

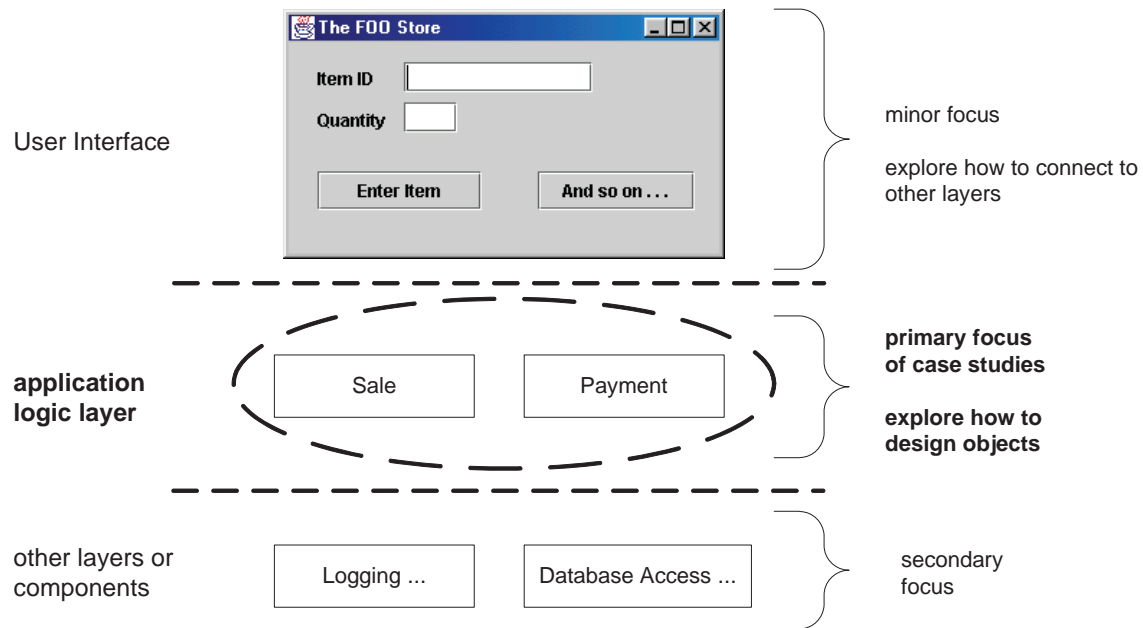
Chapter 3

Case Studies

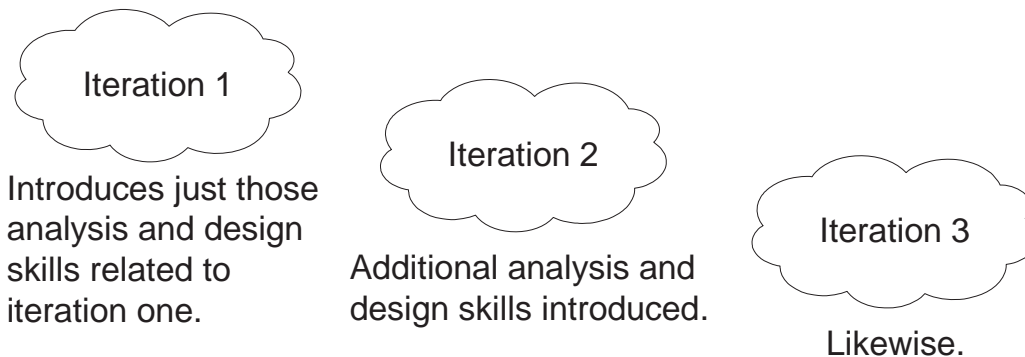
Case Study Focus

- Applications generally can be divided into 3 layers
 - User interface
 - Application logic
 - Other components/layers
- Focus on application logic layer because:

Sample Layers & Objects in OO System



Learning Path Following Iterations



Case Study I: NextGen POS System

- NextGen point-of-sale (POS) system
 - Computerized application used (in part) to record sales and handle payments typically used in a retail store
- Components
 - Hardware: computer and bar code scanner etc.
 - Software
- Interfaces to various service applications, such as a third-party tax calculator and inventory control
- Must be relatively fault-tolerant
 - Even if remote services are temporarily unavailable (such as the inventory system), they must still be capable of capturing sales and handling at least cash payments
- Increasingly must support multiple and varied client-side terminals and interfaces
 - Thin-client Web browser terminal
 - Regular personal computer with graphical user interface
 - Touch screen input
 - Wireless PDAs, etc.

Case Study I: NextGen POS System

- Sold to different clients with disparate needs in terms of business rule processing
- Each client will desire a unique set of logic to execute at certain predictable points in scenarios of using the system, such as
 - When a new sale is initiated
 - When a new line item is added
- Need a mechanism to provide flexibility and customization.
- Using an iterative development strategy, we are going to proceed through requirements, object-oriented analysis, design, and implementation.



Case Study II: Monopoly Game System

- Domain and requirements not at all like business system such as the NextGen POS
- Still relevant and useful:
 - Domain modeling
 - Object design with patterns
 - Applying the UML
- Run as a simulation.
 - One person starts the game
 - Indicate number of simulated players
 - Watch while the game runs to completion
 - Presenting a trace of activity during simulated player turns

