Object Oriented Programming II

Table of contents

See csse2002[1] for course logistics. Continues 2026-03-05-object-oriented-programming-i[2].

Today's outline

1. Object Oriented Programming II
2. Specification

Recap: the four (five) pillars

Last week covered Objects/Classes/OOP, Encapsulation, Inheritance, and Abstraction. This week adds **Polymorphism**, plus revisits/extends a few Week 2 topics in more depth.

Interface vs abstract class

Week 2 left the "when to use which" comparison as homework — now covered properly in java-abstraction[3].

[1] courses/csse2002/csse2002.pdf
[2] courses/csse2002/2026-03-05-object-oriented-programming-i.pdf
[3] courses/csse2002/java-abstraction.pdf

Revising encapsulation

A quick recap of java-encapsulation[4]'s `Person` example — bundling fields with the methods that access them, restricting direct access via `private` + getters.

Mutable vs immutable classes

See java-mutability[5] — object state that can (mutable) or cannot (immutable) change after construction, and the arrays/lists gotcha where `final` only protects the reference, not the contents.

Polymorphism

See java-polymorphism[6] — compile-time (overloading), runtime (overriding/dynamic dispatch), and subtype polymorphism.

Casting

See java-casting[7] — primitive widening/narrowing casts, and reference upcasting/downcasting.

Specification

See java-specification[8] — Javadoc, restrictiveness, generality, clarity, formality, contracts, and defensive programming.

[4] courses/csse2002/java-encapsulation.pdf
[5] courses/csse2002/java-mutability.pdf
[6] courses/csse2002/java-polymorphism.pdf
[7] courses/csse2002/java-casting.pdf
[8] courses/csse2002/java-specification.pdf