

# UML Diagrams

CSE 403

# Think of it as ...

- A more concise and accurate documentation of software architecture than English
- A way to guide discussion to reveal assumptions and miscommunications
- A compact and quick way to take notes in meetings

# Think of it as ... (Cont.)

- A way to verify that a system is capable of meeting its requirements
- An intuitive way to identify a system's weaknesses
- A way to quickly onboard new team members
- A way to remind yourself of decisions you made months ago

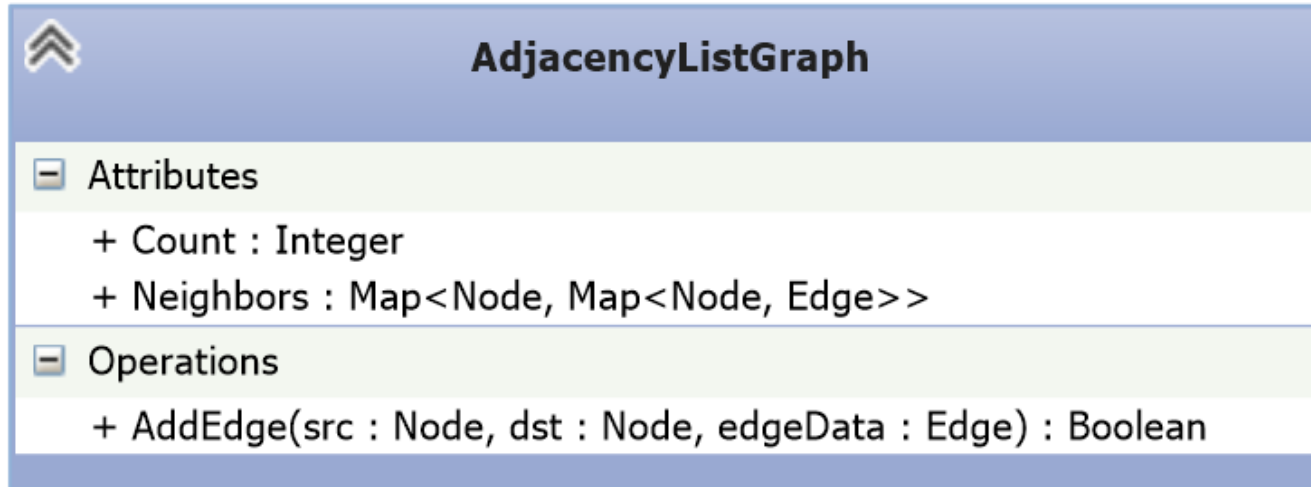
# Class Diagrams

# A Brief Review

Describes

- The system's classes
- Their attributes
- Their operations
- Their relationships with each other

# Representing a class



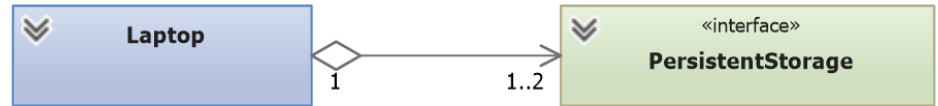
# General Association

A two-way connection  
between peers



# Aggregation

One or more classes are generally treated as members of some larger whole, but may belong to multiple objects during their lifetimes





