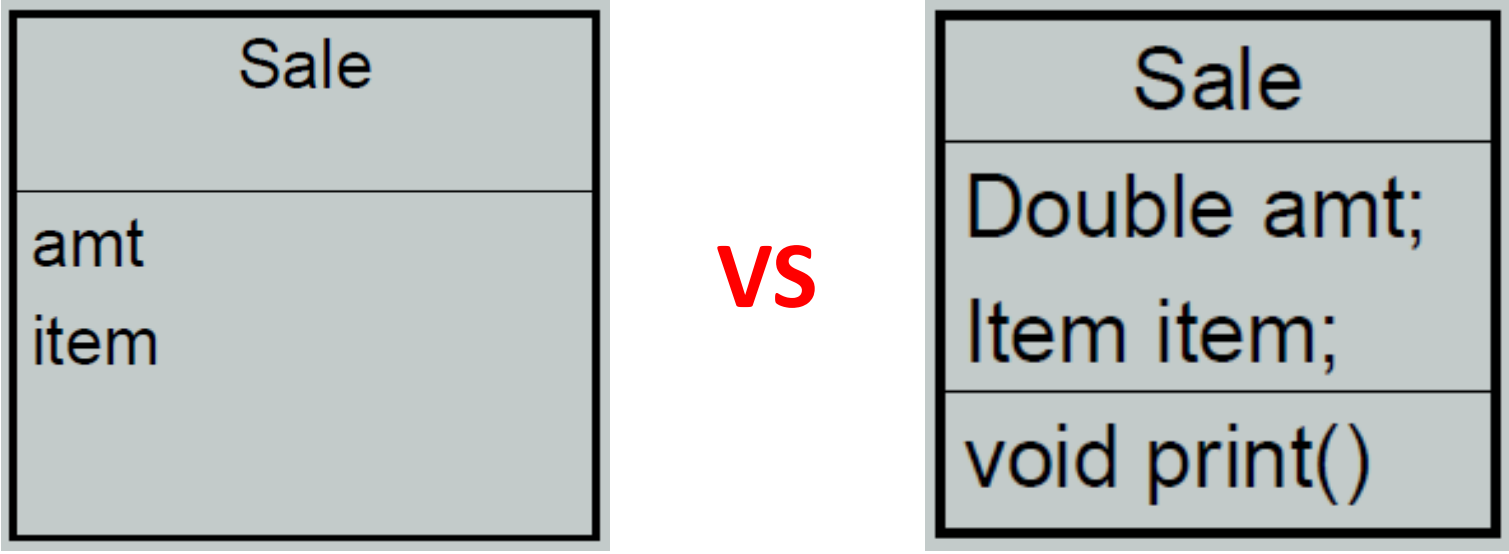
Class Model/Domain Model

Class Diagram

What is a Domain Model?

- Illustrates meaningful conceptual classes in problem domain
- Represents real-world concepts, not software components



Why domain model?

- Gives a conceptual framework of the things in the problem space
- Helps you think – focus on semantics
- It is a static view - meaning it allows us convey time invariant business rules
- Foundation for modelling
- Based on the defined structure, we can describe the state of the problem domain at any time.

Features of a domain model

- **Domain classes** – each domain class denotes a type of object.
- **Attributes** – an attribute is the description of a specified type in a domain class; each instance of the class separately holds a value.
- **Associations** – an association is a relationship between two (or more) domain classes that describes links between their object instances. Associations can have roles, describing the multiplicity and participation of a class in the relationship.

Steps to Create a Domain Model

- Identify candidate conceptual classes
- Draw them in a UML as a Class Diagram
- Add associations necessary to record the relationships that must be retained
- Add attributes necessary for information to be preserved

Domain classes

- Each domain class denotes a type of object. It is a descriptor for a set of things that share common features. Classes can be:-
  - *External entities*
  - *Things*
  - *Occurrences or events*
  - *Roles*
  - *Organizational units*
  - *Places*
  - *Structures*

Identifying Classes

- Identify, classify, and define objects, ideas, and events as **Classes** by examining the usage scenarios developed in the use case model, glossary, and other diagrams.
- Methods: -
  - Reuse an existing domain model
  - Use a conceptual class category list
  - Identify noun phrases

