Thread Exercise

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 and print out a message as to whether they received 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.

```java
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();
    }
}
```