

Refactoring

Table of contents

Today's outline	1
Refactoring	1
Cohesion and coupling	1
SOLID principles	2

See [csse2002¹](#) for course logistics.

Today's outline

1. Good Practice Refactoring, When to Refactor, Code Smells
2. Cohesion and Coupling
3. SOLID Principles

Refactoring

See [java-refactoring²](#) — what refactoring is, why it's needed, good practice, and code smells (bad naming, duplication, feature envy).

Cohesion and coupling

See [java-cohesion-and-coupling³](#) — modularization, the levels of coupling (content → data) and cohesion (coincidental → functional), and why high cohesion + low coupling is the goal.

¹[courses/csse2002/csse2002.pdf](#)

²[courses/csse2002/java-refactoring.pdf](#)

³[courses/csse2002/java-cohesion-and-coupling.pdf](#)

SOLID principles

See [java-solid-principles⁴](#) — why large software becomes rigid/fragile, and the first two SOLID principles: Single Responsibility and Open-Closed (the rest are covered in a later lecture).

⁴[courses/csse2002/java-solid-principles.pdf](#)