Conceptual Class Category List

- Physical or tangible objects
  - Register, Airplane
- Specifications, or descriptions of things
  - Product Specification, Flight Description
- Places
  - Store, Airport
- Transactions
  - Sale, Payment, Reservation
- Transaction Items
  - Sales Line Items
- Roles
  - Cashier, Pilot
- Containers of other things
  - Store, Airplane, Cinema
- Things in a container
  - Item, Passenger, Audience
- Catalogs
  - Product Catalogs, Part Catalogs
- Organizations
  - Department, Class

Identifying Noun Phrases

- Classes are determined by underlining each noun or noun phrase and entering it into a simple table.
- Synonyms should be noted. Beware of ambiguity.
- If the class (noun) is required to implement a solution, then it is part of the solution space; otherwise, if a class is necessary only to describe a solution, it is part of the problem space.

The SafeHome security function enables the homeowner to configure the security system when it is installed, monitors all sensors connected to the security system, and interacts with the homeowner through the Internet , a PC or a control panel . During installation , the SafeHome PC is used to program and configure the system. Each sensor is assigned a number and type , a master password is programmed for arming and disarming the system, and telephone number(s) are input for dialing when a sensor event occurs. When a sensor event is recognized , the software invokes an audible alarm attached to the system. After a delay time that is specified by the homeowner during system configuration activities, the software dials a telephone number of a monitoring service, provides information about the location , reporting the nature of the event that has been detected. The telephone number will be redialed every 20 seconds until telephone connection is obtained. The homeowner receives security information via a control panel, the PC, or a browser, collectively called an interface . The interface displays prompting messages and system status information on the control panel, the PC, or the browser window. Homeowner interaction takes the following form . . .

Potential Class

Potential Class	General Classification
Homeowner	Role / external entity
Sensor	External entity
Control panel	External entity
Installation	Event / occurrence
System / security system	Thing
Number, type	Not object, attributes of sensor
master password	Thing
Telephone number	Thing
Sensor event	Event / occurrence
Audible alarm	External entity
Monitoring service	Organization unit / external entity

Class selection characteristics

- Use six selection characteristics to select classes for the analysis model: -
  1. Retained information - if information about the class must be remembered so that the system can function
  2. Needed services - the potential class must have a set of identifiable operation that is needed by the system to operate
  3. Multiple attributes – a class with a single attribute may, in fact, be useful during design, but is probably better represented as an attribute of another class during the analysis activity

Class selection characteristics

- Use six selection characteristics to select classes for the analysis model: -
  4. Common attributes - A set of attributes can be defined for the potential class and these attributes apply to all instances of the class.
  5. Common operations - A set of operations can be defined for the potential class and these operations apply to all instances of the class.
  6. Essential requirements - External entities that appear in the problem space and produce or consume information essential to the operation of the system

Potential Class

Potential Class	Selection Characteristics
Homeowner	Failed 1, 2.
Sensor	Passed
Control panel	Passed
Installation	Passed
System / security system	Passed
Number, type	Failed 3. attributes of the sensor
master password	Failed 3. attributes of the system
Telephone number	Failed 3. attributes of the system
Sensor event	Passed
Audible alarm	Passed
Monitoring service	Failed 1, 2

