Overview of Systems Analysis

Key Ideas

- Many systems failures were because analysts tried to build wonderful systems without understanding the organization and applying a systematic approach.
- The primarily goal is to create value for the organization.

Key Ideas

- The systems analyst is a key person analyzing the business, identifying opportunities for improvement, and designing information systems to implement these ideas.
- It is important to understand and develop through practice the skills needed to successfully design and implement new information systems.

Types of Information Systems

- Transaction Processing Systems
- Management Information Systems
- Decision Support Systems
- Expert Systems

Development of these systems require a good understanding of business processes and data

Systems Development Lifecycle Approach

- The project
  - Moves systematically through phases where each phase has a standard set of outputs
  - Produces project deliverables
  - Uses deliverables in implementation
  - Uses *gradual refinement and an iterative* approach

Project Phases

- **Planning**
  - Why build the system?
- **Analysis**
  - What will the system be?
- **Design**
  - How will the system work?
- **Implementation**
  - System delivery

Planning

- Identifying business value
- Analyze feasibility
- Develop work plan
- Staff the project
- Control and direct project

Analysis

- Analysis
- Information gathering
- Process modeling
- Data modeling

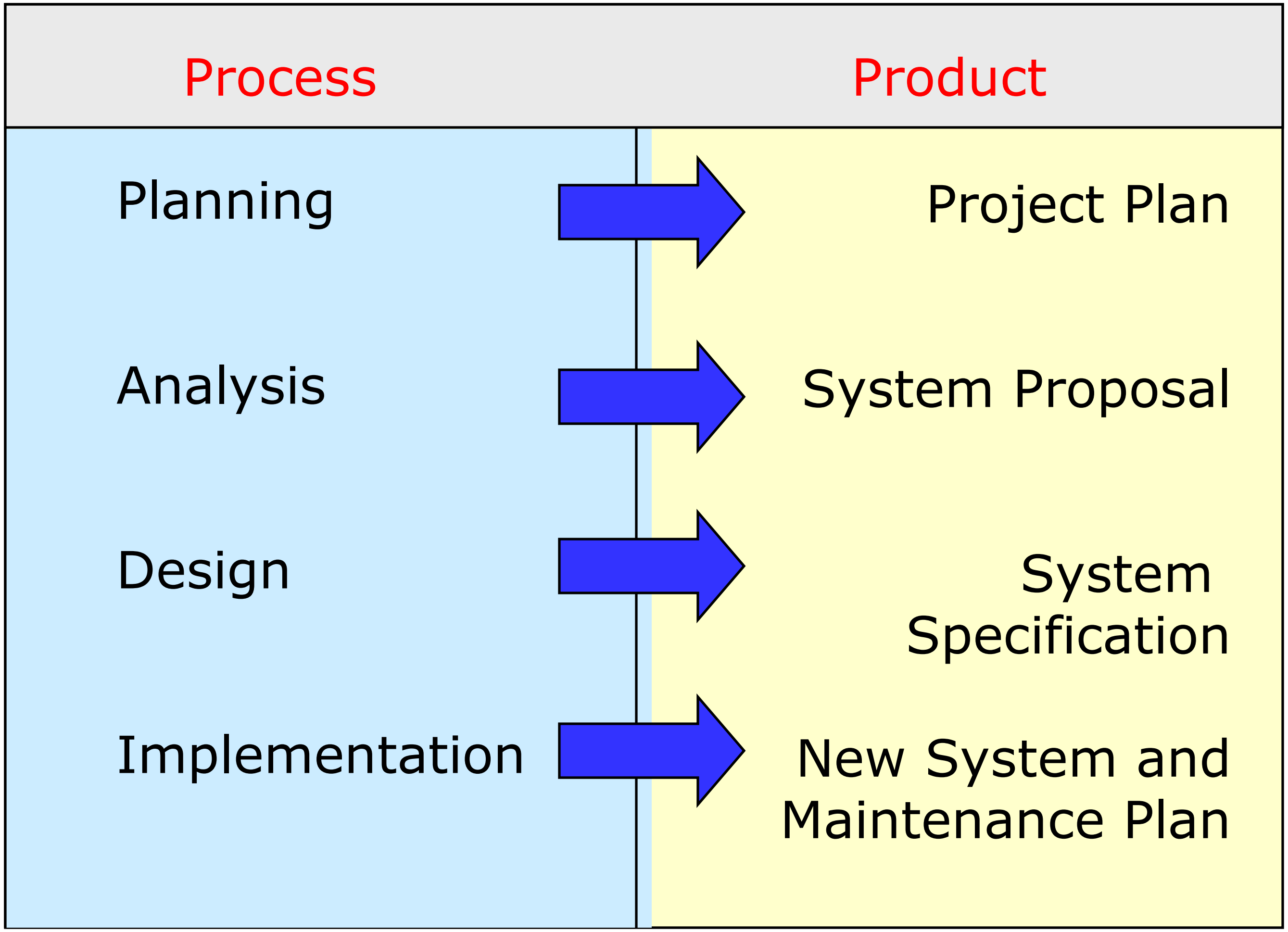
Design

- Physical design
- Architectural design
- Interface design
- Database and file design
- Program design

Implementation

- Construction
- Verification and testing
- Installation

Processes and Deliverables



Process	Product
Planning	Project Plan
Analysis	System Proposal
Design	System Specification
Implementation	New System and Maintenance Plan

SYSTEM DEVELOPMENT METHODOLOGIES

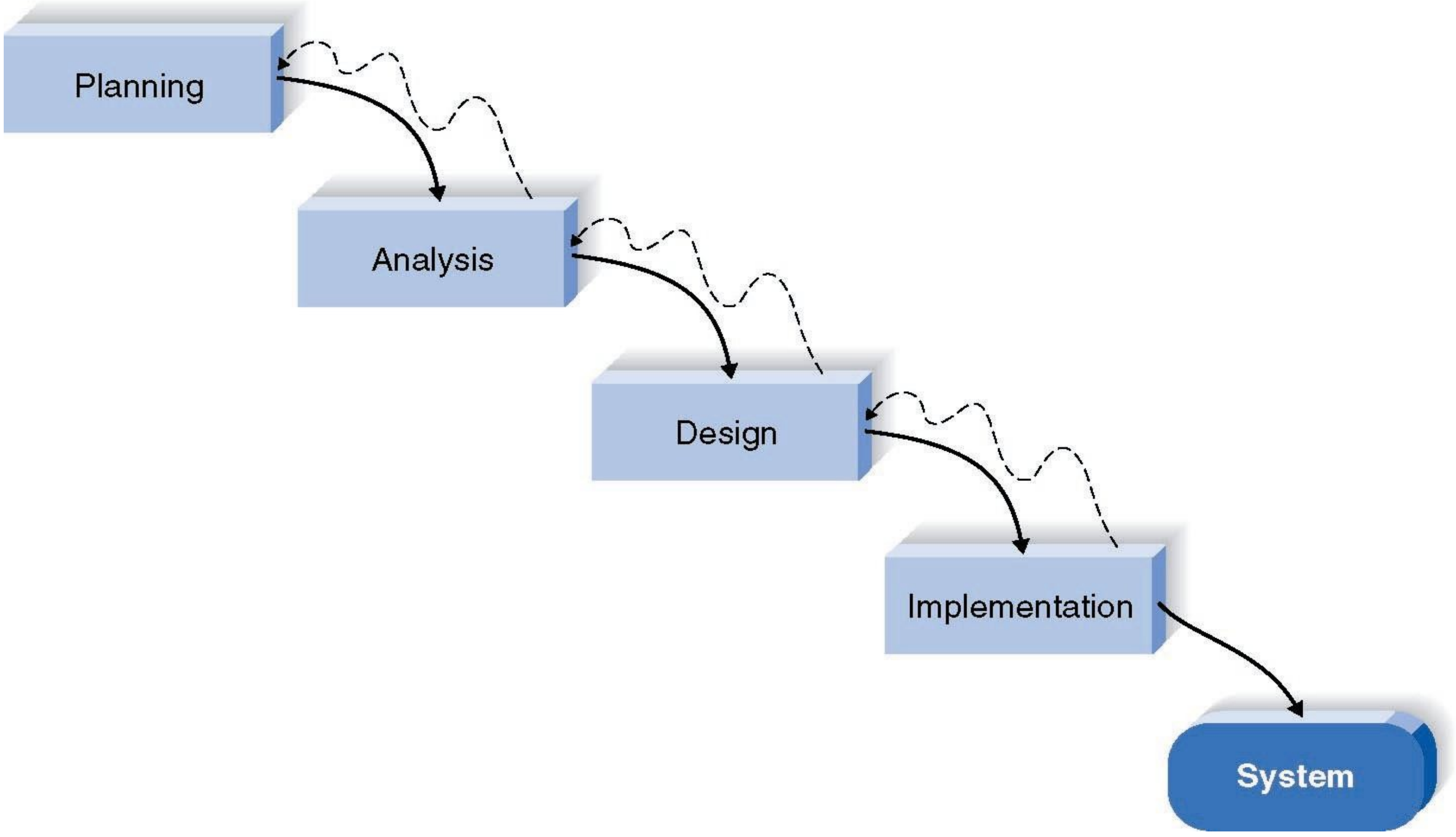
What Is a Methodology?

