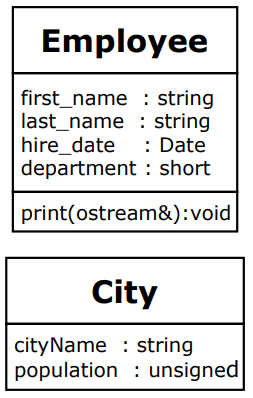
# Unified Modeling Language

Collection of notations representing software designs from three points of view:

* Class model: Describes static structure of objects and relationships in system
  + Comprises object and class diagrams
  + Provides new and useful abstractions for reasoning
* State Model: Describes dynamic aspects of objects and nature of control
* Interaction model: Describes how ojects in a system cooperate to achieve broader results

Class Diagram Notation

* Boxes denote classes
  + Class name
  + Attributes
  + Operations
* Form: visibility name: type multiplicity = default {property-string}
  + Visibility: + or – or # or ~
    - Indicates public, private, protected, or package
  + Name: Name of field or assosciation
  + Type: Kind of object
  + Multiplicity:
    - 0..1 optional
    - 1..\* at least one
    - 0..\* any number
  + Default: Value of default value
  + Property-string: Additional properties using keywords (eg ReadOnly)

Associations:

* Notes a property
* Relationship to other pclasses or objects
* Solid line from source class to target class