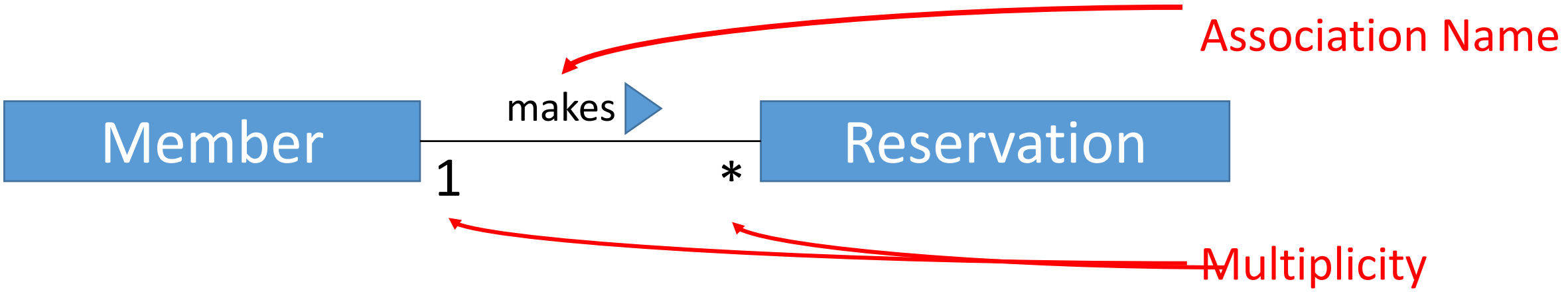
Class Names

- **Class Name** creates the vocabulary of our analysis
  - Use nouns as class names
  - Verbs can also be made into nouns
- Use pronounceable names
  - If you cannot read aloud, it is not a good name
- Use CamelCasing
  - E.g., CardReader rather than CARDREADER or card_reader
- Avoid obscure, ambiguous abbreviations
  - E.g., is TermProcess something that terminates or something that runs on a terminal?
- Try *not* to use digits within a name, such as CardReader2

Associations

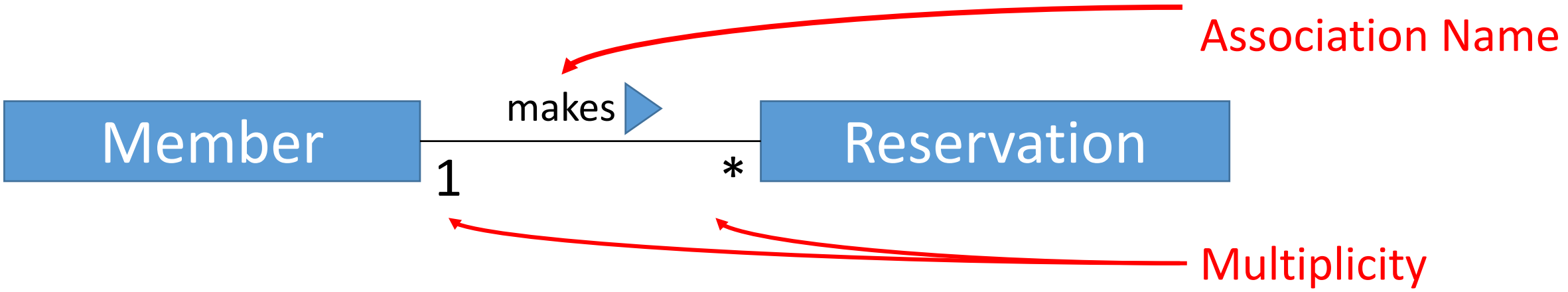
- Association is the relationship between 2 separate classes (“has a”).
- For example:
  - Member class is associated to Reservation.

