Introduction to Object-Oriented System Development

What is inside?

- Characteristics of OO development
- OO concepts: object, class, instance, attributes, methods, and encapsulation
- Object interaction through methods and association relationships
- The concept of inheritance applied to classes of objects
- Benefits of using OO development

Object-Oriented System Development

- OO information system development involves:
  - OOA
    - Using an OO approach to system analysis
  - OOD
    - Using an OO approach to system design
  - OOP
    - Using an OO approach to programming

Understanding OO Development

- OO approach
  - System is defined as a collection of objects that work together to accomplish tasks
    - Objects carry out actions when asked
    - Each object maintains its own data
- Procedural approach
  - System is defined as a set of procedures that interact with data
    - Data is maintained separately from procedures

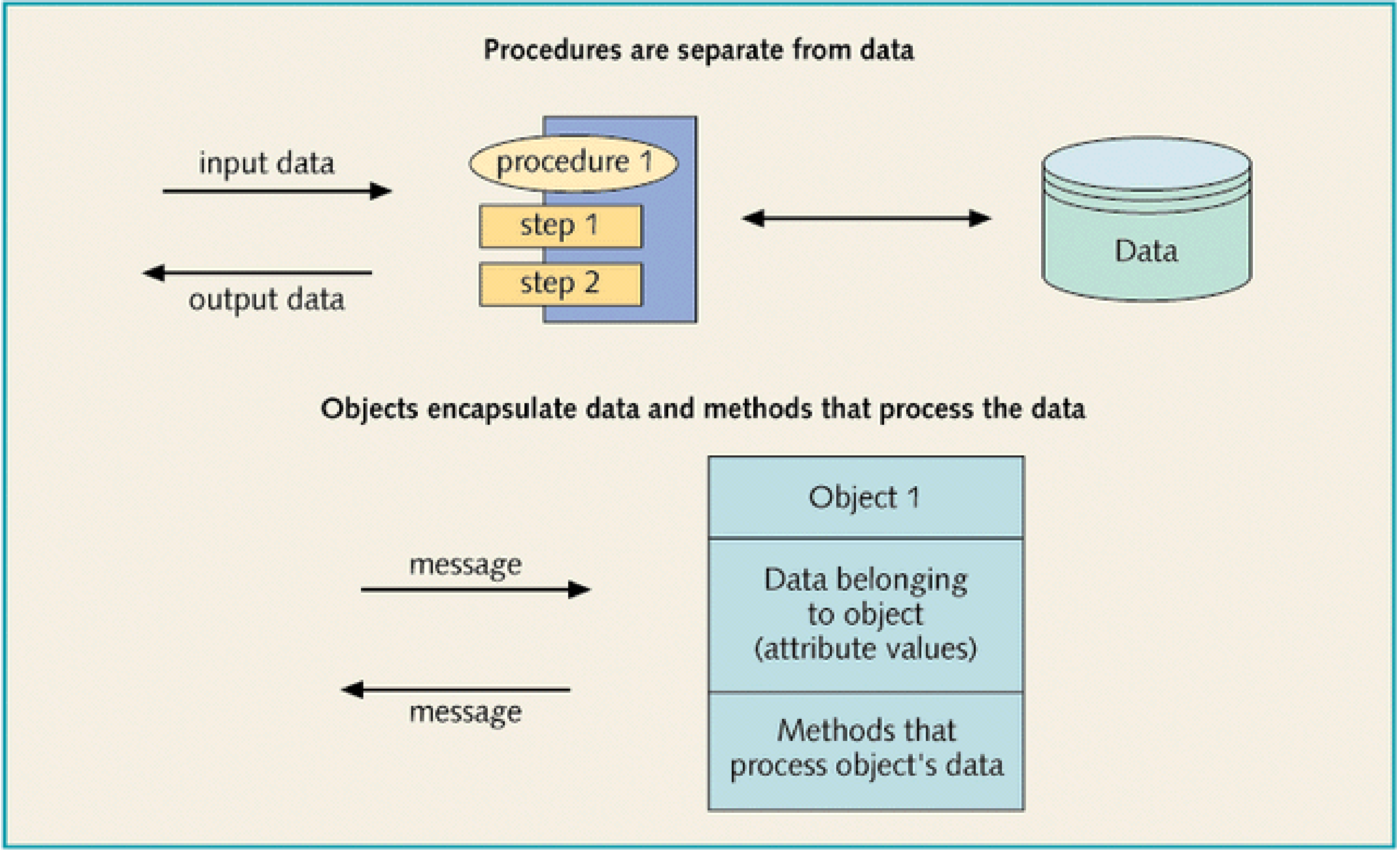

Procedures are separate from data
input data
output data
procedure 1
step 1
step 2
Data
Objects encapsulate data and methods that process the data
message
message
Object 1
Data belonging to object (attribute values)
Methods that process object's data

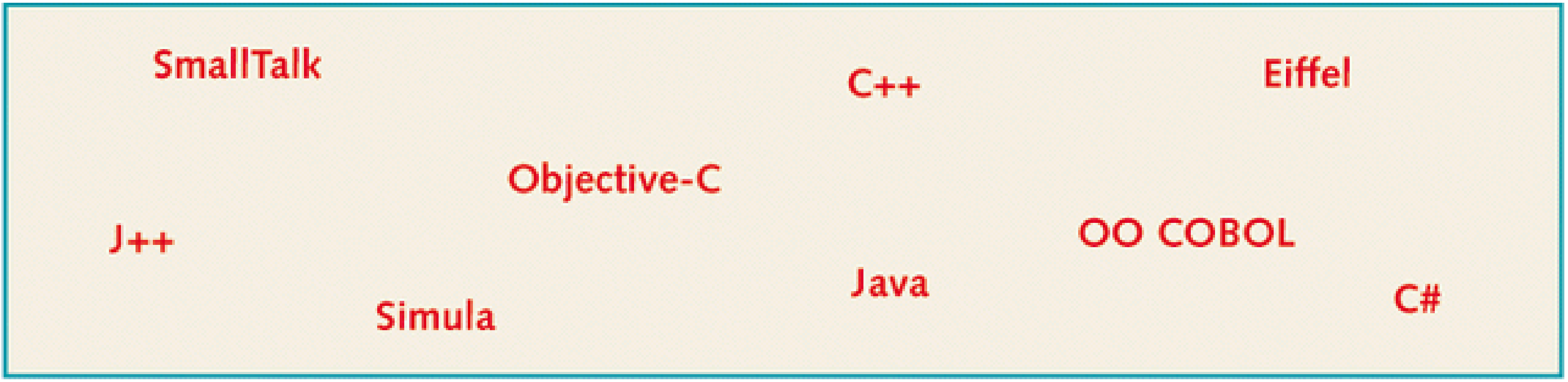
Figure 1-1 Procedural approach versus object-oriented approach

