

[Introduction to Interactive Programming](#)

by [Lynn Andrea Stein](#)

A [Rethinking CS101](#) Project

Interactive Programming In Java

Chapter Outlines

[Lynn Andrea Stein](#)

[Massachusetts Institute of Technology](#)

[Department of Electrical Engineering and Computer Science](#)

Cambridge, MA 02139 USA

ipij@ai.mit.edu

<http://www.ai.mit.edu/projects/cs101/>

Front Matter

- I. Table of Contents
2. [Preface](#)
 1. Why Interactive Programming?
 2. Ramifications for Later Curriculum
 3. A Short History of the Rethinking CS101 Project
 1. Research Roots
 2. Classroom Experience
 4. How to Use This Book
 1. Part By Part
 2. Pedagogical Elements and Supplementary Materials
 5. About the Author
 6. Acknowledgements
3. List of Figures
4. List of Tables
5. [List of Sidebars](#)
 1. Java Sidebars
 2. Style Sidebars

Part 1: Introduction To Interactive Program Design

Chapter 1: [Introduction to Program Design](#)

Chapter Overview

Objectives of this Chapter

1. Computers and Programs
2. Thinking Like a Programmer
3. Programming Primitives, Briefly
4. Ongoing Computational Activity
5. Coordinating a Computational Community
 1. What is the Desired Behavior of the Program?
 2. Who are the Members of the Community?
 3. What Goes Inside Each One?
 4. How Do They Interact?
6. The Development Cycle
7. The Interactive Control Loop

Chapter Summary

Exercises

Chapter 2: The Programming Process

Chapter Overview

Objectives of this Chapter

1. The Problem
 1. Problem Requirements
 1. Assumptions
 2. Promises/Guarantees
 2. The Community Around You
 1. Program Libraries
 2. Users
 3. Physical Environment
 4. Understand their interfaces (and assumptions)
 3. Requirements are a Moving Target
 1. Software Lifecycle
 2. Documenting your Design
2. Designing a solution
 1. Who are the members?
 2. How do they Interact

1. What Promises?
2. What Assumptions?
3. Who does what?
3. What goes inside?
4. Nouns and Verbs
5. Acting it out
3. The process
 1. Keep it simple
 2. Keep it working
 3. You'll still have to debug (debugging is normal)
 1. When it's not working
 2. make state manifest
 3. Explain it to someone
 4. Act it out
 4. Documentation

Chapter Summary

Exercises

Interlude: [A Community of Interacting Entities](#)

Overview

Objectives of this Interlude

1. Introduction: Word Games
2. Designing a Community
 1. A Uniform Community of Transformers
 2. The User and the System
 3. What Goes Inside
3. Building a Transformer
 1. Transformer Examples
 2. Strings
 1. String Concatenation
 2. String Methods
 3. Rules and Methods
 4. Classes and Instances
 5. Fields and Customized Parts
 6. Generality of the Approach

4. Summary

Suggested Exercises

- Sidebar: Selected String Methods

Part 2: Entities and Interactions

Chapter 3: [Things, Types, and Names](#)

Chapter Overview

Objectives of this Chapter

1. Things

1. Primitive Things and Literals

1. Numbers
2. Characters and Strings
3. Booleans

2. Objects

2. Naming Things

1. Referring to Things
2. Assignment

3. Types

1. Declarations and The *type-of-thing name-of-thing* Rule
2. Definition = Declaration + Assignment
3. Primitive Types
4. Object Types

4. Types of Names

1. Shoebox Names
2. Label Names

Chapter Summary

Exercises

- Sidebar: Java Naming Syntax and Conventions
- Sidebar: Java Primitive Types

Chapter 4: [Specifying Behavior: Interfaces](#)

Chapter Overview

Objectives of this Chapter

1. Interfaces are Contracts
 1. Generalized Interfaces and Java Interfaces
 2. A Java Interface Example
2. Method Signatures
 1. Name
 2. Parameters and Parameter Types
 3. Return Type
 4. Putting It All Together: Abstract Method Declaration Syntax
 5. What a Signature Doesn't Say
3. Interface Declaration
 1. Syntax
 2. Method Footprints and Unique Names
 3. Interfaces are Types: Behavior Promises
 4. Interfaces are Not Implementations

Chapter Summary

Exercises

- Style Sidebar: Method Documentation
- Style Sidebar: Interface Documentation

See also Java Chart on Interfaces.

