The Chocolate Factory

Consider the following code to work on the exercises below

```
public class ChocolateBoiler {
        private boolean empty;
        private boolean boiled;

        public ChocolateBoiler() {     //public constructor!
                empty = true;
                boiled = false;
        }

        public void fill() {
                if (isEmpty()) {
                        empty = false;
                        boiled = false;
                        // fill the boiler with a milk/chocolate mixture
                }
        }

        public void drain() {
                if (!isEmpty() && isBoiled()) {
                        // drain the boiled milk and chocolate
                        empty = true;
                }
        }

        public void boil() {
                if (!isEmpty() && !isBoiled()) {
                        // bring the contents to a boil
                        boiled = true;
                }
        }

        public boolean isEmpty() {
                return empty;
        }

        public boolean isBoiled() {
                return boiled;
        }
}
```

1. Understand the code, what it does?

2. Strength point of the code?

3. What will happen if we have multiple instances of the class ChocolateBoiler?

4. Now improve the code using the Singleton, for each of the following you need to have a classes
    a. Assuming the ChocolateBoiler will only work on one thread, rewrite the code to use the idea of Singleton (lazy instantiation)

    b. to make the ChocolateBoiler thread-safe,
        i. use the idea of synchronization
        ii. eager instantiation
        iii. double-checked