Understanding OO Development

- OO Programming (OOP)
  - Started in 1960s in Norway
    - Simula
      - First language designed to run computer simulations
        - » Simulations involve objects that maintain their own data values and can interact independently
  - 1970s at Xerox PARC
    - Smalltalk
      - First general purpose OO programming language

Understanding OO Development

- OO Programming (OOP)
  - Java
    - Introduced in 1995 by Sun Microsystems
    - “pure” OO language
    - Syntax similar to C++
    - Cross platform
      - Ideal for Web-based applications



SmallTalk

C++

Eiffel

Objective-C

OO COBOL

J++

Java

C#

Simula

Figure 1-2 Some object-oriented programming languages

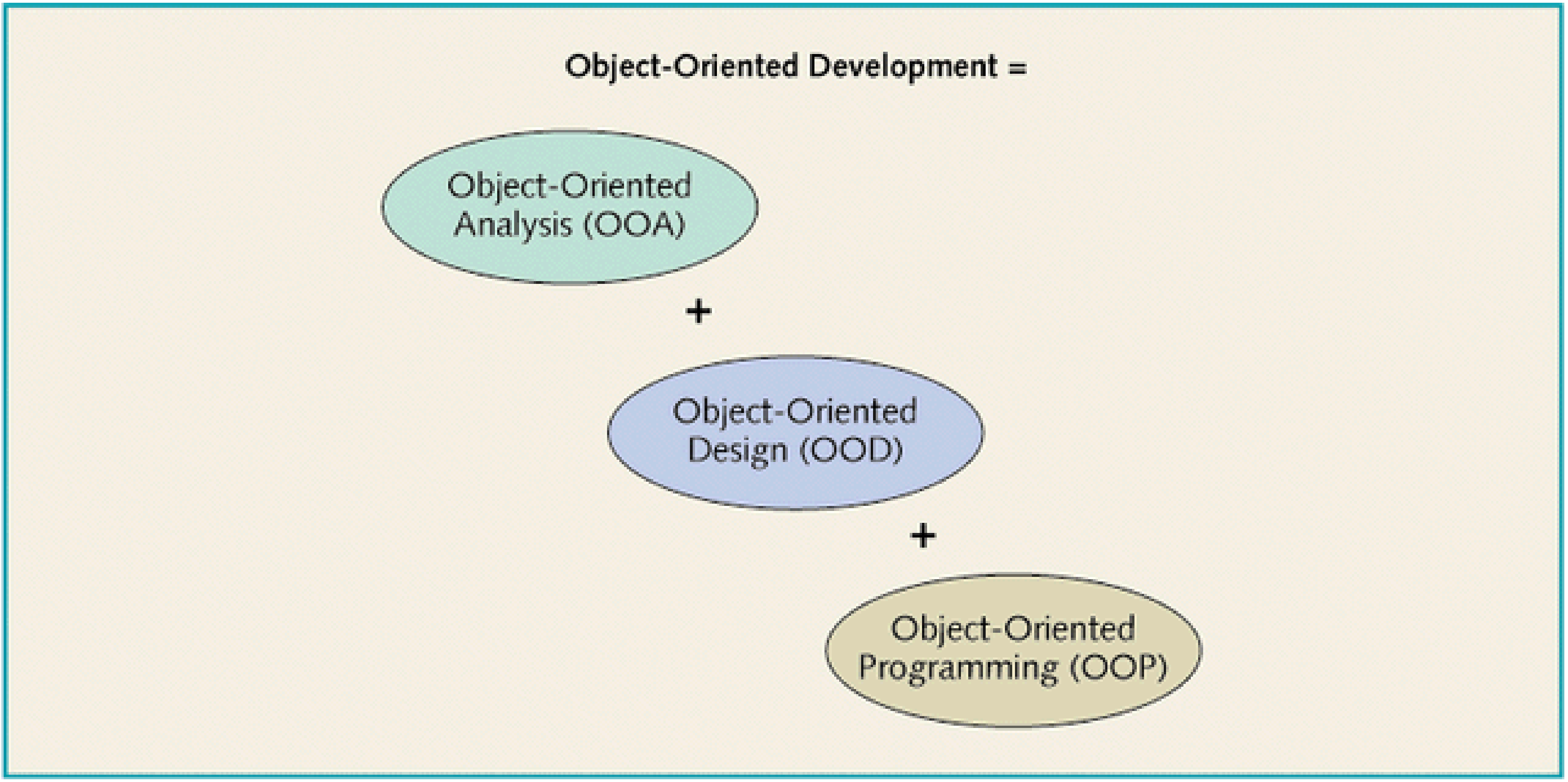

Object-Oriented Development =
Object-Oriented Analysis (OOA)
+
Object-Oriented Design (OOD)
+
Object-Oriented Programming (OOP)

