Introduction to OOD & UML

Lecture-1

Object Oriented Design (OOD)

Designing systems using self-contained objects and object classes

Charachteristics of OOD

- Objects are abstractions of real-world or system entities and manage themselves
- Objects are independent and encapsulate state and representation information.
- System functionality is expressed in terms of object services
- Shared data areas are eliminated. Objects communicate by message passing
- Objects may be distributed and may execute sequentially or in parallel

Advantages of OOD

- Easier maintenance. Objects may be understood as stand-alone entities
- Objects are appropriate reusable components
- For some systems, there may be an obvious mapping from real world entities to system objects

Object Oriented Development

- Object-oriented analysis, design and programming are related but distinct
- OOA is concerned with developing an object model of the application domain
- OOD is concerned with developing an object- oriented system model to implement requirements
- OOP is concerned with realising an OOD using an OO programming language such as Java, C++, C# etc

Object & Object Classes

- Objects are entities in a software system which represent instances of real-world and system entities
- Object classes are templates for objects. They may be used to create objects
- Object classes may inherit attributes and services from other object classes

Generalization & Inheritance

- Objects are members of classes which define attribute types and operations
- Classes may be arranged in a class hierarchy
  where one class (a super-class) is a generalisation of one or more other classes (sub-classes)
- A sub-class inherits the attributes and operations from its super class and may add new methods or attributes of its own
- Specialization in the UML is implemented as inheritance in OO programming languages