- A formalized approach or series of steps

Structured Design

- *Projects move methodically from one to the next step*
- *Generally, a step is finished before the next one begins*

Waterfall Development Method



Pros and Cons of the Waterfall Method

Pros	Cons
Identifies systems requirements long before programming begins	Design must be specified on paper before programming begins Long time between system proposal and delivery of new system

Parallel Development

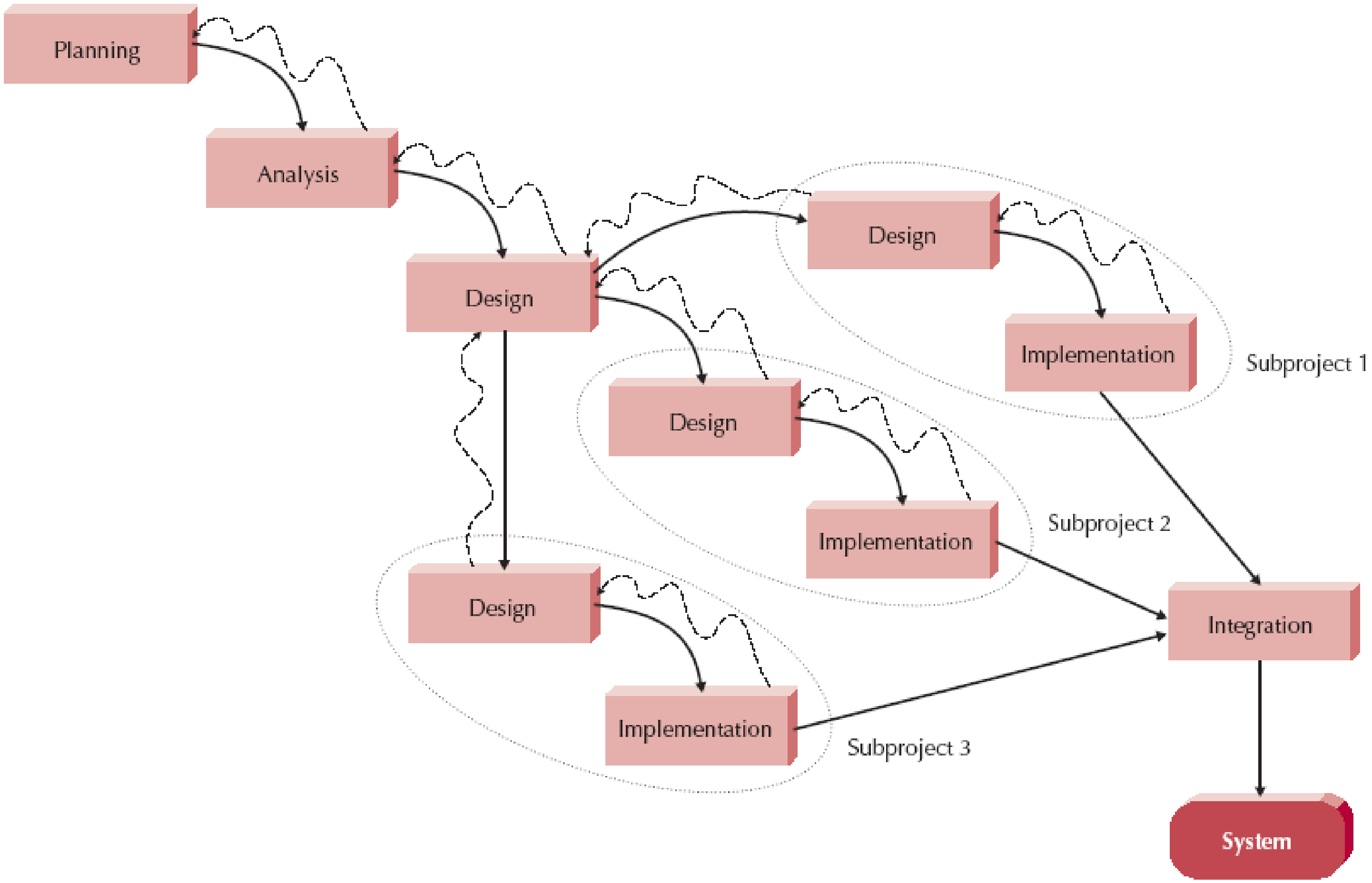


FIGURE 1-3 A Parallel Development-based Methodology