Figure 1-3 Object-oriented development

Understanding OO Development

- OO Analysis and Design
  - Unified Modeling Language (UML)
    - Standard OOA&D modeling notation
    - Defined by: Grady Booch, James Rumbaugh & Ivar Jacobson
    - Uses model-driven approach:
      - Enables creation of graphical models of the system requirements and system design

Understanding OO Development

- OO Analysis and Design
  - Unified Modeling Language (UML)
    - Components
      - Class diagrams
      - Use Case diagrams
      - Sequence diagrams
      - Statecharts

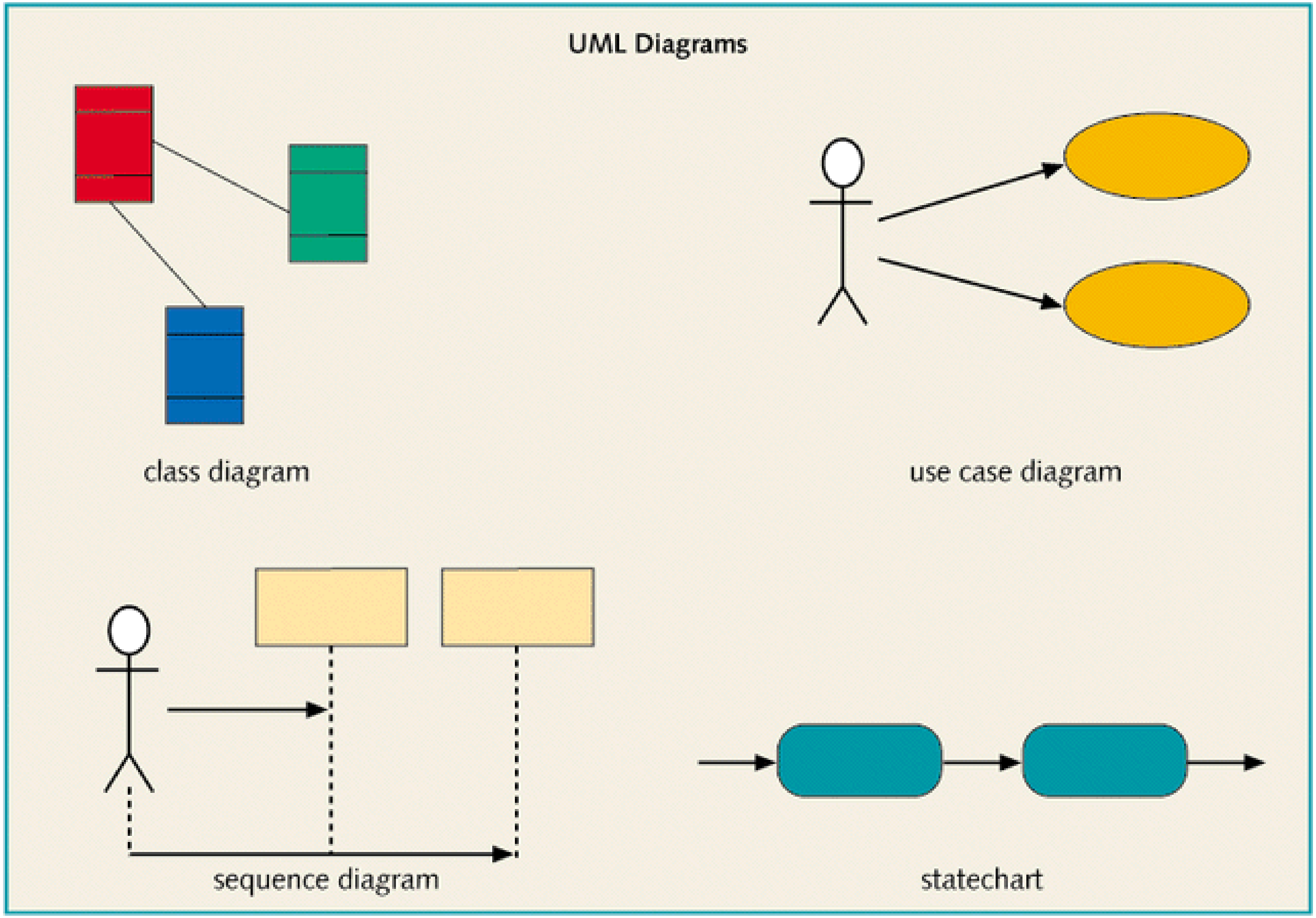


Figure 1-4 Class diagrams, use case diagrams, sequence diagrams, and statecharts

Understanding OO Development

- OO Analysis and Design
  - System Development Life Cycle (SDLC)
    - Project management framework that defines project phases and activities
    - Phases:
      - Planning
      - Analysis
      - Design
      - Implementation
      - Support

Understanding OO Development

- OO Analysis and Design
  - Prototyping
    - Creating a working model of one or more parts of the system for user evaluation and feedback
  - Joint Application Development (JAD)
    - Key system stakeholders and decision makers work together to rapidly define system requirements and designs

Understanding OO Development

- OO Analysis and Design
  - Other requirements:
    - Project management
    - Interviewing
    - Data collection
    - User interface (UI) design
    - Testing
    - Conversion techniques