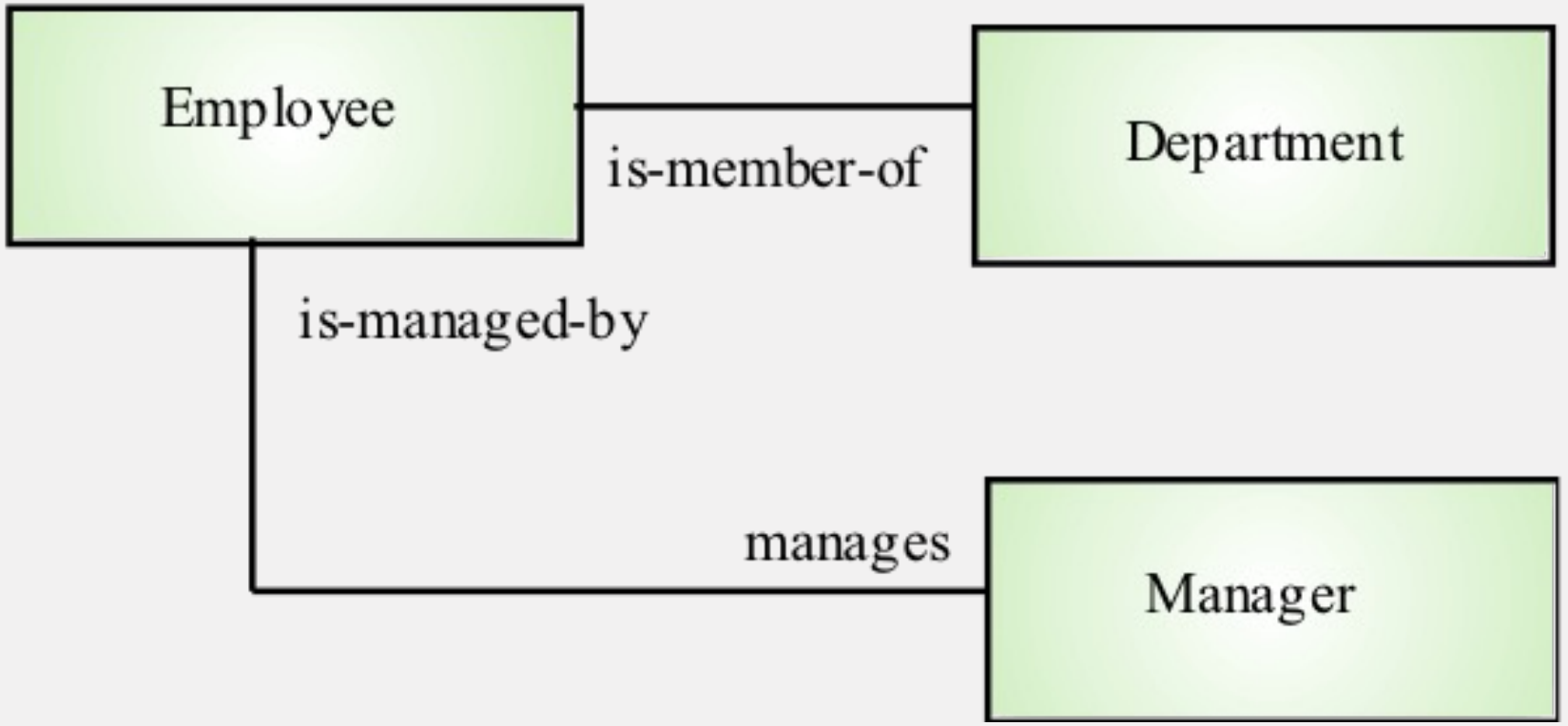
The Unified Modeling Language

- Several different notations for describing object- oriented designs were proposed in the 1980s and 1990s
- The Unified Modeling Language is an integration of these notations
- It describes notations for a number of different models that may be produced during OO analysis and design
- It is now a *de facto* standard for OO modelling

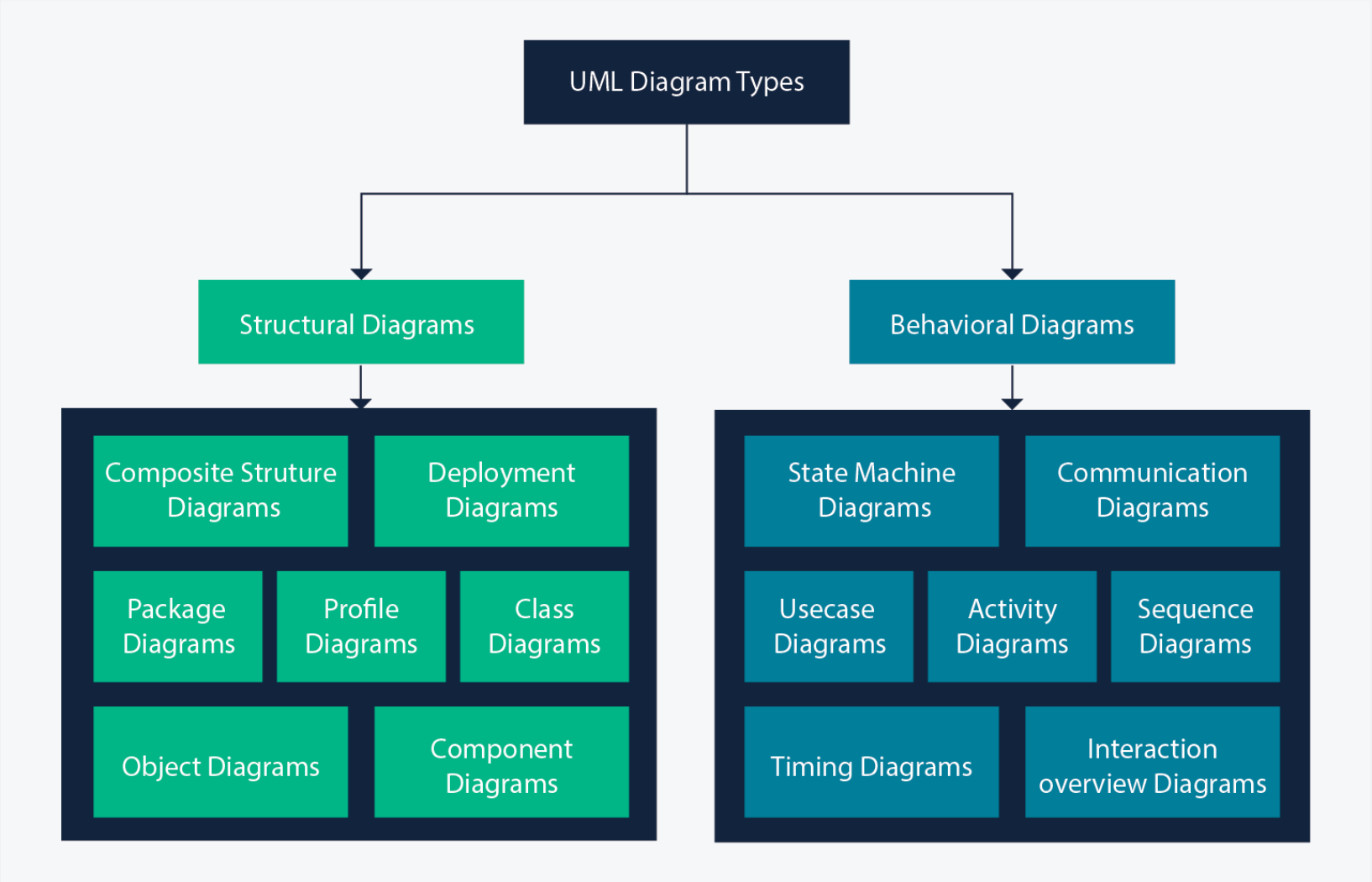
UML Associations

- Objects and object classes participate in relationships with other objects and object classes
- In the UML, a generalised relationship is indicated by an association
- Associations may be annotated with information that describes the association
- Associations are general but may indicate that an attribute of an object is an associated object or that a method relies on an associated object

