

Mid Term Test – Section L0101

Duration: 50 minutes
Aids allowed: none

Make sure that your examination booklet has 6 pages (including this one). Write your answers in the spaces provided. Write legibly. You may use page 6 for rough work (tear it off, if you like). If you require more space to answer a question, write on the back of the previous page, and indicate in the answer space where you answer is.

Surname: _____

Given name(s): _____

Student #: _____

TA's name: _____ (please circle below)

Chao Yu

Vivian Tsang

Xiahui Yu

Diana Inkpen

Please note that if you write the mid term in pencil, you will *not* be allowed to submit a remark request.

1.	/	6
2.	/	14
3.	/	5
Total:	/	25

Question 1: Design by Contract

[6 marks total]

Below is a method, `blah`, that does something to a linked list of `IntNodes` (as well as the code for `IntNode`)

<pre>void blah(IntNode f, int z, int x) { IntNode i = f; IntNode t = null; while (i.value != x) { if (i.value < z) { t.next = i.next; } t = i; i=i.next; } }</pre>	<pre>class IntNode { public int value = 0; public IntNode next = null; }</pre>
---	--

Part A: Preconditions:

Below, list any preconditions (requirements) for the method `blah`
(2 marks)

Part B: Method comment:

Below, write an appropriate method comment for the method `blah`. Phrase your comments at a high level – don't just re-write the Java code in English.
(2 marks)

Part C: Representation Invariant:

A Representation Invariant explains two different aspects of a class. What are they?
(2 marks)

- 1.
- 2.

Question 2: Queue & Linked Lists

[14 marks]

On the last page of the exam, you will find the code for the `Queue` interface. We will be writing a new class, `LinkedList`, that will implement `Queue` and use a linked-list of `LLNodes` to maintain the `Queue`.

(a) First, give the header for the class:

[2 marks]

(b) I have written the beginning of the class, including the `enqueue` method. You will write the `dequeue` and `size` methods. Note that you may *not* add any instance variables to the class, or modify `enqueue` in any way.

Note that you *must* write good internal documentation for your methods, including algorithm & other comments.

[12 marks: 8 for implementation and 4 for comments]

See the next page for the contents of the `LinkedList` class. Here is the code for the `LLNode` class:

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

    public LLNode(Object o) { this.data = o; }
    public LLNode(Object o, LLNode n) { this.data = o; this.next = n; }
}
```

```
LLNode tail = null; // the tail of the queue.
// NOTE: YOU MAY NOT ADD ANY INSTANCE VARIABLES TO THE CLASS

// Add o to the front of the Queue.
void enqueue(Object o) {

    // Algorithm: add o to the head of the Queue, moving all other
    // elements down one (do not modify this method)

    LLNode temp = new LLNode(o); // to hold the new object, will be
                                // inserted into the queue.

    temp.next = tail; // put the new object at the tail of the queue

    tail = temp; // update the pointer

} // end enqueue

// Remove and return the element at the front of the Queue
// Precondition: the stack must not be empty
Object dequeue() {

} // end dequeue

//Returns the number of elements in the queue.
int size() {

} // end size
} // end LinkedList
```

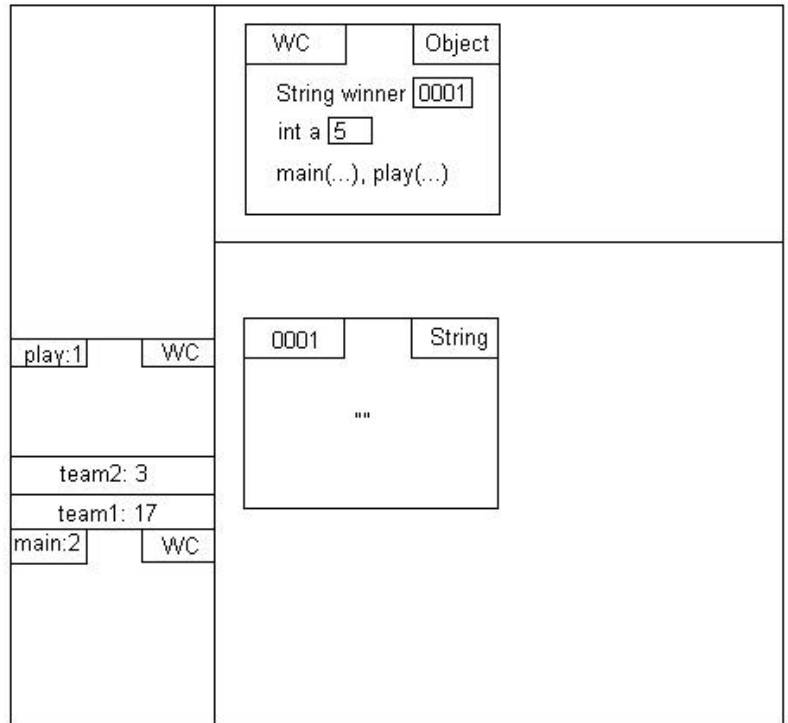
Question 3: Memory Model Tracing

[5 marks]

Here is a memory model (this leaves out the address & object for argv, but don't worry about that). This memory model is from running the `main()` method in class `WC`.

Below are 4 possible classes `WC`. Indicate which class matches this memory model by circling the letter above the class.

(the line numbering for method `play()` in the memory model may be slightly off – don't let this bother you)



<p>A:</p> <pre>class WC { static String winner = null; static int a = 5; public static void main(String [] argv) { winner = ""; play(3,17); } private static int play(int team1, int team2) { // pause here } }</pre>	<p>B:</p> <pre>class WC { static String winner = ""; static int a = 2; public static void main(String [] argv) { a = 5; play(17,3); } private static int play(int team1, int team2) { // pause here } }</pre>
<p>C:</p> <pre>class WC { static String winner = ""; int a = 2; public static void main(String [] argv) { a = 5; play(17,3); } private static int play(int team1, int team2) { // pause here } }</pre>	<p>D:</p> <pre>class WC { static String winner = ""; static int a = 2; public static void main(String [] argv) { int a = 5; play(17,3); } private static int play(int team1, int team2) { // pause here } }</pre>

This page intentionally left blank