Chapter 5: [Expressions: Doing Things With Things](#)

Chapter Overview

Objectives of this Chapter

1. Simple Expressions
 1. Literals
 2. Names
2. Method Invocation
3. Combining Expressions
4. Assignments and Side-Effecting Expressions
5. Other Expressions that Use Objects
 1. Field Access
 2. Instance Creation
 3. Type Membership

6. Complex Expressions on Primitive Types: Operations
 1. Arithmetic Operator Expressions
 2. Explicit Cast Expressions
 3. Comparator Expressions
 4. Logical Operator Expressions
7. Parenthetical Expressions and Precedence

Chapter Summary

Exercises

- Style Sidebar: Don't Embed Side-Effecting Expressions
- Sidebar: Java Operators
- Sidebar: Arithmetic Expressions
- Sidebar: Coercion and Casting
- Sidebar: Java Operator Precedence
- Sidebar: Other Assignment Operators

See also Java Chart on Expressions

Chapter 6: [Statements and Rules](#)

Chapter Overview

Objectives of this Chapter

1. Statements and Instruction-Followers
2. Simple Statements
3. Declarations and Definitions
4. Sequence Statements
5. Flow of Control
 1. Simple Conditionals
 2. Simple Loops
6. Statements and Rules
 1. Method Invocation Execution Sequence
 2. Return

Chapter Summary

Exercises

- Style Sidebar: Formatting Declaration Statements

- Style Sidebar: Formatting Blocks
- Style Sidebar: Using Booleans
- Style Sidebar: Documentation

See also Java Chart on Statements

Interlude: [Entities and Aggregates/Rules and Roles](#)

Overview

Objectives of this Interlude

1. The Problem
2. Representation
3. Interacting with the Rules
4. Paying Attention to the World
5. Fancy Dot Tricks
6. Remembering State
 1. Fields
 2. Fields vs. Variables
7. Summary

Suggested Exercises

Chapter 7: Building New Things: [Classes and Objects](#)

Chapter Overview

Objectives of this Chapter

1. Classes are Object Factories
 1. Classes and Instances
 2. Recipes Don't Taste Good
 3. Classes are Types
2. Class Declaration
 1. Classes and Interfaces
 1. implements and type inclusion
 2. contract vs. implementation
3. Data Members, or Fields
 1. Fields are not Variables
 1. Hotel Rooms and Storage Rental
 2. Whose Data Member is it?

3. Scoping of Fields
4. Comparison of Kinds of Names
2. Static Members
4. Methods
 1. Method Declaration
 2. Method Body and Behavior
 3. A Method ALWAYS Belongs to an Object
 1. this.
 2. Static Methods
 4. Method Overloading
5. Constructors
 1. Constructors are Not Methods
 2. Syntax
 3. Execution Sequence
 4. Multiple Constructors and the Implicit No-Arg Constructor
 5. Constructor Functions

Chapter Summary

Exercises

- Style Sidebar: Class Declaration
- Sidebar: Java Types and Default Initialization
- Table: Comparison of Kinds of Names
- Style Sidebar: Field Documentation
- Style Sidebar: Method Implementation Documentation
- Sidebar: Method Invocation and Execution
- Style Sidebar: Constructor Documentation
- Style Sidebar: Capitalization Conventions

See also Java Charts on Classes, Methods, and Fields.

Part 3: Refining Designs

Chapter 8: [Designing with Objects](#)

Chapter Overview

Objectives of this Chapter

1. Object-Oriented Design

1. Objects are Nouns
 2. Methods are Verbs
 3. Interfaces are Adjectives
 4. Classes are Object Factories
 5. Some Counter Code (An Example)
 6. Public and Private
2. Kinds of Objects
 1. Data Repositories
 2. Resource Libraries
 3. Traditional Objects
3. Types and Objects
 1. Declared Types and Actual Types
 2. Use Interface Types
 3. Use Contained Objects to Implement Behavior
 4. The Power of Interfaces

Chapter Summary

Exercises

- Style Sidebar: Class and Member Documentation
- Sidebar: Final
- Sidebar: class Math

Collections: An Extended Example

Overview

Objectives

1. Exercises

Chapter 9: [Animate Objects](#)

Chapter Overview

Objectives of this Chapter

1. Animate Objects
2. Animacies are Execution Sequences
3. Being Animate-able
 1. Implementing Animate

2. AnimatorThread
 3. Creating the AnimatorThread in the Constructor
 4. A Generic AnimateObject
 4. More Details
 1. AnimatorThread Details
 2. Delayed Start and the init() Trick
 3. Threads and Runnables
 5. Where do Threads come from?
 1. Starting a Program
 2. Why Constructors Need to Return
- Sidebar: class AnimatorThread
 - Sidebar: Thread Methods
 - Sidebar: class Main
 - Style Sidebar: Using main()

Chapter Summary

Exercises

Chapter 10: Reusing Implementation: [Inheritance](#)

Chapter Overview

Objectives of this Chapter

1. Derived Factories
 1. Simple Inheritance
 2. java.lang.Object
 3. Superclass Membership
2. Overriding
 1. super .
 2. The Outside-In Rule
 3. Problems with Private
3. Constructors are Recipes
 1. this()
 2. super()
 3. Implicit super()
4. Interface Inheritance
5. Relationships Between Types