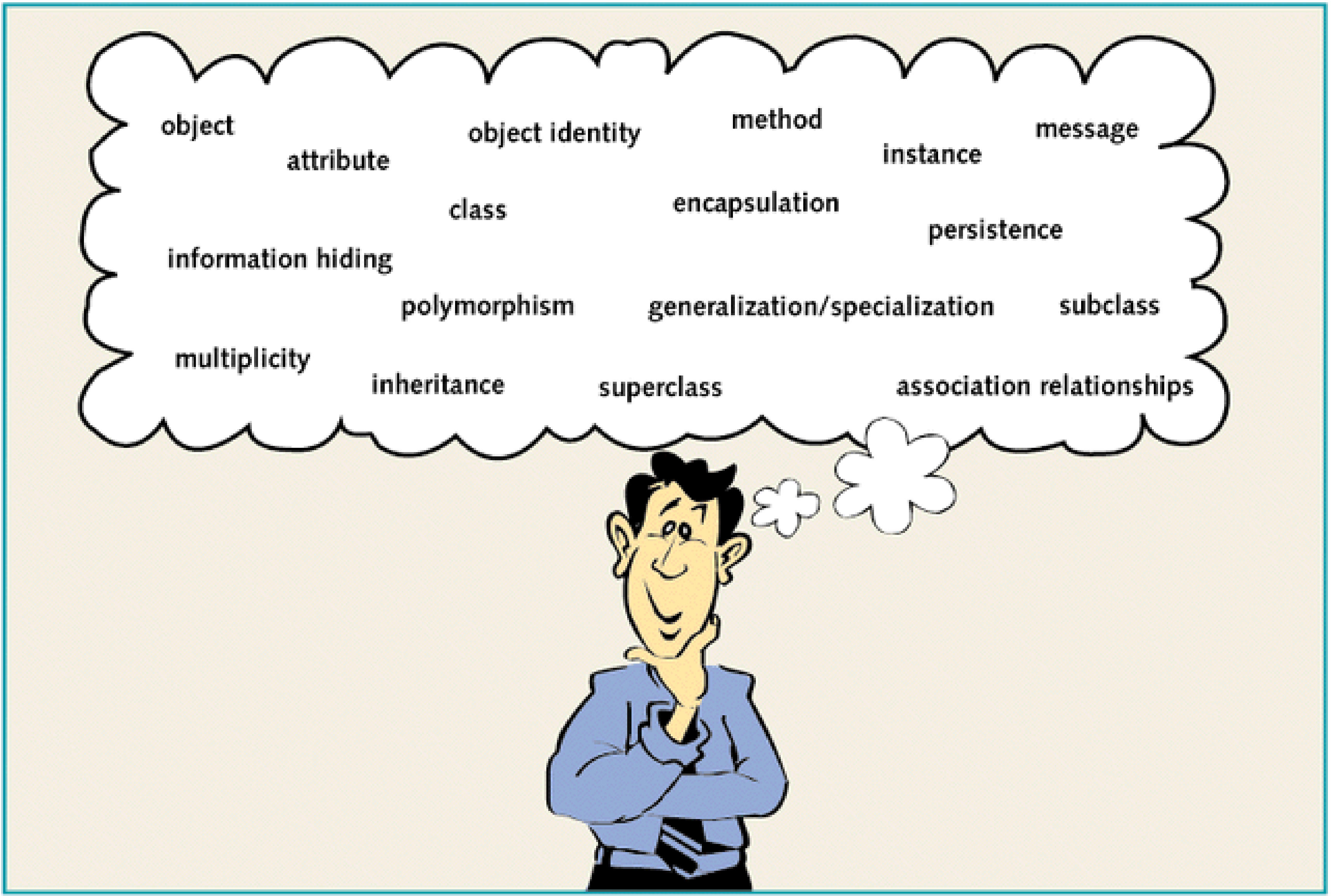
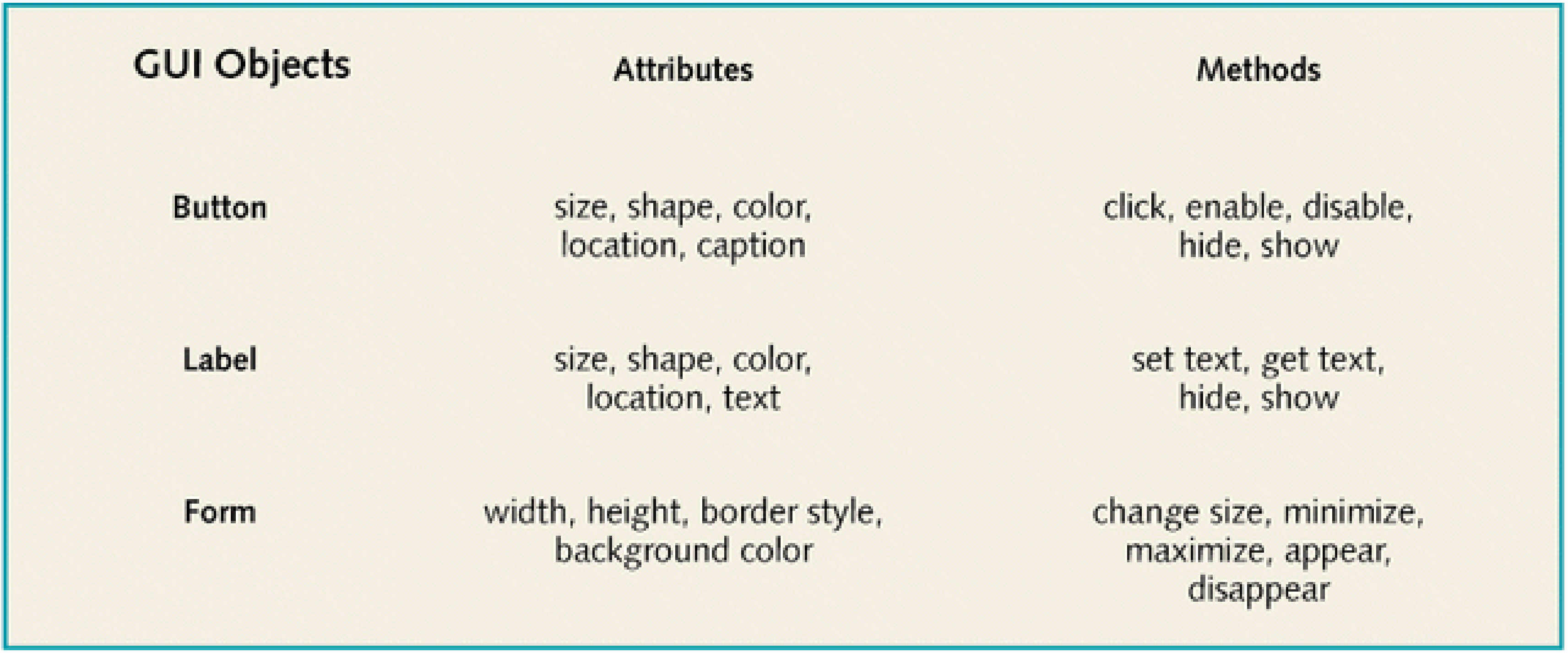


