

Copyright © 2016 The Learning House.

All rights reserved. No part of these materials may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of The Learning House. For permission requests, write to The Learning House, addressed “Attention: Permissions Coordinator,” at the address below.

The Learning House  
427 S 4<sup>th</sup> Street #300  
Louisville KY 40202



---

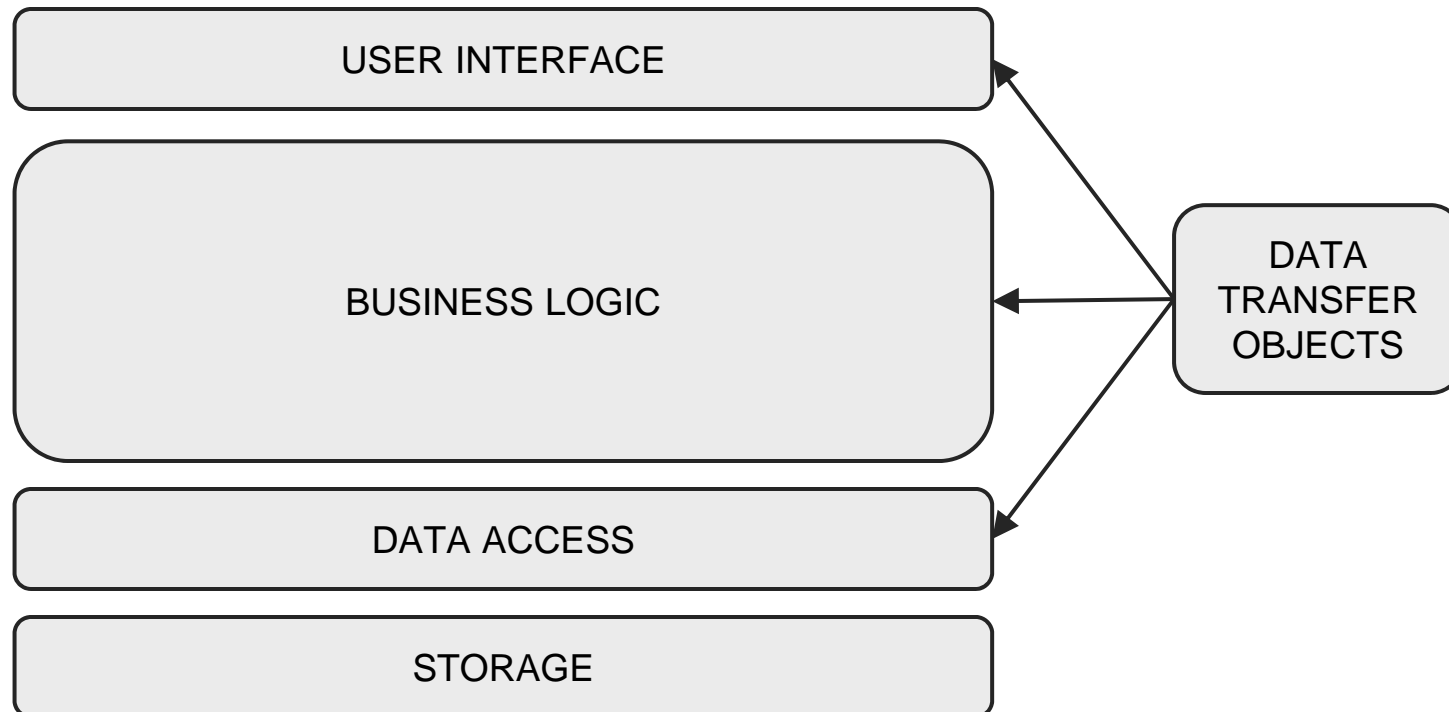
# Java Object-Oriented Concepts

## Lesson 7 - Tiered Application Design Concepts

# Why Tiers?

- We want to keep business logic and data access logic separate from UI logic
- This allows us to reuse these components for several platforms (web, mobile, B2B)
- Without, we would have to duplicate code for each platform

# Tiered Design



# Tiered Design

- **Storage:** files, database, or other persistent storage
- **Data Access:** classes that handle retrieving and storing data (think VendingMachine or AddressBook)
- **Business Logic:** classes that handle algorithms specific to your problem domain
- **User Interface:** interaction with the user

# Packages

- Each of the major areas in the tiered model usually gets its own package
- For example:
  - Data Transfer objects — domain or dto package
  - Data Access objects — persistence or dao package
  - Business Logic objects — operations package
  - User Interface objects — ui package

# Tiered Design and Web

- In our labs so far, tiered design is complete overkill
- We will use this design approach as we move into web applications
- We will use a framework called Model-View-Controller (MVC)
- MVC will seem familiar because we already use this approach