- The class Object
- Style Sidebar: Explicit Use of `this .` and `super ()`
- Sidebar: Abstract Classes

1. Chapter Summary

Exercises

Interlude: A System of Animate Objects

Objectives

Overview

Suggested Exercises

Chapter 11: When Things Go Wrong: [Exceptions](#)

Chapter Overview

Objectives of this Chapter

1. Exceptional Events
 1. When Things Go Wrong
 2. Expecting the Unexpected
 3. What's Important to Record
2. Throwing an Exception
3. Catching an Exception
4. Throw vs. Return
5. Designing Good Test Cases
 - Sidebar: Throw Statements and Throws Clauses
 - Sidebar: Try Statement Syntax
 - Sidebar: Exceptions, Errors, and RuntimeExceptions

1. Chapter Summary

Exercises

Part 4: Refining Interactions

Chapter 12: [Dealing with Difference: Dispatch](#)

Chapter Overview

Objectives of this Chapter

1. Conditional Behavior
2. If and else
 1. Basic Form
 2. Else
 3. Cascaded Ifs
 4. Many Alternatives
3. Limited Options: Switch
 1. Constant Values
 1. Symbolic Constants
 2. Using Constants
 2. Syntax
 1. Basic Form
 2. The Default Case
 3. Variations
 4. Switch Statement Pros and Cons
4. Arrays
 1. What is an Array?
 1. Array Declaration
 2. Array Construction
 3. Array Elements
 2. Manipulating Arrays
 1. Stepping Through an Array Using a For Statement
 2. Using Arrays for Dispatch
5. When to Use Which Construct
 - Sidebar: `if` Statement Syntax
 - Sidebar: `final`
 - Style Sidebar: Use Named Constants
 - Sidebar: `break` and `continue` statements
 - Sidebar: `switch` Statement Syntax
 - Sidebar: Array Syntax
 - Sidebar: `for` Statement Syntax

1. Chapter Summary

Exercises

Chapter 13: [Encapsulation](#)

Chapter Overview

Objectives of this Chapter

1. Design, Abstraction, and Encapsulation
2. Procedural Abstraction
 1. The Description Rule of Thumb
 2. The Length Rule of Thumb
 3. The Repetition Rule of Thumb
 4. An Example
 5. The Benefits of Abstraction
3. Protecting Internal Structure
 1. private
 2. Packages
 1. Packages and Names
 2. Packages and Visibility
 3. Inheritance
 4. Clever Use of Interfaces
4. Inner Classes
 1. Static Classes
 2. Member Classes
 3. Local Classes and Anonymous Classes
5.
 - Style Sidebar: Procedural Abstraction
 - Sidebar: Package Naming Summary
 - Sidebar: Package Visibility Summary
 - Sidebar: Inner Classes
1. Chapter Summary

Exercises

Chapter 14: [Intelligent Objects and Implicit Dispatch](#)

Chapter Overview

Objectives of this Chapter

1. Procedural Encapsulation and Object Encapsulation
2. From Dispatch to Objects
 1. A Straightforward Dispatch

2. Procedural Encapsulation
 3. Variations
 4. Pushing Methods Into Objects
 5. What Happens to the Central Loop?
3. The Use of Interfaces
 4. Runnables as First Class Procedures
 5. Callbacks
 6. Recursion
 1. Structural Recursion
 1. A Recursive Class Definition
 2. Methods and Recursive Structure
 3. The Power of Recursive Structure
 2. Functional Recursion

Chapter Summary

Exercises

Chapter 15: Event-Driven Programming

Chapter Overview

Objectives of this Chapter

1. Control Loops and Handler Methods
 1. Dispatch Revisited
2. Simple Event Handling
 1. A Handler Interface
 2. An Unrealistic Dispatcher
 3. Sharing the Interface
3. Real Event-Driven Programming
 1. Previous Examples
 2. The Idea of an Event Queue
 3. Properties of Event Queues
4. Graphical User Interfaces: An Extended Example
 1. java.awt
 2. Components
 3. Graphics
 4. The Story of `paint`
5. Events and Polymorphism

Chapter Summary

Exercises

See also the AWT Quick Reference.

Interlude: Achieving Customized Behavior

Objectives

Overview

Suggested Exercises

Chapter 16: [Event Delegation \(and AWT\)](#)

Chapter Overview

Objectives of this Chapter

1. Model/View: Separating GUI Behavior from Application Behavior
 1. The Event Queue, Revisited
2. Reading What the User Types: An Example
 1. Setting up a User Interaction
 2. Listening for the Event
 3. Registering Listeners
 4. Recap
3. Specialized Event Objects
4. Listeners and Adapters: A Pragmatic Detail
5. Inner Class Niceties

- Sidebar: `cs101.awt.DefaultFrame`

1. Chapter Summary

Exercises

See also the AWT Quick Reference.