Figure 1-5 Key OO concepts

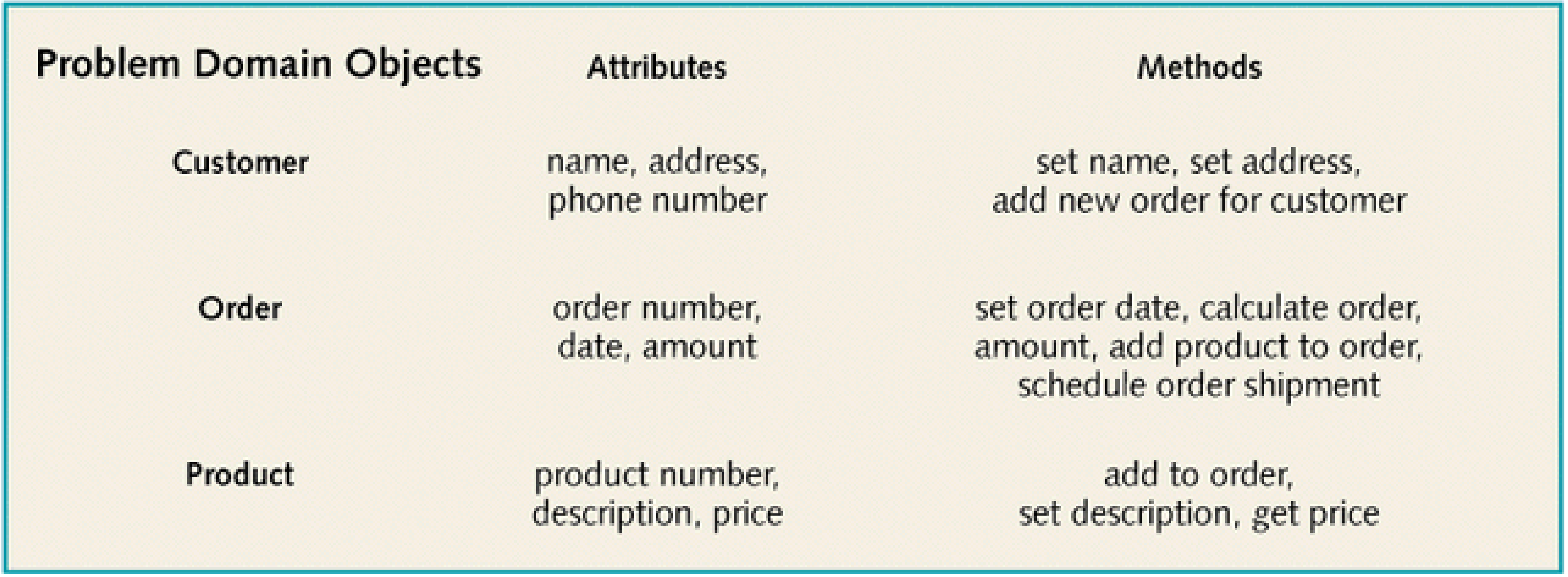
Understanding OO Concepts

- Objects, Attributes, and Methods
  - Object:
    - Attributes
      - Characteristics of an object that have values
    - Behaviors (or methods)
      - Describe what an object can do
    - Examples:
      - GUI objects
      - Problem Domain objects



GUI Objects	Attributes	Methods
Button	size, shape, color, location, caption	click, enable, disable, hide, show
Label	size, shape, color, location, text	set text, get text, hide, show
Form	width, height, border style, background color	change size, minimize, maximize, appear, disappear

Figure 1-6 Attributes and methods of GUI objects



Problem Domain Objects	Attributes	Methods
Customer	name, address, phone number	set name, set address, add new order for customer
Order	order number, date, amount	set order date, calculate order, amount, add product to order, schedule order shipment
Product	product number, description, price	add to order, set description, get price

Figure 1-7 Attributes and methods in problem domain objects

Understanding OO Concepts

- Object Interactions and Messages
  - Messages
    - The means by which objects interact
    - Example:
      - User initiates interaction via messages to GUI objects
      - GUI objects interact with problem domain objects via messages
      - Problem domain objects interact with each other and GUI objects via messages
      - GUI objects respond to user via messages

