Supplement: Extended Discussion on Immutable Objects
For Introduction to Java Programming
By Y. Daniel Liang

Assume class <u>A</u> is immutable. You define a mutable subclass <u>B</u> that extends class <u>A</u>. An instance <u>b</u> of class <u>B</u> is mutable. Since an instance of class <u>B</u> is also an instance of class <u>A</u>. Now an instance <u>b</u> of class <u>A</u> is mutable. This contradicts to the spirits of immutability of class <u>A</u>. It should not be allowed. To prevent this from happening, you should define A as a final class using the final modifier, which will be introduced in Chapter 11.

So to define an immutable class, you need to do the following:

- Make all data fields private;
- Provide no mutator methods for data fields;
- Provide no accessor method that returns a reference to a data field that is mutable.
- Define the class as a final class.