Interlude: An AWT Application

Objectives

Overview

Suggested Exercises

Part 5: Systems of Objects

Chapter 17: Models of Communities

Chapter Overview

Objectives of this Chapter

1. State Machines
2. State Spaces
3. Organizational Behavior
4. Network Models
5. Patterns
6. UML
7. Metrics
 1. Static Complexity
 2. Throughput and Latency

- Sidebar: FSM Rules

1. Chapter Summary

Exercises

Chapter 18: Interfaces and Protocols: Gluing Things Together

Chapter Overview

Objectives of this Chapter

1. Pacing
2. Procedure Calls
3. Callbacks
4. Explicit Communication Channel Objects
5. Protocols

Chapter Summary

Exercises

Chapter 19: [Communication Patterns](#)

Chapter Overview

Objectives of this Chapter

1. What is a Client-Server Interaction?
2. Implementing Client-Server Interactions
 1. Client Pull
 2. Server Push
3. The Nature of Duals
4. Pushing and Pulling Together
 1. Passive Repository
 2. Active Constraint

Chapter Summary

Exercises

Interlude: Combining Events and Interactive Control Loops

Objectives

Overview

Suggested Exercises

Chapter 20: [Synchronization](#)

Chapter Overview

Objectives of this Chapter

1. Reads and Writes
2. An Example of Conflict
3. Synchronization
4. Java synchronized
 1. methods
 2. (blocks)
5. What synchronization buys you
6. Safety rules
7. Deadlock
8. Obscure Details

Chapter Summary

Exercises

Chapter 21: [Network Programming](#)

Chapter Overview

Objectives of this Chapter

1. A Readable Writeable Channel
 1. Tin Can Telephones
 2. Streams
2. Using A Channel
 1. For Writing
 1. Flushing Out the Stream
 2. A Scribe Example
 2. For Reading
 1. Reading and Blocking
 2. A Lector Example
 3. Encapsulating Communications
3. Real Streams
 1. Abstract Stream Classes
 2. Decorator Streams
 3. Stream Sources
 4. Decoration in Action
4. Network Streams: An Example
 1. Starting from Streams
 2. Decorating Streams
 3. Sockets and Ports
 4. Using A Socket
 5. Opening a Client-Side Socket
 6. Opening a Single Server-Side Socket
 7. A Multi-Connection Server
 8. Server Bottlenecks

Chapter Summary

Exercises

Interlude: Client/Server Chat

Objectives

Overview

Suggested Exercises

Chapter 22: Conventional Architectures

Chapter Overview

Objectives of this Chapter

1. Server Architectures
 1. Dumb broadcast server
 2. Routing server
 3. DNS
2. RPC
3. Peer Architectures
 1. Ring
 2. Round Robin
 3. Cubes
4. Arbitration
5. Blackboard
6. Tuple-space

Chapter Summary

Exercises

Appendices

1. [Applets](#)
2. [AWT Quick Reference](#)
 1. AWT Components
 2. Component
 3. Canvas
 4. Widgets and their Event Types
 5. Basic Widgets
 6. ItemSelectable Widgets
 7. Text Widgets
 8. Container
 9. Panel and Frame
 10. Dimension, Point, and Rectangle

11. Graphics
 12. AWT Events
 13. ActionEvent and ActionListener
 14. AWT Listeners and Adapters
3. [IO Quick Reference](#)
 1. InputStream and Reader
 2. OutputStream and Writer
 3. Sources of Streams
 4. InputStreamReader and OutputStreamWriter
 5. Files
 6. Pipes
 7. Streams that Add Features
 8. Buffering
 9. Primitive Data
 10. Object Streams and Serialization
 11. Other Useful Streams
 12. IOExceptions
 4. [Java Charts](#)
 1. About Java Charts
 2. Program File
 3. Class Declaration
 4. Field Declaration
 5. Method Declaration
 6. Expression
 7. Statement
 8. Disclaimers, Notes, Amendments, etc.
 5. [Glossary](#)
 6. Indices
 1. Syntax Sidebars
 2. Style Sidebars
 3. Interludes
 4. Case Studies
 5. Terms

© 2003 Lynn Andrea Stein

This chapter is excerpted from a draft of [Introduction to Interactive Programming In Java](#), a forthcoming textbook. It is a part of the course materials developed as a part of [Lynn Andrea Stein's Rethinking CS101 Project](#) at the [Computers and Cognition Laboratory](#) of the [Franklin W. Olin College of Engineering](#) and

formerly at the [MIT AI Lab](#) and the [Department of Electrical Engineering and Computer Science](#) at the [Massachusetts Institute of Technology](#).

Questions or comments:
<webmaster@cs101.org>

