

## Reading 3: Chapter 11 –

## Requirements Modeling: Classes and Patterns

(pp. 233–252)

 **Goal: Move from Behavior & Structure to Design-Oriented Thinking**

### **Main Models**

#### 1. **Class-Based Modeling**

- *Class*: An abstraction that describes attributes and operations.
- *Responsibility*: What the class knows and does.
- Includes associations (has-a), aggregations (part-of), and inheritance (is-a).

#### 2. **CRC Models**

- *Class-Responsibility-Collaborator* cards help brainstorm design structure.
- Useful in agile teams or workshops.

#### 3. **Pattern-Based Modeling**

- Use **analysis patterns** (e.g., Observer, MVC) to solve recurring design challenges
- Encourages reuse and best practices in early-stage modeling.

#### 4. **Packages and Collaborations**

- Large systems are broken into logical **packages**.
- Packages describe how groups of classes work together.
- Supports **modularity** and scalability.

### **Design Transition**

- These models act as a **bridge to object-oriented design (OOD)**.
- Help developers transition from abstract user requirements to implementable components.