Association Name and Multiplicity



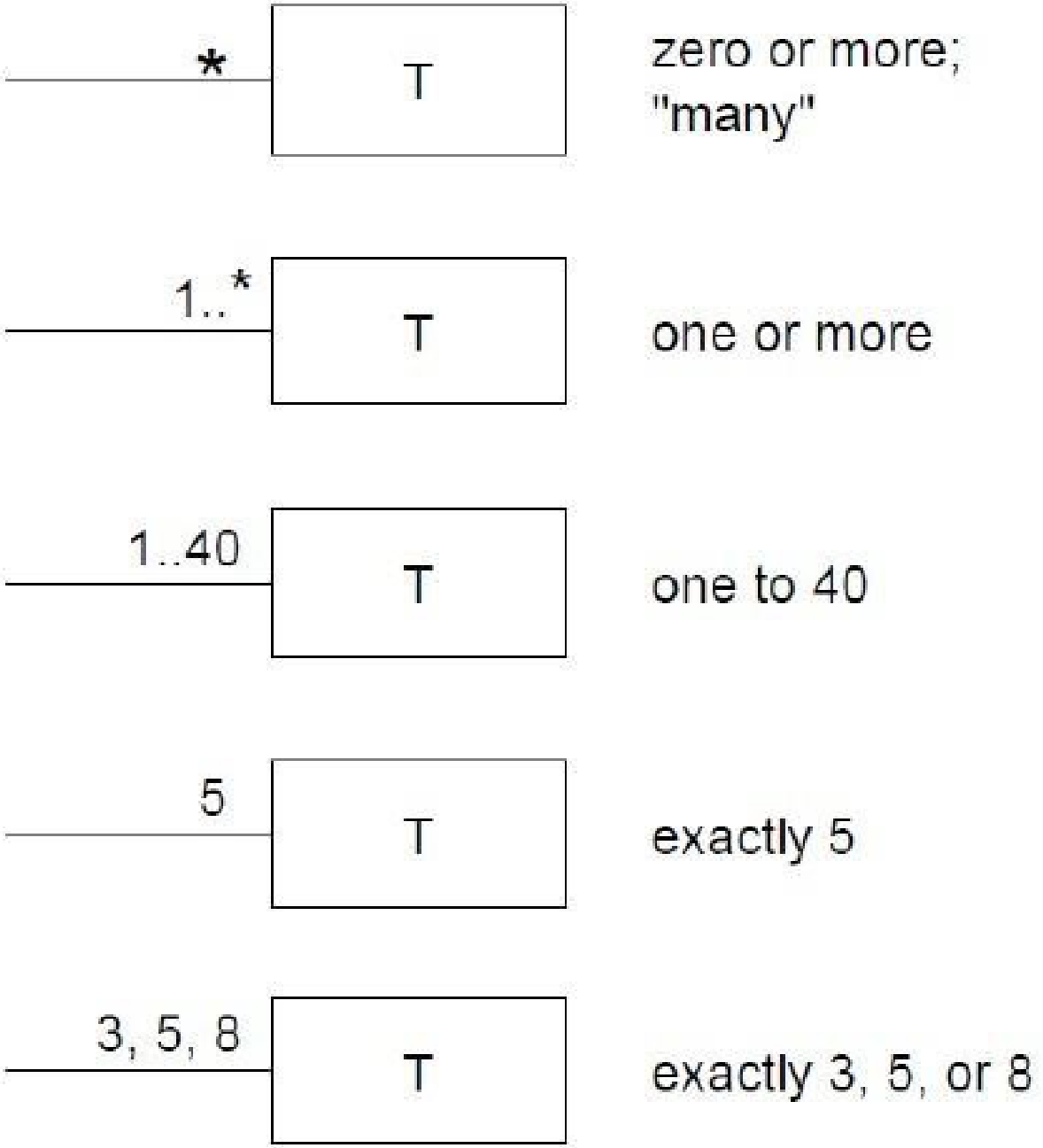
- Association can be further defined to indicate multiplicity.
- Multiplicity indicate the number of instances exist in an association
- For example:
  - Each Member ***makes*** none or many Reservation; Each Reservation belongs to 1 Member

Association Name and Multiplicity



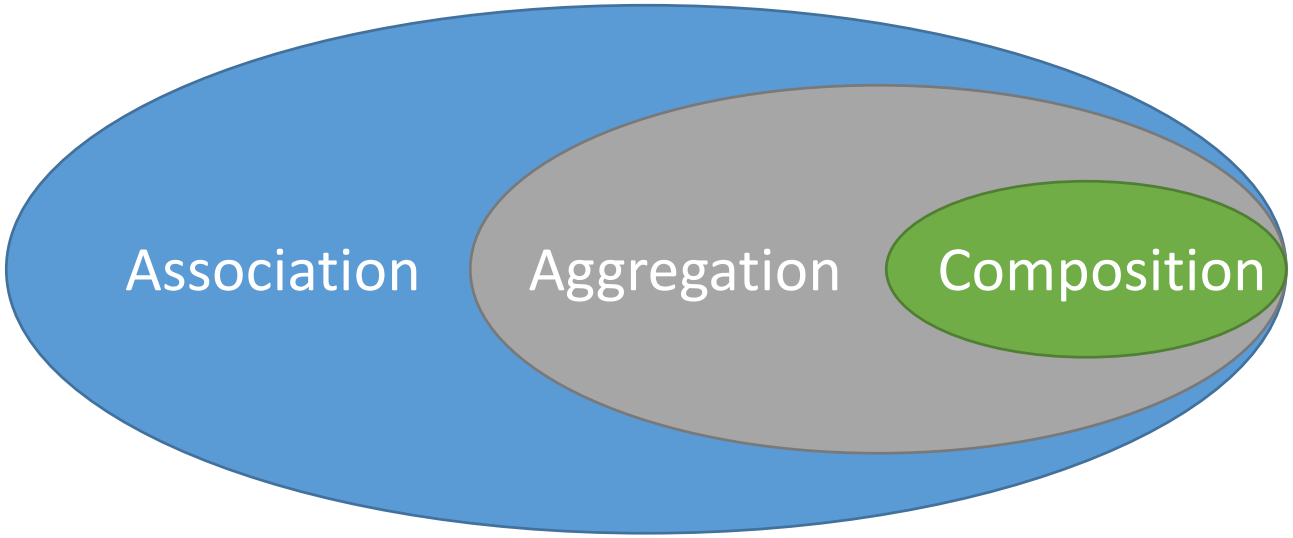
- Use verbs as the Association Name
- Use arrow to aid the reading of the Association Name