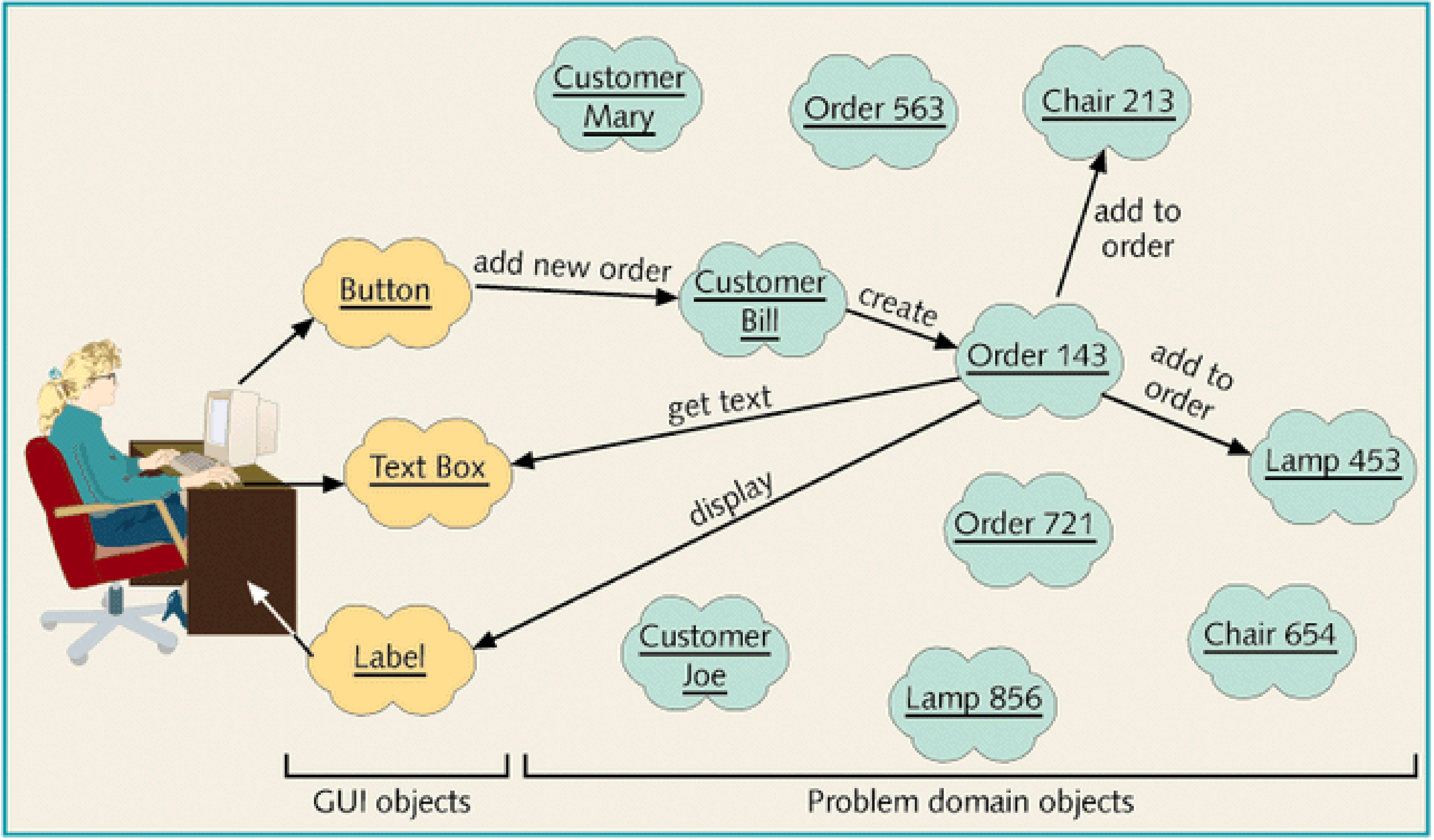


Figure 1-8 Order-processing system where objects interact by sending messages

Understanding OO Concepts

- Encapsulation and Information Hiding
  - Encapsulation
    - Objects have attributes and methods combined into one unit
  - Information Hiding
    - Hiding the internal structure of objects, protecting them from corruption

Understanding OO Concepts

- Encapsulation and Information Hiding
  - Identity
    - Unique reference for each object
  - Persistent objects
    - Defined as available for use over time

Understanding OO Concepts

- Classes, Instances, and Associations
  - Class
    - Defines what all objects of the class represent
  - Instances
    - Objects of the class