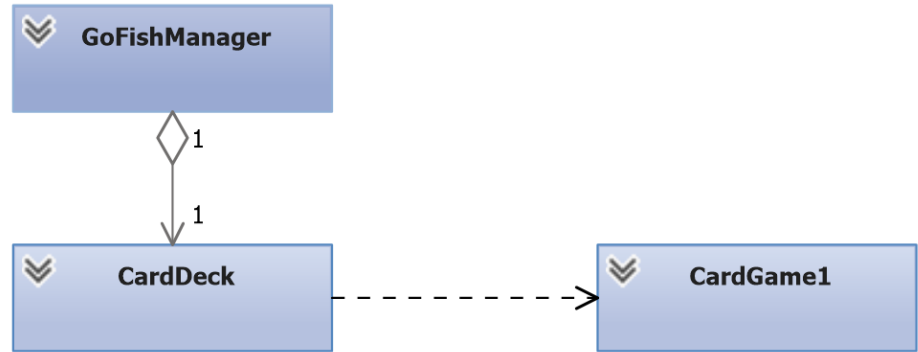
# Composition

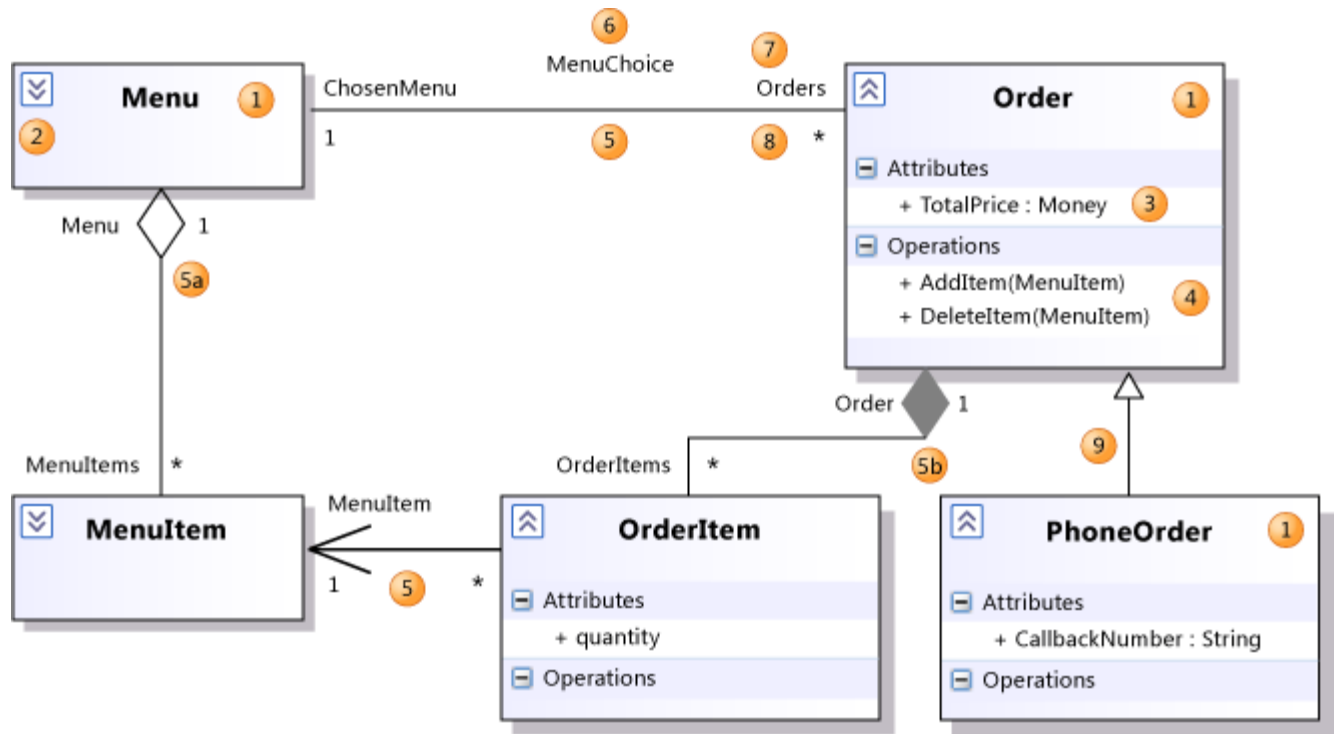
One or more classes are always treated as members of some larger whole, and are destroyed with the whole



# Dependency

Changing the interface of one class may change the implementation or interface of another

