

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 5 - Exceptions

# Objectives

- Define **exception**
- Understand how to handle exceptions
- Understand the difference between **checked** and **unchecked** exceptions
- Understand the **Catch or Specify** requirement
- Understand what the **finally** block does

# Exception

- Short for 'exceptional event'
- An exception is an event which occurs in the normal execution of a program that disrupts the normal flow of the program's instructions.
  - Oracle documentation

# Catch or Specify

- Code that might throw an exception must:
  - Contain code to handle the possible exception
  - OR
  - Specify that it may throw the possible exception
- Code that does not meet this requirement will not compile
- Not all exceptions are created equal... some exceptions are not subject to this rule

# Exception Types

1. Checked exceptions
  - a. Subject to Catch or Specify
2. Errors
  - a. Not subject to Catch or Specify
3. Runtime exceptions
  - a. Not subject to Catch or Specify
4. Errors and Runtime exceptions are known as 'unchecked exceptions'

# Handling Exceptions

## First Step: Try block

```
try {  
    code goes here  
}
```

- Surrounds the code that might throw the exception

# Handling Exceptions

## Second Step: Catch block

```
try {  
    code  
} catch (ExceptionType name) {  
    code  
} catch (ExceptionType name) {  
    code  
}
```

- Contains code to react to the exception if it occurs



# Handling Exceptions

## Third Step: Finally block

- `try {`
- `code`
- `} catch (ExceptionType name) {`
- `code`
- `} finally {`
- `code`
- `}`
- Always executes

# Example

## File I/O

# Specifying Exceptions Thrown by a Method

- Only have to specify checked exceptions
- Use the throws keyword