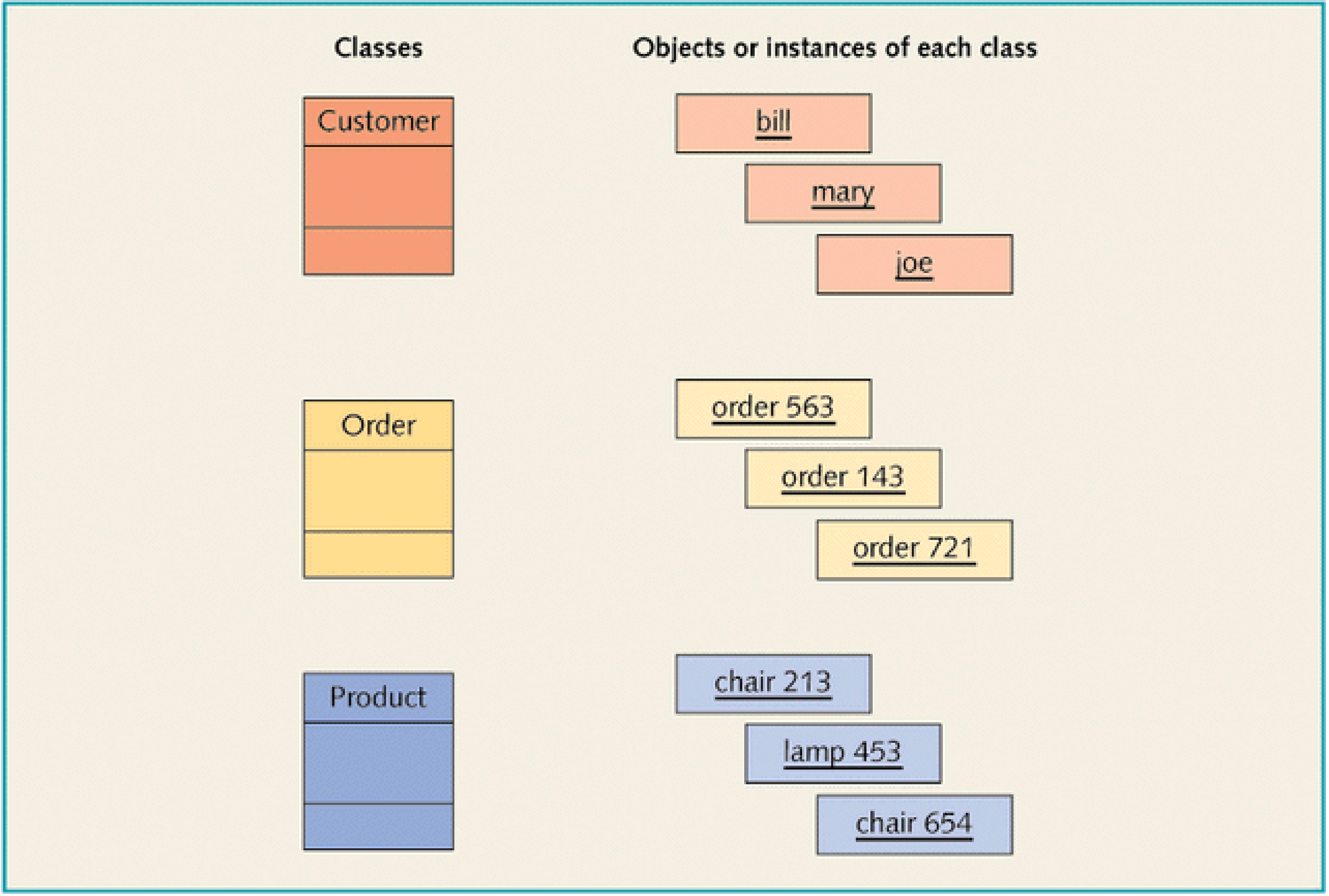


Figure 1-9 Class versus objects or instances of the class

Understanding OO Concepts

- Classes, Instances, and Associations
  - Association relationships
    - Each object is responsible for maintaining relationships with other objects
      - One-to-one
      - One-to-many
    - Multiplicity of the association
      - Number of associations in UML terminology
    - Cardinality of the association
      - Number of associations in ERD terminology

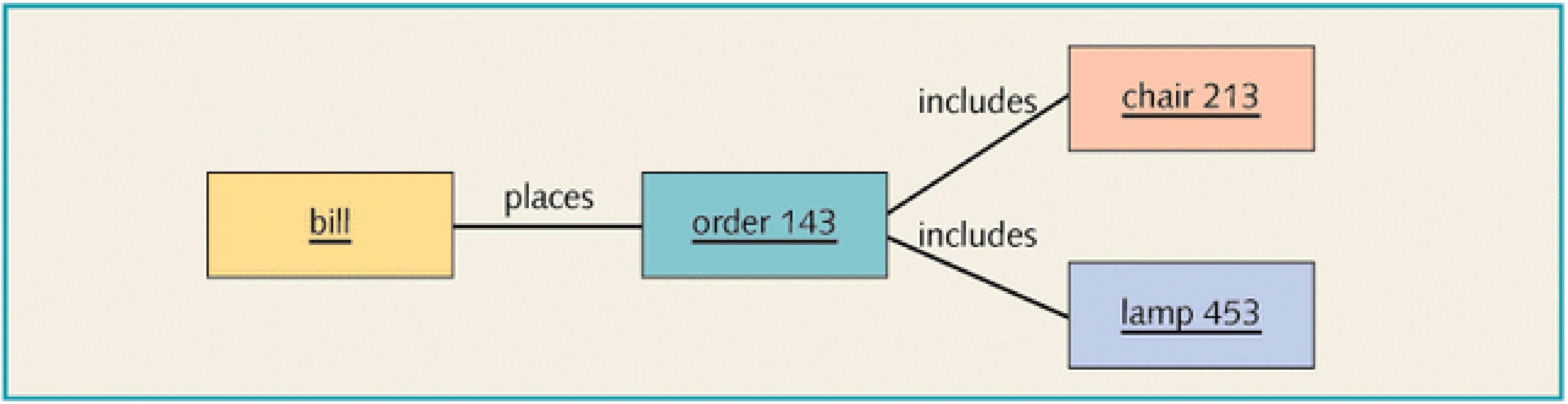


Figure 1-10 Associating objects with other objects

Understanding OO Concepts

- Inheritance and Polymorphism
  - Inheritance
    - One class of objects takes on characteristics of another class and extends them
      - Superclass → subclass
      - Generalization/specialization hierarchy
        - » Also called an inheritance hierarchy
        - » Result of extending class into more specific subclasses