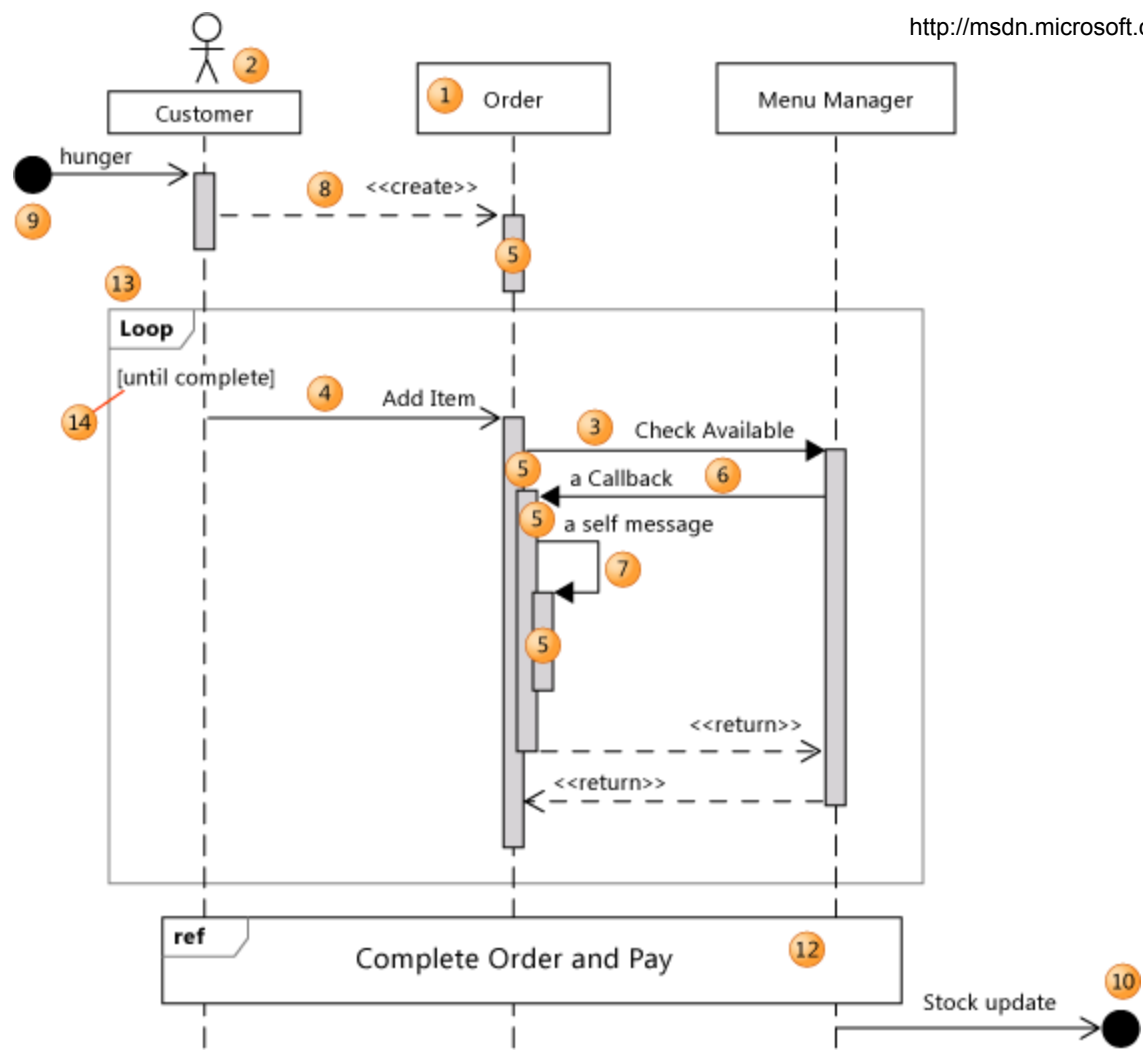




# Sequence Diagrams

# A Brief Review

- Describes the sequence of messages passed between instances of components over time
- Time flows down the diagram
- Shows how control flow moves from one component to another



# Design Exercise

Design the models (not any views or controllers) for course registration software based on the UW's system. Create:

- A sequence diagram for the use case “user attempts to register for one or more courses by entering their SLNs”.
- A class diagram.

The system should let a student register if and only if they have taken the prerequisite courses and have no scheduling conflicts.