

SE 350 - Object-Oriented Software Development

Java Coding Conventions

Instructor: Stefan Mitsch



Coding Conventions

- What are coding conventions?
 - Sets of guidelines about style and best practices
 - File organization, indentation, comments, declarations, statements, white space, naming conventions, ...
- Benefits
 - Improve readability
 - Maintain unified code style
 - Reduce maintenance cost
 - Faster development
- [Oracle Coding Conventions \(1999\)](#)
- [Google Java Style Guide \(2018\)](#)



Source File Conventions

- Length <2000LOC
- Package declaration
- Documentation comment
- Imports
- Class comment
- Class/interface signature
- [Java File Organization](#)

```
package edu.depaul.cdm.soc.se350;

/**
 * Demonstrates Java File Organization
 */
import java.util.List;

/**
 * Class comment
 */
public class Demo {
    // Static variables, ordered public to private
    public static final int MY_CONST = 1;
    private static final int OTHER = 2;
    // Instance variables, ordered public to private
    private int i;
    // Constructors, overloaded in sequential order
    public Demo() { this(3); }
    public Demo(int i) { this.i = i; }
    // Methods, ordered public to private
    public void print() { System.out.println("i = " + i); }
}
```



Naming Conventions

- Set of rules to make code uniform
- Prefer short and descriptive names
- Package: all lowercase
`edu.depaul.cdm.soc.se350`
- Class, enum, interface, and annotation:
PascalCase `MyClass`
- Methods, field names, variables:
camelCase `printMe`
- Constants: all caps + underscore
`MY_CONST`
- File: matches the class name
`MyClass.java`

```
// prefer                                // avoid (too detailed or redundant)
int schoolId;                            // identifierOfSchool;
int[] uniqueSchoolIds;                   // schoolIdsAfterRemovingDuplicates;
Map<Integer, User> usersById;             // idToUserMap
String value;                             // valueString
```