Object Relationships
UML Class diagrams

Software Requirements and Design
CITS 4401
Lecture 4

UML Class Diagrams
- Describe the static structure of the system
  - classes,
  - class attributes,
  - associations between classes,
  - association roles and multiplicity

classes: features and facts about the problem domain which matter in the system we are building to support it

UML First Pass: Class Diagrams

Class diagrams represent the structure of the system.

Class Diagrams
- Class diagrams represent the structure of the system.
- Class diagrams are used
  - during requirements analysis to model problem domain concepts
  - during system design to model subsystems and interfaces
  - during object design to model classes.
Classes

- A **class** represents a concept.
- A class encapsulates state (attributes) and behavior (operations).
- Each attribute has a **type**.
- Each operation has a **signature**.
- The class name is the only mandatory information.

### Attributes
- `TariffSchedule`
- `zone2price`
- `getZones()`
- `getPrice()`

### Methods
- `TariffSchedule`
- `Table zone2price`
- `Enumeration getZones()`
- `Price getPrice(Zone)`

**Exercise: Class Diagrams**

- A **book** is composed of a number of **parts**
- Each **part** is composed of a number of **chapters**
- **Chapters** are composed of **sections**
- Draw a class diagram representing a book. Focus only on classes and relationships

**Instances**

- An **instance** represents a phenomenon.
- The name of an instance is **underlined** and can contain the class of the instance.
- The attributes are represented with their **values**.

```
tariff_1974: TariffSchedule
zone2price = {
  {'1', .20},
  {'2', .40},
  {'3', .60}}
```

**Associations**

- Associations denote relationships between classes
- **A is associated with B** is: **A has to know about B**
- The multiplicity of an association end denotes how many objects the source object can legitimately reference.
### 1-to-1 and 1-to-Many Associations

- **Country**
  - name: String
  - Has-capital 1
- **City**
  - name: String
  - Capital 1

1-to-1 association

- **Polygon**
  - draw()
  - 1
- **Point**
  - x: Integer
  - y: Integer
  - 1..* 1

1-to-many association

### Exercise 2: Class Diagram Attributes

- Extend your book class diagram to include the following attributes:
  - a book includes a publisher, publication date, and an ISBN
  - a part includes a title and a number
  - a chapter includes a title, a number, and an abstract
  - a section includes a title and a number

### Aggregation

- An **aggregation** is a special case of association denoting a "consists of" hierarchy.
- The **aggregate** is the parent class, the **components** are the children class.

### Composition

- A solid diamond denotes **composition**, a strong form of aggregation where components cannot exist without the aggregate.
Aggregation: Shared vs Composition
- "One of the most frequent sources of confusion in the UML is aggregation and composition" [UML Distilled]
- Shared aggregation means that both objects can exist independently whereas composition means that the "part" cannot exist independently
  - Shared (A "uses" B): B exists independently (conceptually) from A
  - Composition (A "owns" B): B has no meaning or purpose in the system without A
- e.g. A Company is a shared aggregation of People. When a Company ceases exist, the People continue to exist
- e.g. A Text Editor owns a Buffer (composition). A Text Editor uses a File (shared). When the Text Editor is closed, the Buffer is destroyed but the File itself is not destroyed.

Generalization
- Generalization relationships denote inheritance between classes.
- The children classes inherit the attributes and operations of the parent class.
- Generalization simplifies the model by eliminating redundancy.

Exercise 3
- Create a class diagram to depict A house with a number of rooms. The house will have a floor area, a garage, a kitchen, and a number of exterior doors and windows. All rooms will have dimensions (length and width). Some rooms will be bedrooms; others will be bathrooms, toilets, or games rooms.