University of Toronto

Department of Computer Science

CSC 148H1S — Winter 2004
Section L0301

Midterm test

No aids allowed.

Time: 50 minutes

ANSWERS

1	
2	
3	
Total	

1. [10 marks]

Here is a Java program. It compiles and runs without errors.

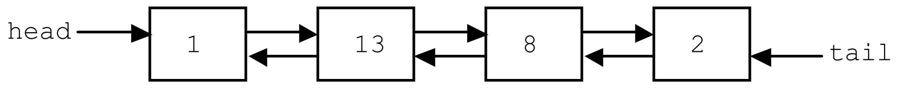
```
class A {
    public void m() { }

    public A() {
        m();
    }
}

class B extends A {
    private int f = 4;

    public void m() {
        System.out.println(f);
    }

    public B() {
        m();
        f = 3;
        m();
    }
}

public class Init {
    public static void main(String[] args) {
        B b = new B();
    }
}
```

Give the output from this program in the space below. (Hint: it is very short.)

```
ANSWER:
0
4
3
```

2. [10 marks]

A doubly-linked list is to be used to store integers. Odd numbers are to go at the beginning and even numbers at the end, so the list might look like this:



Apart from the odd-even separation, the order of the items in the list does not matter. The nodes in the list are instances of this class:

```
    class Node {
        public int data;
        public Node next, prev;
    }
```

The list itself is represented as an instance of the class OddEvenList. Write the instance variables for this class, and the insert method, in the spaces below. Include comments describing both the variables and the method.

```
class OddEvenList {

    // Put your instance variables here.
ANSWER:
    private Node head;
    private Node tail;

    // Put your method comment for insert() here.
ANSWER:
    /** Inserts data into this OddEvenList at the head or tail according
     * to whether it is even or odd.
     * @param data The value to be put into the list.
     */
    // Put insert() itself here.
    void insert (int data) {
ANSWER:
        Node newnode = new Node();
        Newnode.data = data;
        if (head == null) {
            head = newnode;
            tail = newnode;
            newnode.prev = null;
            newnode.next = null;
            return;
        }
        if (data % 2 == 0) { // even
            newnode.prev = tail;
            newnode.next = null;
            tail.next = newnode;
            tail = newnode;
        }
        else { // odd
            newnode.next = head;
            newnode.prev = null;
            head.prev = newnode;
            head = newnode;
        }
}
```

3. [10 Marks]

This interface defines the characteristics of a thing that can be flown:

```
public interface Flyable {
    int getCrewSize();
    int getNumEngines();
}
```

Write two ***classes***, `JumboJet` and `Kite`, that implement the `Flyable` interface and that obey these rules:

1. The constructor of each class is used to set the characteristics of the object. Besides the standard characteristics of crew size and number of engines, a `JumboJet` has a number of passengers and a `Kite` has a tail length. All `Kites` have 0 engines, but there are no other restrictions on either class.
2. One or more methods are provided to retrieve the characteristics of objects belonging to the class.

```
ANSWER:
class JumboJet implements Flyable {
    private int crewSize;
    private int numEngines;
    private int numPassengers;
    public JumboJet(int crew, int eng, int pass) {
        crewSize = crew;
        numEngines = eng;
        numPassengers = pass;
    }
    public int getCrewSize() { return crewSize; }
    public int getNumEngines() { return numEngines; }
    public int getNumPassengers() { return numPassengers; }
}

class Kite implements Flyable {
    private int crewSize;
    private double tailLength;
    public Kite(int crew, double tail) {
        crewSize = crew;
        tailLength = tail;
    }
    public int getCrewSize() { return crewSize; }
    public int getNumEngines() { return 0; }
    public double getTailLength() { return tailLength; }
}
```