

Object-Oriented Programming

Dependency inversion changes everything

- None is unique to OO
- Encapsulation & Inheritance are trivial
- Polymorphism is the **KEY**
 - Lead to a safe and efficient replacement for function pointers
 - Enable *dependency inversion*
 - Fully control the direction of all source code dependencies in the system
 - High-level policies are independent from low-level details
 - Independent develop-ability and deployability

Functional Programming

Variables are immutable

- Concurrent applications
 - All race conditions, deadlock conditions and concurrent update problems are due to mutable variables
- Immutability is not always practical
- Segregation of application into mutable and immutable components
 - Push as much as processing as possible into the immutable components
 - Event sourcing to store transactions, but not the state
 - Appropriate disciplines to protect mutable variables
 - Transactional memory, compare & swap algorithm