Examples of Multiplicity

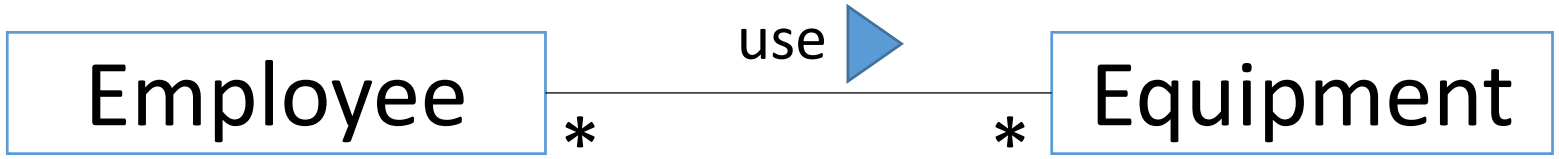


Association, Aggregation, Composition

- **Composition** and **Aggregation** are the two forms of association (object made up of another object).



Association



- **Association**
  - Relationship among objects
  - The objects merely aware of each other
  - One object can call the methods of the other object
  - The lifecycle of both objects are independent of each other

Aggregation

- **Aggregation**
  - Special type of Association
  - One object “containing” or “owning” the other object
  - “has a” relationship
  - The lifecycle of both objects are independent of each other
    - E.g. If I close the Dept, the Employee can still exist
    - Employee can work for many departments (non-unique relationship)

Composition

- **Composition**
  - A special type of Association, “Strong Association”
  - “part of” relationship
  - The containing object, the Company, exclusively “own” the Dept object
  - When containing object ended it’s lifecycle, the Dept object end with in

Implementation in Java

- Association
  - One object aware of the other object by maintaining a reference to the object
  - Creation and destruction of the reference object is not the major concern
- Aggregation
  - Containing object has the contained object(s) as one of the object properties
  - **Not responsible** for the creation and destruction of the contained object(s)
- Composition
  - Containing object has the contained object(s) as one of the object properties
  - **Responsible** for the creation and destruction of the contained object(s)

Exercise

- Identify the possible relationship between the following classes
  - Student, Instructor
  - Person, Head
  - Classroom, Desk
  - School, Programme
  - Staff, Task
  - Book, Page

Common Associations

- A is subpart/member of B. (SaleLineItem-Sale)
- A uses or manages B. (Cashier –Register, Pilot-airplane)
- A communicates with B. (Student -Teacher)
- A is transaction related to B. (Payment -Sale)
- A is owned by B. (Plane-Airline)
- A is an event related to B. (Sale-Store)

Identifying classes

- Badminton court reservation involve the following process. The player calls up the stadium to make reservation. Only member can make reservation. The player (who is also the member) tell the clerk his preferred date and time. The player can also ask the clerk for available time slot. After the clerk response with the available time slots, the player decides and confirm his reservation for the preferred time slot. Some times, a reservation may consisting several time slots so that the member can play for the longer time. The clerk will reserve a badminton court accordingly by updating the reservation records.

