Duration: **50 minutes**
Aids Allowed: **NONE**

Student Number: |_|_|_|_|_|_|_|_|_|

Last Name: ________________________________

First Name: ________________________________

Closed Lab:	Mon. 10am	Mon. 10am	Wed. 6pm	Wed. 6pm	Fri. 12pm	Fri. 12pm
(circle one)	BA 2200	BA 2240	BA 2200	BA 2240	BA 2200	BA 2240

*Do **not** turn this page until you have received the signal to start.*
(In the meantime, please fill out the identification section above,
and read the instructions below *carefully*.)

This term test consists of 3 questions on 4 pages (including this one), printed on one side of the paper. *When you receive the signal to start, please make sure that your copy of the test is complete.*

Answer each question directly on the test paper, in the space provided, and use the reverse side of the pages for rough work. If you need more space for one of your solutions, use the reverse side of the page and *indicate clearly the part of your work that should be marked.*

You are strongly encouraged to write the test using a pen, because remarking requests will *not* be granted for tests written using a pencil. (Simply cross off any part of your work that you do not want to be marked.)

If you are unable to answer a question, or you can only give a partial answer, you may get up to 20% of the marks for the question if you state clearly that you do not know how to answer. Note that you will *not* get those marks if your answer contains contradictory statements (such as "I do not know how to answer" followed or preceded by a solution).

General Hint: We were careful to leave ample space on the test paper to answer each question. Also, remember that hints are just hints: you are not required to follow them if you can think of a different solution.

Marking Guide

# 1: ______/16

# 2: ______/17

# 3: ______/17

TOTAL: ______/50

Good Luck!

Question 1. [16 MARKS]

Part (a) [2 MARKS]

Complete the identification section at the top of page 1, then write your student number **legibly** at the bottom of every page of this test except page 1 (where indicated).
(HINT: The questions on this test are in no particular order; start with the easier questions first!)

Part (b) [14 MARKS]

Below, write code for methods `deleteAfter` and `contains` so that they meet their specification.

```
public class SomeNode {
    public Comparable data;    public SomeNode link;
    public SomeNode(Comparable data, SomeNode link)
      { this.data = data;  this.link = link; }
}

public class SomeList {
    private SomeNode head; // the first node in this list; ‘null’ if this list is empty
    // Constructor and other methods go here...

    // Remove the node immediately following ‘n’ from this list.  If n == null, remove
    // the first node in this list.  (Do nothing if there is no node following ‘n’.)
    private void deleteAfter(SomeNode n) {









    } // deleteAfter()

    // Return true if this list contains ‘item’; false otherwise.
    // Requires: item != null  (Use the ‘equals’ method to compare elements.)
    public boolean contains(Object item) {









    } // contains()
} // SomeList
```

Question 2. [17 marks]

Consider the following Java program.

```
public class A {
    public Integer i;
    public A(Integer i)  { this.i = i; }
    public boolean equals(Object o)
      { return this.i == ((A) o).i; }
}
```

```
public class B extends A {
    public Integer i;
    public B(Integer i)  { super(i);  this.i = i; }
    public boolean equals(Object o) { // DRAW HERE
        return this.i.equals(((A) o).i); }
    }
}
```

```
public class Driver {
    public static void main(String[] args) {
        A a = new A(new Integer(73));    B b = new B(new Integer(73));    sioc(a, b);
        System.out.println(a.i+", "+b.i+", "+a.equals(b));         // OUTPUT:
    }
    public static void sioc(A a, A b) {
        b.i = new Integer(35);
        System.out.println(((B) b).i+", "+b.i+", "+b.equals(a));   // OUTPUT:
        b = new A(new Integer(99));
    }
}
```

Part (a) [5 marks]

Write the output of the program, next to each call to `System.out.println` (where indicated by the comments).
(This is *not* a "trick question": the program compiles and runs without error.)

Part (b) [12 marks]

Below, draw a picture of the memory model when execution of the program reaches the line labelled "`DRAW HERE`". Include only the Call Stack and the Object Space in your picture (*i.e.*, do <u>not</u> draw the Static Space).

Question 3. [17 MARKS]

Suppose that you have a class SomeQueue that implements the following Queue interface.

```
public interface Queue {
    int size(); // Return the number of elements I contain.
    void enqueue(Object o); // Add ‘o’ to the back of my elements.
    Object dequeue(); // Remove and return my front element.  Requires: I am not empty.
}
```

Below, write the body of method removeFirst so that it meets its specification.

```
// Remove the first occurrence of ‘o’ from ‘q’ ("first" means "closest to the front").
// Requires: q != null, o != null.  Ensures: ‘q’ contains the same elements as before,
// in the same order, except for the first occurrence of ‘o’ that has been removed.
// Throws: NoSuchElementException if ‘o’ does not appear in ‘q’.
public static void removeFirst(Queue q, Object o) {
    // HINT: Create a new object and add it to the queue as a "sentinel", so that you
    // know when to stop examining elements (this is only part of what you must do).
```

```
} // removeFirst()
```

Total Marks = 50