

Threads/Synchronization Exercise I

1. What are two advantages to multi-threading?
2. What are two disadvantages to using multi-threading?
3. What are two ways to create threads in Java?
4. What is a daemon thread?
5. What is a deadlock? How can you avoid it?
6. What is a data race? How can you avoid it?
7. Give an example of Java code with a data race.
 - a. Eliminate the data race using synchronized methods, e.g., `synchronized foo() { ... }`
 - b. Eliminate the data race using synchronized objects, e.g., `synchronized(bar) { ... }`
8. The following class implements a model of a student dining hall serving pizzas to students. 10 pizzas are baked, then served to 20 students. Students are numbered between 0 and 19 in the order they are served. A message is printed indicating whether a student starved or was served a pizza.
 - a. Rewrite the DiningHall class so that after the `makePizza()` method is called 10 times, the `servePizza()` method is called once each from 20 different threads.
 - b. Insert synchronization to eliminate data races in your code, if any exist.
 - c. Describe what data races may occur in your multithreaded code without synchronization.

```
public class DiningHall {
    static int pizzaNum;
    static int studentID;
    public void makePizza() { pizzaNum++; }
    public void servePizza() {
        String result;
        if (pizzaNum > 0) { result = "Served "; pizzaNum--; }
        else result = "Starved ";
        System.out.println(result + studentID);
        studentID++;
    }
    public static void main(String[] args) {
        DiningHall d = new DiningHall();
        for (int i = 0; i < 10; i++)
            d.makePizza();
        for (int i = 0; i < 20; i++)
            d.servePizza();
    }
}
```