Immutable Object

An immutable object doesn't change state, once it's created.

An immutable object is a secure object, meaning calling code can't maliciously or mistakenly alter it.

An immutable object simplifies concurrency design.

Strategies for Declaring a Class, to produce immutable objects

This slide describes the strategies of creating a class, that when used, produces immutable objects.

- Make instance fields private and final.
- Do not define any setter methods.
- Create defensive copies in any getters.
- Use a constructor or factory method to set data, making copies of mutable reference data.
- Mark the class final, or make all constructors private.

Need your objects to be mutable?

If your class must be mutable, you should still use some of these techniques to minimize mutability!