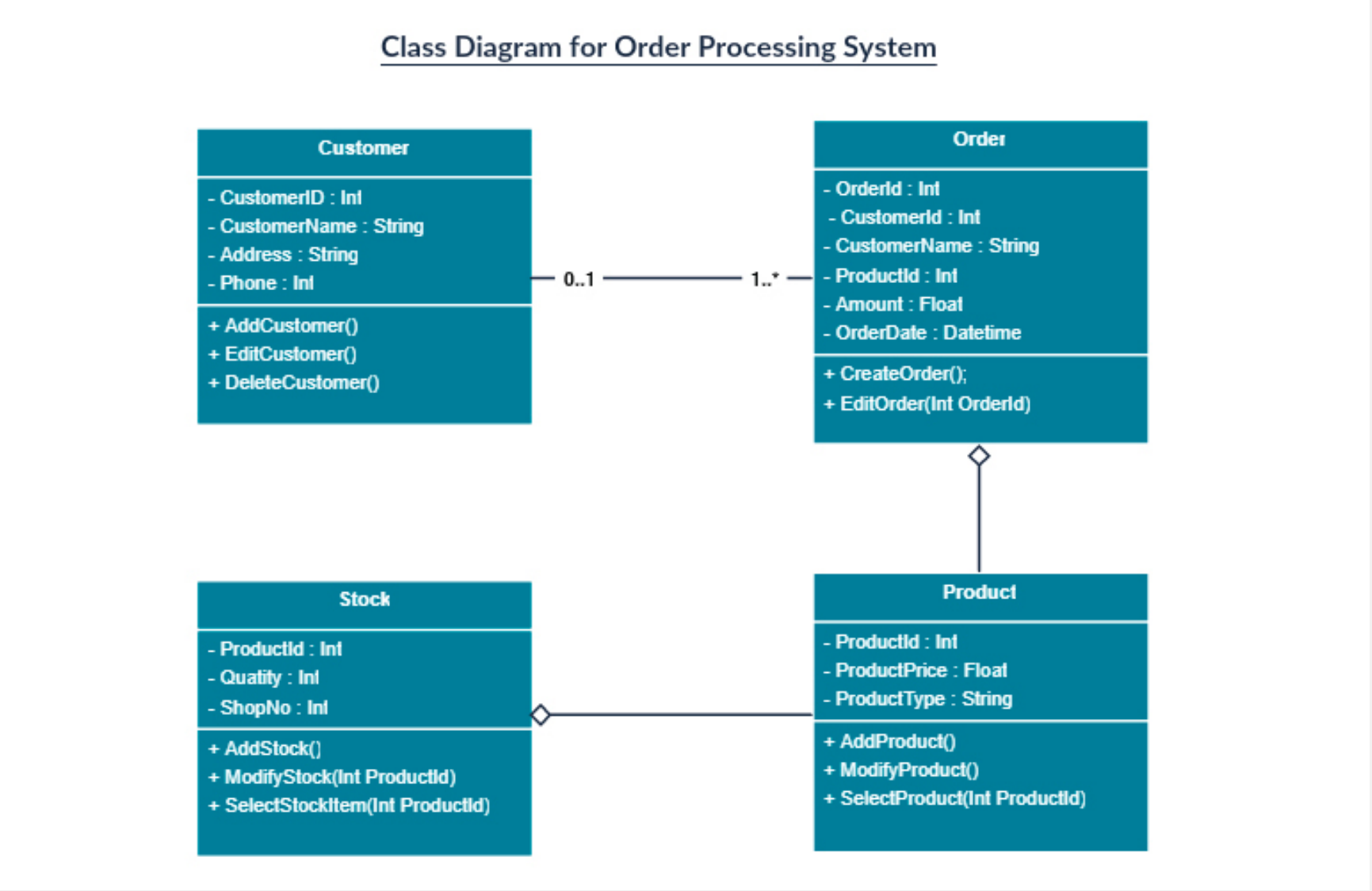
Association Model



UML Diagram Types



Class Diagram



UML Relation Notations