

CS 62 Quiz

Race Conditions

Mawhorter Spring 2017

(Please write your name at the bottom of the back page of this quiz)

1. Define the term “race condition.”

2. Consider the code on the back of this page. It implements an **Amoeba** class where each **Amoeba** runs in a separate thread, and they attempt to absorb each other, getting bigger when they succeed (the code for **main** has been left out, but we assume it creates multiple **Amoeba** objects and calls their **start** methods, thus triggering a call to their **run** methods in a new thread).

Because this code does not use the **synchronized** keyword, it has several race conditions. Explain *one* scenario **involving a race condition** where the **Amoeba** code fails. In your explanation, clearly state **what behavior occurs** and **the specific sequence of events leading to it**. *Hint: look at the **kill** method.*

```

class Amoeba extends Thread {
    protected boolean alive;
    protected int size;
    protected static final Random rng = new Random();
    protected static List<Amoeba> population = new ArrayList<Amoeba>();

    public Amoeba() {
        this.alive = True;
        this.size = 3 + Amoeba.rng.nextInt() % 5;
        Amoeba.population.add(this);
    }

    public void kill() {
        this.alive = False; // to stop the while loop in run
        Amoeba.population.remove(this);
    }

    public void run() {
        while (this.alive) {
            Thread.sleep(Amoeba.rng.nextInt() % 50); // sleep randomly

            // pick a random target and try to absorb it
            Amoeba prey = Amoeba.population.get(
                Amoeba.rng.nextInt() % Amoeba.population.size()
            );

            // we can only absorb things smaller than us
            if (this.size > prey.size) {
                prey.kill();
                this.size += prey.size; // size is conserved
            }
        }
    }
}

```