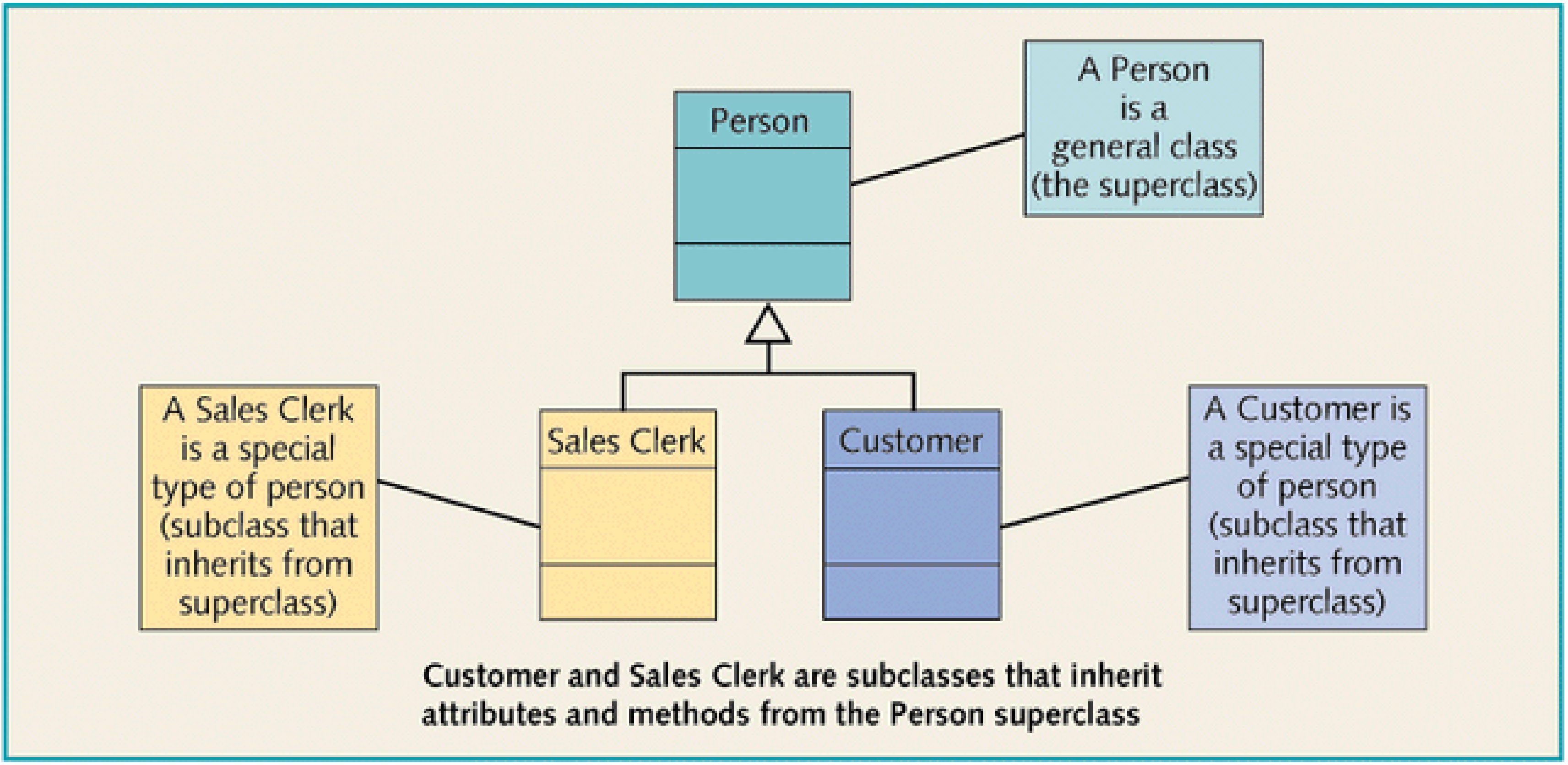


Figure 1-11 Superclass and subclass

Understanding OO Concepts

- Inheritance and Polymorphism
  - Polymorphism
    - “many forms”
    - Different objects can respond in their own way to the same message

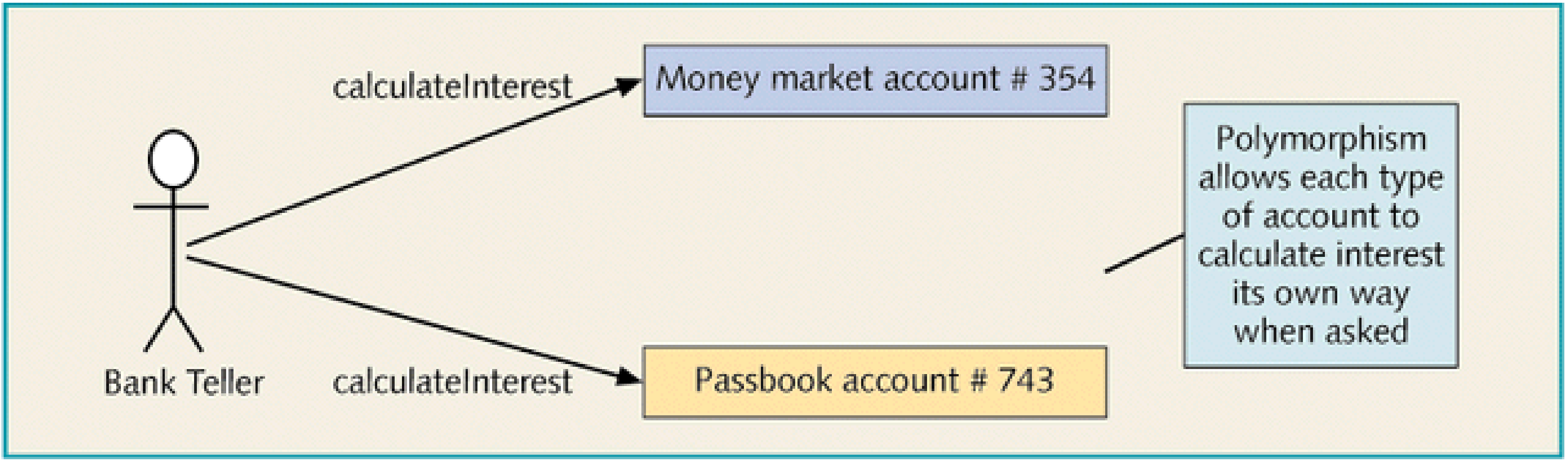

calculateInterest
Money market account # 354
Polymorphism allows each type of account to calculate interest its own way when asked
Bank Teller
calculateInterest
Passbook account # 743

Figure 1-12 Polymorphism for two different types of bank accounts

Recognizing the Benefits of OO Development

- Objects are More Natural
  - Naturalness
    - Based on the fact that people usually think about their world in terms of objects
    - Natural to define the classes of objects involved
    - OO vs. procedural
      - Which is harder to learn?

Recognizing the Benefits of OO Development

- Classes of Objects can be Reused
  - Reuse
    - Classes and objects can be invented once and used many times
      - During analysis, design, and programming
    - Do not need source code for reused class, simply need to know interface

Learning OO Development

- Introducing Three-Tier Design
  - Objects that interact in OO system are separated into three categories of classes:
    - Problem domain classes
      - Specific to a particular business application
    - GUI classes
      - Define objects that make up the UI to the application
    - Data access classes
      - Work with DBMS to store/retrieve object information

END