Alternatives to the SDLC

- Rapid Application Development (RAD)
- Prototyping

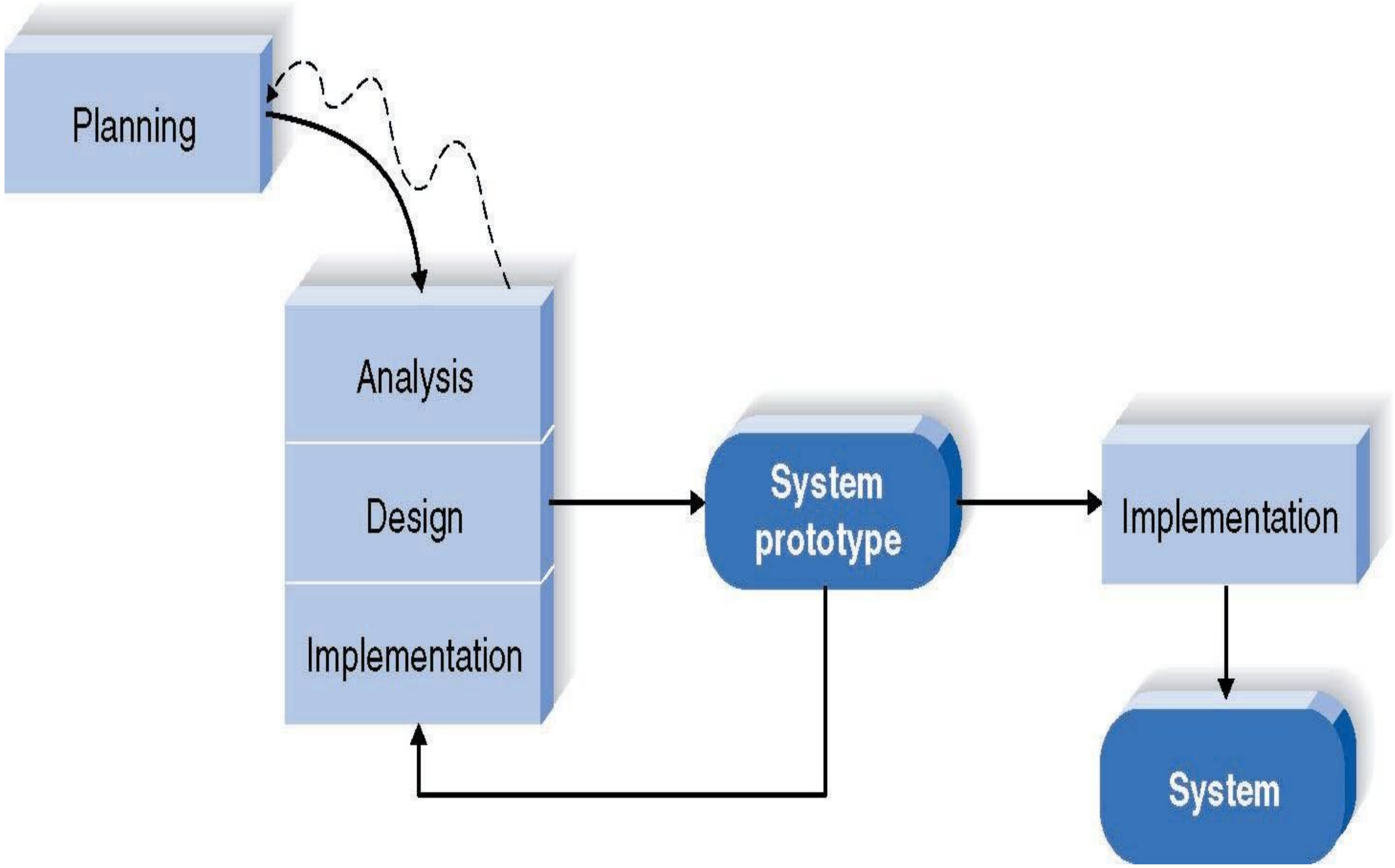
Rapid Application Development

- Critical elements
  - CASE (computer-aided software engineering) tools
  - JAD (join development design) sessions
  - Fourth generation/visualization programming languages
  - Code generators

Rapid Application Development Categories

- Phased development
  - A series of versions
- Prototyping
  - System prototyping
- Throw-away prototyping
  - Design prototyping
- Agile Development
  - Extreme Development

