

## Benefits of Design Patterns

- Design patterns enable large-scale reuse of software architecture
  - They also help document systems to enhance understanding
- Patterns explicitly capture expert knowledge and design tradeoffs, and make this expertise more widely available
- Patterns help improve developer communication
  - Pattern names forms a vocabulary
- Patterns help ease the transition to object-oriented technology

78

## Drawbacks to Patterns

- Patterns do not lead to direct code reuse
- Patterns are deceptively simple
- Teams may suffer from pattern overload
- Patterns are validated by experience and discussion rather than by automated testing
- Integrating patterns into a software development process is a human-intensive activity

79

## How to Use Patterns Effectively

- Do not recast everything as a pattern
- Institutionalize rewards for developing patterns
- Directly involve pattern authors with application developers and domain experts
- Clearly document when patterns apply and do not apply
- Manage expectations carefully

80

## Summary

- An introduction to design patterns
- A sorting example containing five design patterns
- Details of several design patterns
- Strategy
- Bridge
- Adapter
- Iterator
- Abstract Factory
- Composite
- Observer

81

---

## Summary

- Mature engineering disciplines have handbooks that describe successful solutions to known problems
    - E.G., Automobile designers don't design cars using the laws of physics, they adapt adequate solutions from the handbook known to work well enough
    - They don't start from scratch to gain the few percent of performance
- 

82

---

## Summary

- Patterns can form the basis for the handbook of software engineering
    - If software is to become an engineering discipline, successful practices must be systematically documented and widely disseminated
- 

83