

Mid Term Test – Section L5101

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)

Lisa Liu Ying Qiu Patricio Simari

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 an array of ints

```
void blah(int[] a, int z, int x) {
    int i = 0;
    while (a[i] != x) {
        if (a[i] > z) {
            a[i] = 0;
        }
        i++;
    }
}
```

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]

You will be writing a new class, `LinkedList`, that will implement `Queue` and use a linked-list of `LLNode`s to maintain the `Queue`.

On the next page, I have written the beginning of the class, including the `dequeue` method. You will write:

1. The class header (2 marks)
2. the `enqueue` method (4 marks for implementation, 2 for comments)
3. the `size` method (4 marks for implementation, 2 for comments)

Note that you may *not* add any instance variables to the class, or modify `dequeue` in any way.

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; }
}
```

```
class _____ [2 marks]

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

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

        // Algorithm: grab the first object, remove the node,
        // return the object. (do not modify this method)

        Object temp = head.data; // the object to be returned

        head = head.next; // remove the head from the list

        return temp;

    } // end dequeue

    // Add o to the back of the Queue.
    Void enqueue(Object o) { [6 marks]

} // end enqueue

//Returns the number of elements in the queue.
int size() { [6 marks]

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

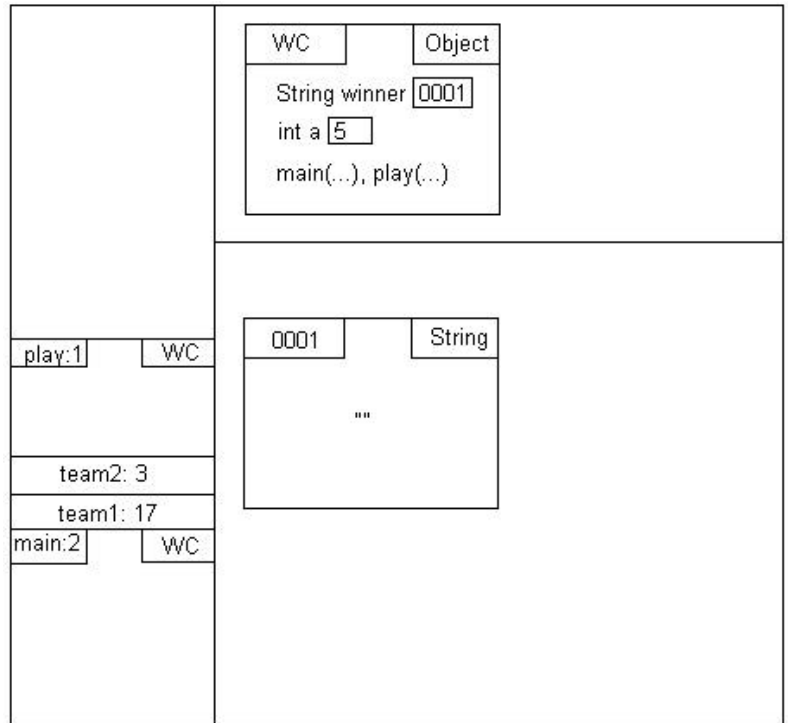
Question 3: Memory Model Tracing

[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 = ""; 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>B:</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>
<p>C:</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>D:</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>

This page intentionally left blank