10.

    // Preconditions: none.
    public static Node appendList (Node list1, Node list2) {








    }
    // Other methods would go here.
}
```

Question 3 [6 marks in total]

Consider the problem of representing a set of clients of a law firm. Below is an outline for code to represent the set.

Rewrite the outline to improve its design. You may change anything or add anything, but do not add any more detail than is already here. (For example, do not add new data or write any method bodies.) Make your changes directly on the code below.

```
class linkedClientSet {

     // First client in a linked list.
     private Client firstClient;


     // Adds client 'c' to this set.
     public void add (Client c) {   // Details omitted.  }


     // Removes and returns a client from the set.
     // Precondition: the set is not empty.
     public Client remove () {   // Details omitted.  }


     public boolean isEmpty () {   // Details omitted.  }
}


class Client {
     public String name;      public String address;
     // Other client information would go here.

     // The next client in a linked list.
     public Client next;
}
```

Question 4 [10 marks in total]

(a) [6 marks]
Fill in the boxes to show the conclusion that is valid at each point in the following proof:

Case 1: P is true
  Assume Q is false
    Assume R is false
      ⋮
      Therefore, some contradiction holds.
    $\boxed{}$
  $\boxed{}$
$\boxed{}$

(b) [2 marks]
The following structure for a proof by induction jumps up by 2 in the induction step. Fill in the box in a way that makes the proof valid.

BASE CASES: Prove $S(4)$ is true and $S(5)$ is true.

Let $k \geq \boxed{}$ be an arbitrary integer.

INDUCTION HYPOTHESIS: Assume $S(k)$ is true.

INDUCTION STEP: Prove $S(k+2)$ is true.

CONCLUSION: $S(n)$ is true for all $k \geq 4$

(c) [2 marks]
The induction proof structure above has two base cases. If we remove the second base case, what value should go in the box so that the proof will be valid? Circle one answer.

0
1
4
5
6
no value will make it valid

END OF TEST