University of Toronto
csc148— Introduction to Computer Science Fall 1999

Midterm Test – Morning Section

Duration: 50 minutes

Aids allowed: None

Make sure that your examination booklet has 7 pages (not including this one). Write your answers in the spaces provided. Write legibly.

Surname: ______________________ **First name:** ______________________

Student #: ______________________

Tutor (circle one): John Watkinson Jennifer Listgarten Ovidiu Calin

0.	________	/ 1
1.	________	/ 5
2.	________	/ 10
3.	________	/ 8
4.	________	/ 4
5.	________	/ 7
Total	________	/ 35

Question 0 [1 marks in total]

Write your name and student number on each page of this test.

Question 1 [5 marks in total]

(a) [3 marks]
Circle the answer that best describes each of the following. The answers are: (1) ADT (2) data structure (3) can be both (4) neither

Binary search	1	2	3	4
Circular queue	1	2	3	4
Unsorted array	1	2	3	4
Doubly-linked list	1	2	3	4
Priority queue	1	2	3	4
List	1	2	3	4

(b) [2 marks]
Say that a class `C` is in package `P` and has a member `m`, with no accessibility level specified. Suppose another class wants to use `m`. Which of the following kinds of classes can? Circle the appropriate answer for each.

a class that is a subclass of `C` and is in package `P`	CAN	CANNOT
a class that is a subclass of `C` and is in different package	CAN	CANNOT
a class that is not a subclass of `C` and is in package `P`	CAN	CANNOT
a class that is not a subclass of `C` and is in different package	CAN	CANNOT

Question 2 [10 marks in total]

A *container* is a class capable of storing various things. For example, a stack or a queue is a container. In this problem we will work with containers implemented using linked lists. You are given the following abstract class:

```
public class ContainerNode {
   public Object stuff;
   public ContainerNode next;

   // constructor
   ContainerNode (Object stuff) {
     this.stuff = stuff;
     next = null;
   }
}

public abstract class Container {
   protected ContainerNode head;

   // Constructor.
   public Container () {
      head = null;
   }

   // insert stuff into this container
   public void insert (Object stuff) {
     ContainerNode s = new ContainerNode(stuff);
     s.next = head;
     head = s;
   }

   // remove an object from the container and return it.
   // Precondition:  the container is not empty.
   public abstract Object remove ();
}
```

Your goal is to use `Container` to implement a queue of `Objects`.

(a) [6 marks]
Complete the class. The only methods that Queue should have are `insert()`, `remove()`, `isEmpty()` and a constructor. Write a body for `insert()`. It should maintain an invariant that `head` references the beginning of the queue and `tail` – the end of the queue. For the other methods, include only their signature. The signature (also called the header) is the single line heading that begins the method's definition. Comments are not required, although may be helpful.

```
public class Queue extends Container  {
     // end of the queue
     private ContainerNode tail;





}
```

(b) [2 marks] Explain how `Queue` relates to `Container`. For each method in class `Queue`, circle one of the following: `Queue` overrides this method, `Queue` completes this method, `Queue` adds this method.

`insert()`	COMPLETE	OVERRIDE	ADD
`remove()`	COMPLETE	OVERRIDE	ADD
`isEmpty()`	COMPLETE	OVERRIDE	ADD
`Queue()`	COMPLETE	OVERRIDE	ADD

(c) [2 marks]
Instead of inheriting from `Container`, could `Queue` just include `Container` as an instance variable? Answer YES or NO, and if NO, explain why not.

Answer:

Reason:

Question 3 [8 marks in total]

(a) [4 marks]
Our goal is to use induction to prove $S(n)$ for all $n \geq 3$. Fill in the blanks so that the following proof structure will be valid, in other words, so that the conclusion is valid assuming that the sub-proofs in the base case(s) and induction step are properly completed.

BASE CASE(S): Prove that ________________________________
Let k be an arbitrary integer ________________________________
INDUCTION HYPOTHESIS: Assume that $S(k)$ is true
INDUCTION STEP: Prove that $S(k+2)$ is true
INDUCTION CONCLUSION: $S(n)$ is true for all $n \geq 3$.

(b) [4 marks]
What will the following prove:

BASE CASE: Prove that $S(5)$ is true
Let $k > 5$ be an arbitrary integer
INDUCTION HYPOTHESIS: Assume that $S(k)$ is true
INDUCTION STEP: Prove that $S(k+1)$ is true
INDUCTION CONCLUSION: ________________________________

Question 4 [4 marks in total]

Prove that $n^2 + 1 < (n+1)^2$ for all $n > 1$. Do not use induction.

Question 5 [7 marks in total]

An overworked TA in 148 is marking assignments. She knows that the following describes a node in a linked list.

```
public class Node {
   public Object data;
   public Node next;

   // constructor
   Node (Object data) {
     this.data = data;
     next = null;
   }
}
```

But now it is 2:00 in the morning and she is trying to understand the following mysterious function written by one of her students:

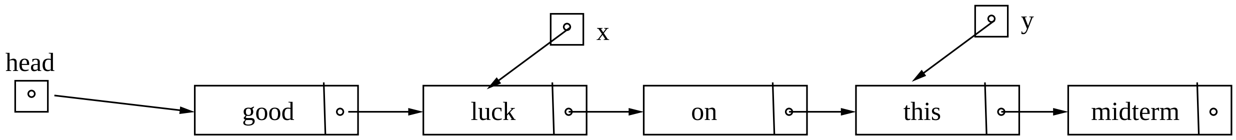
```
public class LinkedList {
   private Node head;

   // constructor and other methods are here

   // pre: a and b are in the list but are neither is the
   //      first element of the list.  a and b are also
   //      not adjacent
   public void Jibberish (Node a, Node b) {
      Node temp = b.next;
      Node preva = head;
      // find predecessors of a and b
      while (preva.next != a)
          preva = preva.next;
      Node prevb = head;
      while (prevb.next != b)
          prevb = prevb.next;

      preva.next = b;
      prevb.next = a;
      b.next = a.next;
      a.next = temp;
   }
}
```

(a) [1 mark]
Is this code legal in Java? Answer YES or NO, and if it is not, explain what is wrong.

Answer:

Reason:

(b) [4 marks]
Let the list be as shown below. Assuming that the code runs, draw this list after executing `Jibberish (x, y)`.



(c) [2 marks]
In one sentence, explain what `Jibberish (a, b)` does.

END OF TEST