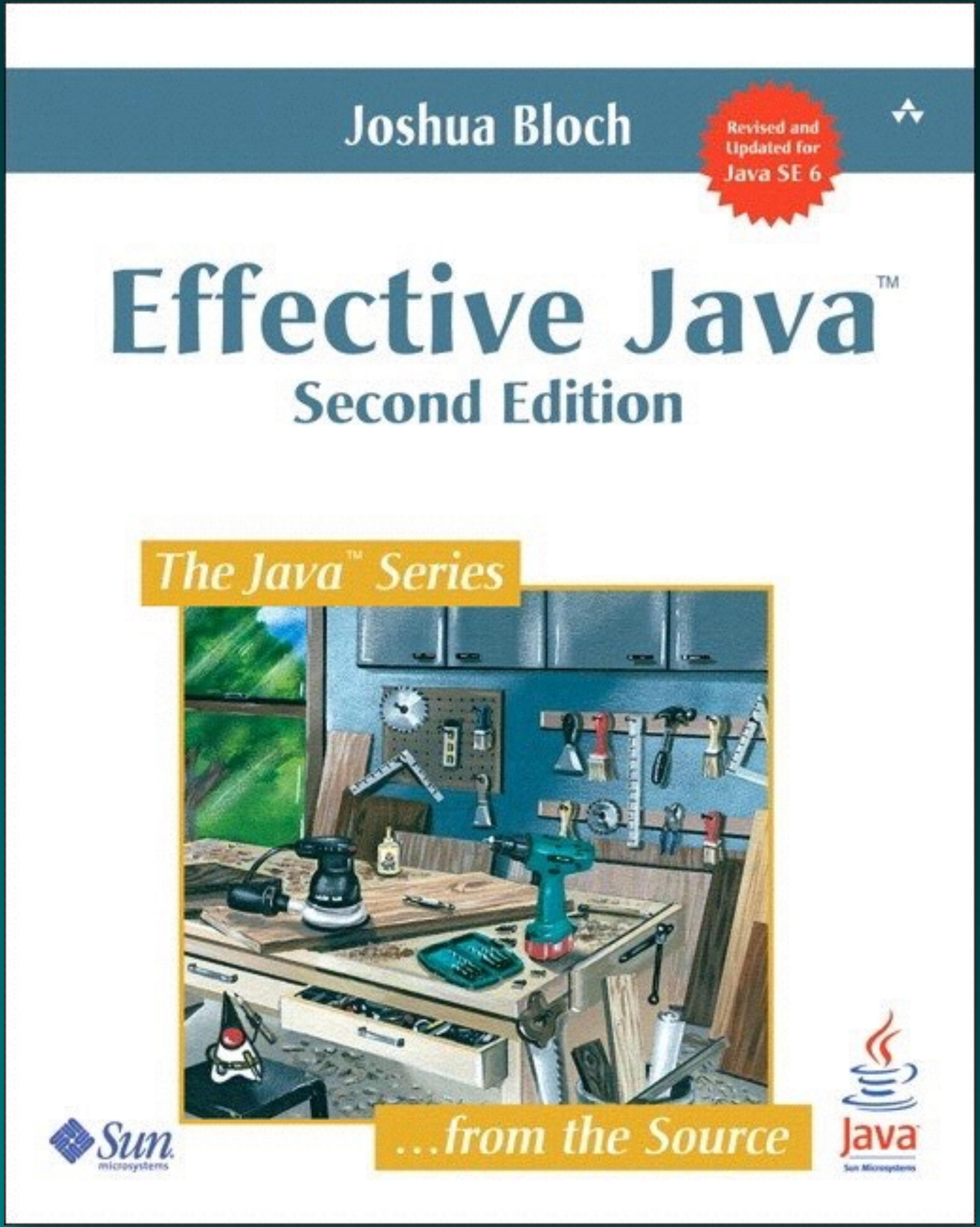

Joshua Bloch

Revised and Updated for Java SE 6

Effective Java™

Second Edition

The Java™ Series

…from the Source

Sun microsystems

Java Sun Microsystems

Classes should be *immutable* unless there is a very good reason to make them *mutable*.

Joshua Bloch

If a class cannot be made immutable, *limit its mutability* as much as possible.

Make every field *final* unless there is a compelling reason to make it nonfinal.

Joshua Bloch

Singleton classes represent objects for which only one single instance should exist.

– javapractices.com

OK

```java
public class Elvis {
  private static final Elvis instance = new Elvis();
  private Elvis() { ... }
  public Elvis getInstance() { return instance; }
  public void leaveTheBuilding() { ... }
}
```

✓

```java
public enum Elvis {
  INSTANCE;
  public void leaveBuilding() { ... }
}
```

```
public class StudentRecord {

  private String name;
  private int[] quizScores;

  public String getName() {
    return name;
  }

  public int getQuizScore(int n) {
    return quizScore[n];
  }

  public int[] getAllQuizScores() {
    return quizScores;
  }

  ...
```

⊗

```java
List roster = new ArrayList();
```

no generics

⊗

```java
List<Person> roster = new ArrayList();
```

raw type

⊗

```java
List<Person> roster = new ArrayList<Person>();
```

redundant

⊗

```java
ArrayList<Person> roster = new ArrayList<>();
```

use interface

✓

```java
List<Person> roster = new ArrayList<>();
```

Which type implements Comparable?

- ☐ String
- ☐ LocalDate
- ☐ Sex (enum type)

Which type implements Comparable?

- ☑ String
- ☑ LocalDate
- ☑ Sex (enum type)

```
public interface Comparable<T> {
   public int compareTo(T obj);
}
```

functional interface

```
public class Name implements Comparable<Name> {
  ...
  @Overrides
  public int compareTo(Name name) { ... }
```

example usage

```
 this.compareTo(that) < 0  // this < that
 this.compareTo(that) == 0 // this == that
 this.compareTo(that) > 0  // this > that
```

compareTo meaning

What important property do these fields share?

```
String name;
LocalDate birthday;
Sex gender;   // enum
String emailAddress;
```

What important property do these fields share?

```
String name;
LocalDate birthday;
Sex gender;   // enum
String emailAddress;
```

Immutability!

Is this Person class immutable?

Is this Person class immutable?

No! – But it *should* be ☺