How Prototyping Works



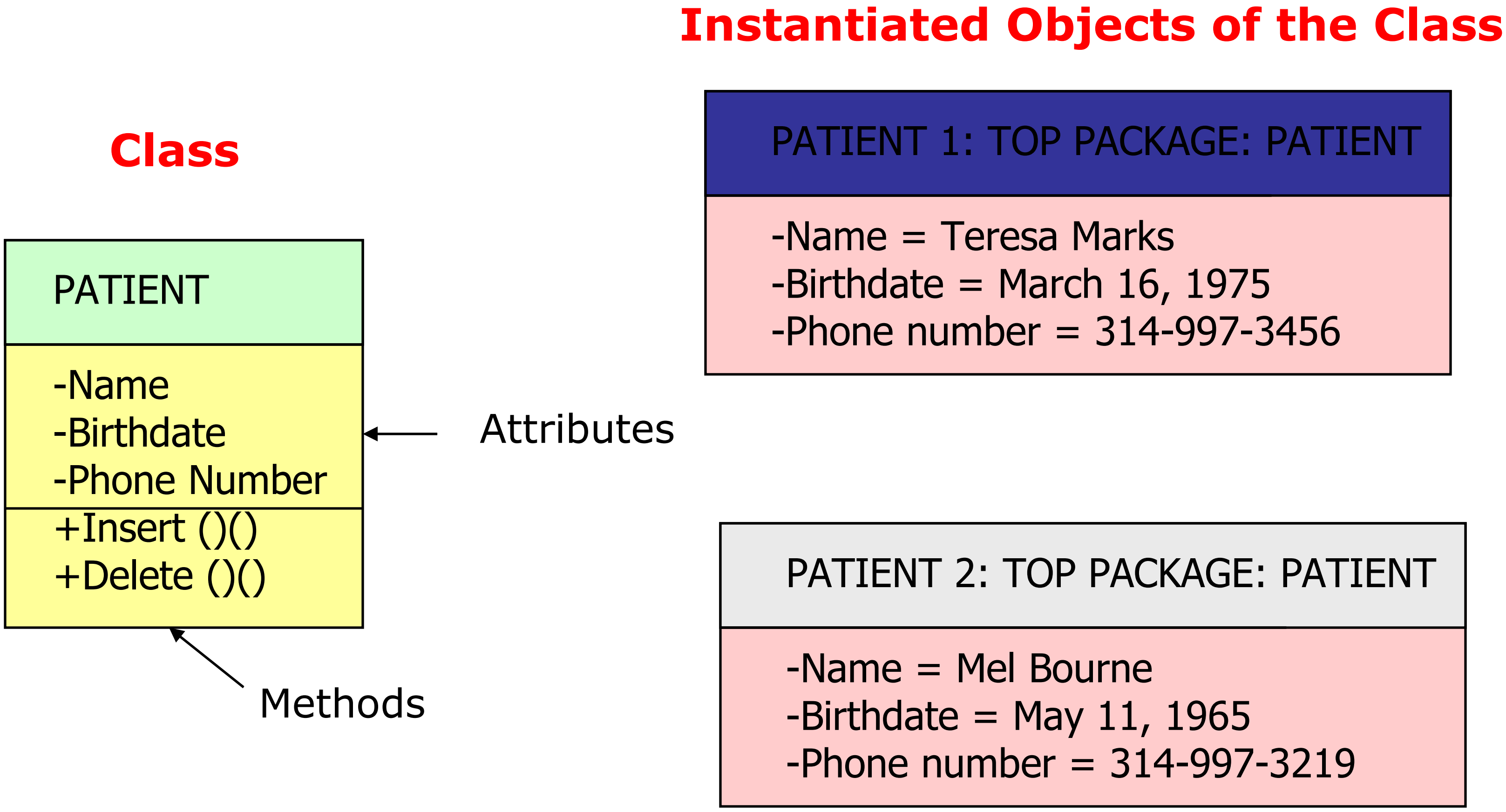
Object-Oriented Analysis and Design

- Attempts to balance emphasis on data and process
- Uses Unified Modeling Language (UML) for diagramming
  - Use-case Driven
  - Architecture Centric
    - Functional, Static, and Dynamic views
  - Iterative and Incremental

Key Elements

- *Classes* -- template to define objects
- *Instances* -- specific examples of class members
- *Objects* -- building block of the system
- *Attributes* -- describe data aspects of the object
- *Methods* -- the processes the object can perform
- *Messages* -- instructions sent to or received from other objects

A Class and Its Objects



Class

PATIENT

-Name
-Birthdate
-Phone Number

+Insert ()()
+Delete ()()

Attributes

Methods

Instantiated Objects of the Class

PATIENT 1: TOP PACKAGE: PATIENT

-Name = Teresa Marks
-Birthdate = March 16, 1975
-Phone number = 314-997-3456

PATIENT 2: TOP PACKAGE: PATIENT

-Name = Mel Bourne
-Birthdate = May 11, 1965
-Phone number = 314-997-3219

The Key to Reusability

- *Information hiding* is the principle that only information required to use the object is available outside the object
- *Encapsulation* is the mechanism that combines data and processes in a single object

UML

- Unified Modeling Language
- The full UML provides 9 separate diagramming techniques