Surname: ______________________  First name: ______________________

Student #: ____________________  Tutor: ______________________

Instructions

Bring this handout to tutorial will the short answer questions filled in. Also bring a printout of your program from step 2, and printed results from steps 2 and 5. Staple everything together.

Suppose we were to add the following new method to class CircularQueue (see slides 30-32 in the handbook).

```java
// Make me empty.
public void reset() {
    size = 0;
}
```

1. Write down a sequence of queue operations that never violates a method precondition, yet leads to a wrong answer. That is, show that there is a bug. You can describe the operations in English, like “Enqueue ‘a’, then enqueue “b.”

2. Write and run a small program that executes this sequence of operations to prove that you were right about the bug. You can download CircularQueue from the web.

3. Method reset() fails to maintain the representation invariant for class CircularQueue (see slide 30). That is, after calling reset(), some part of the representation invariant may not be true. What part is that?

4. Below, rewrite method reset() to both maintain the representation invariant and fix the bug.

```java
// Make me empty.
public void reset() {

```
5. Type in this new version of `reset()`, and run your little test program again to confirm that the bug is gone (at least in that one test case).

6. Read the document “Software Testing” in the course handbook. Think about a more thorough set of test cases for class `CircularQueue`. You don’t have to hand in anything for this part.