

CSC 207H Fall L0101 2011 Java Quiz

Duration — 25 minutes

Aids allowed: none

Student Number:

Last Name:

First Name:

(Please fill out the identification section above and read the instructions below.)

Good Luck!

This midterm consists of 2 questions on 8 pages (including this one). *When you receive the signal to start, please make sure that your copy is complete.* If you use any space for rough work or need to scratch out an answer, circle what you want marked to indicate that it is the answer you are submitting.

1: _____/ 5

2: _____/12

TOTAL: _____/17

Question 1. [5 MARKS]

Suppose you have just checked out a copy of the repository (local copy 1) with the following files and their contents:

planets.txt	seasons.txt	prime_numbers.txt
Mercury	Summer	2
Mars	Fall	3
Jupiter	Spring	5
Saturn		7
Venus		

Part (a) [2 MARKS]

Suppose you now execute the command `svn remove planets.txt`, append a line containing `Winter` to `seasons.txt`, and create an empty file named `empty.txt` in the directory containing local copy 1. What will the output of `svn status` be?

Part (b) [1 MARK]

If someone were to, without running any more commands in the directory containing local copy 1, check out a new copy of the repository (local copy 2), what files would be found and what would their contents be? (You can describe it in English if you like.)

Part (c) [2 MARKS]

If `empty.txt` in the directory containing local copy 1 is modified to contain the line `Hello world`, what command(s) would you need to execute so that the changes are reflected in the directory containing local copy 2? Make sure to indicate where the commands you write are executed (i.e., local copy 1 or local copy 2).

Question 2. [12 MARKS]

Consider the following code for a doubly-linked list:

```
public static class LinkedListNode {
    private LinkedListNode next;
    private LinkedListNode prev;
    private int value;

    /**
     * Constructor for a new LinkedListNode. Appends this node to prev.
     * Preconditions: prev is the last element in a linked list.
     *
     * @param value The value of this node.
     * @param prev The previous node in the linked list.
     */
    public LinkedListNode(int value, LinkedListNode prev) {
        this.prev = prev;
        if (prev != null) {
            prev.next = this;
        }
        this.value = value;
    }

    /**
     * Return the value stored in this node.
     *
     * @returns the value stored in this node.
     */
    public int getValue() {
        return this.value;
    }

    /**
     * Return the next node in the LinkedList.
     *
     * @returns the next node in the LinkedList.
     */
    public LinkedListNode getNext() {
        return this.next;
    }

    public static void main(String argv[]) {
        LinkedListNode aNode = new LinkedListNode(13, null);
        new LinkedListNode(7, aNode);
        aNode.moveForward();
    }
}
```

Part (a) [2 MARKS]

Draw the object(s) created by the following expression. You do not need to draw the full memory model, just the object. You also do not need to include any method signatures in your object.

```
new LinkedListNode(13, null)
```

Part (b) [4 MARKS]

In the next question, you will be writing a method `moveForward` that takes no parameters. Draw the Java memory model diagram just after Java calls the `moveForward()` method on line 3 of method `main`.

Part (c) [3 MARKS]

Write a method named `moveForward()` that takes no arguments in the `LinkedListNode` class that, if there is another `LinkedListNode` after this one, swaps the current node with the next one. Include comments as necessary.

Part (d) [5 MARKS]

Create a method named `sort()` in `LinkedListNode` that, if called from another method, reorders the linked list connected to the `LinkedListNode` into non-decreasing order (i.e., the value of element n is \leq the value of element $n + 1$). You may assume your `moveForward()` method from the previous question works. We suggest using gnome sort or modifying bubble sort. Include comments as necessary.

This page is for rough work and for answers that didn't fit in the space provided.