

SOCKET
JAVAFUNDAMENTALS → OBJECT ORIENTED DESIGN:
OO → Very Inheritance & Polymorphism

↳ Simplicity

↳ Code Reusability

→ Design & Build Incrementally

Coupling → Change to one thing impacts another one

Every object → Standalone

Object is responsible for its data

Messages → objects Send messages to one another

Methods → handle messages

Object has info that works

Functions → do arbitrary work

Fragility } → use 'final'

↳ to Overcome

Changes to Superclass can break Subclass

Delegation is important

Under Inheritance

Ball & socket version can be useful

↳ open → for Extension

closed → for modification

Change the object witho

Changing the test

↳ Must be able to replace implementation without impacting clients

Delegation → Ask for help, not Information

Prefer implements (uses) to Extends (is)