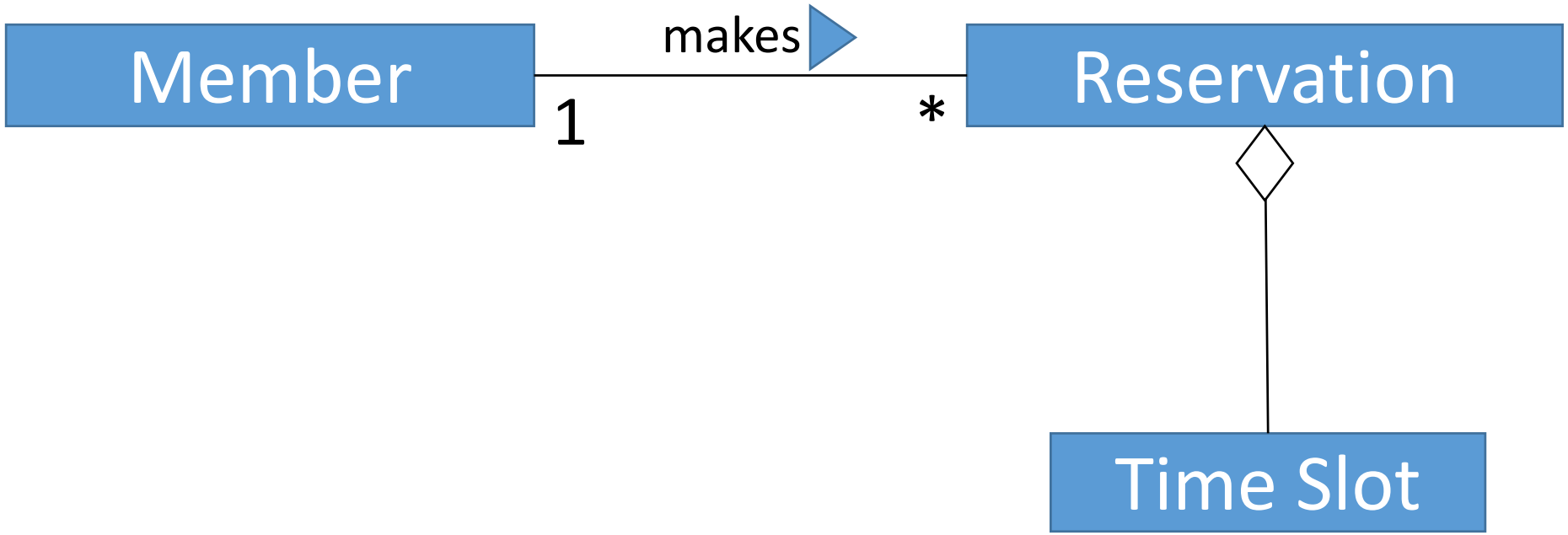
Potential Class

Potential Class	Selection Characteristics
Player (Member)	Passed
Stadium	Failed 1, 2
Reservation	Passed
Clerk	Passed (If we need to retained information such as who enter the reservation; otherwise failed 1, 2)
Time slot	Passed
Preferred date and time	Failed 3. attributes of the system
Badminton court	Failed 3. attributes of the system

Identifying relationship

- member can make reservation
- confirm his reservation for the preferred time slot
- reservation may consisting several time slots
- The clerk will reserve a badminton court

Classes with Relationship



Class Attribute

- Attributes describe an object
- Study use case to select the attribute belonging to a particular class
  - What data items fully define the class in the domain problem?
  - What information needs to be retained for the system to function as required?

Class Attribute

- According to the HomeSafe requirements, which attributes are required for the System class?
  - Master password
  - Telephone number
  - Delay time
  - Dialing delay, dialing retry
  - System status
- Some of the attributes are the compounded attribute that you may or may not want to break it down to elementary attributes, and vice versa.
- In the modeling for requirement analysis, data type and access scope of the attributes are not required

Class operations

- Operations define the behavior of an object.
- 4 broad categories. Operations that: -
  - manipulate data
  - Perform computation
  - Inquire about the state of an object
  - Monitor an object for the occurrence of event
- Operations act on class attributes and/or associations

Identifying class operations

- examining the usage scenarios developed in the use case model.
- Verbs used in the scenario is the potential operations for classes.
- Example: -
  - *arming* and *disarming* the system
  - PC is used to *program* and *configure* the system

Identifying class operations

- arm() and disarm() operations will be applied to the System class
- Some operations are too broad and should be divided into sub-operations.
- configure() operation for the system class can be divided into: -
  - configure()
  - addSensor()

Identifying class operations

- Decisions on which operations should each classes be responsible is the concern in the design phrase
- Design concept involving modularity and coupling