Introduction

- **Domain Model** is the visual representation of conceptual Class, real situation objects in a domain.
- It is a representation of real situation conceptual class not a software object.
- In UP domain model is define as one of the artifact that may be used in business modeling to explain product’s importance in business domain.
- In UML it shows with class diagram but it does not represent operation(method).

Conceptual Perspective of Domain Model

- Domain Object or Conceptual Class
- Association between conceptual class
- Attributes of conceptual class
- It does not contain database code
- It does not contain responsibilities or method

Conceptual Class

- It is a idea, thing or object or abstraction of concept from problem real world domain. For eg. Sale, register, Item etc.
- It is a class of Idea or thing which are view from real world situation not from software artifact.
- It can be considered in terms of its symbol, intention and extension.

Terms of Conceptual Class

- Symbol – Word or Image representing a conceptual class.
- Intension – The definition of conceptual class. For eg. Sale represent the event of purchase transaction.
- Extension – The set of examples to which the conceptual class applies. The extension of sale is sale-1, sale-2.

Steps to create Domain Model

- Find the conceptual class
- Draw them as classes in a UML class diagram without operation.
- Add association and attributes

Strategies to find conceptual class

- Reuse or modify existing model
- Use Category List
- Identify Noun Phrase

Conceptual Class by using Category list

Categories	Examples
Business Transaction which involves Money	Sale, Payment, Reservation
Transaction Line Item	SalesLineItem
Product or service related to transaction or Line Item	Item, flight, seat, book
Where is the transaction recorded	Register, Ledger, FlightManifest and Shelf
Role of people or organization related to the transaction or actor	Cashier, Librarian, Customer, Store, Passenger, airline
Place of transaction or place of service	Store, Library, Airport, Seat
Description of things	ProductDescription, FlightDescription
Catalog	ProductCatalog, FlightCatalog
Financial Instrument	Cash, Check, TicketCredit
Schedules to do work	DailyPriceChangeList, RepairSchedule

Finding Conceptual Classes using noun phrase Identification

- **Customer** arrives at **store** and select goods.
- **Cashier** starts a new sale.
- Cashier enters the **Item Code**.
- System present the total with **tax** calculated.
- System present the **receipt.**

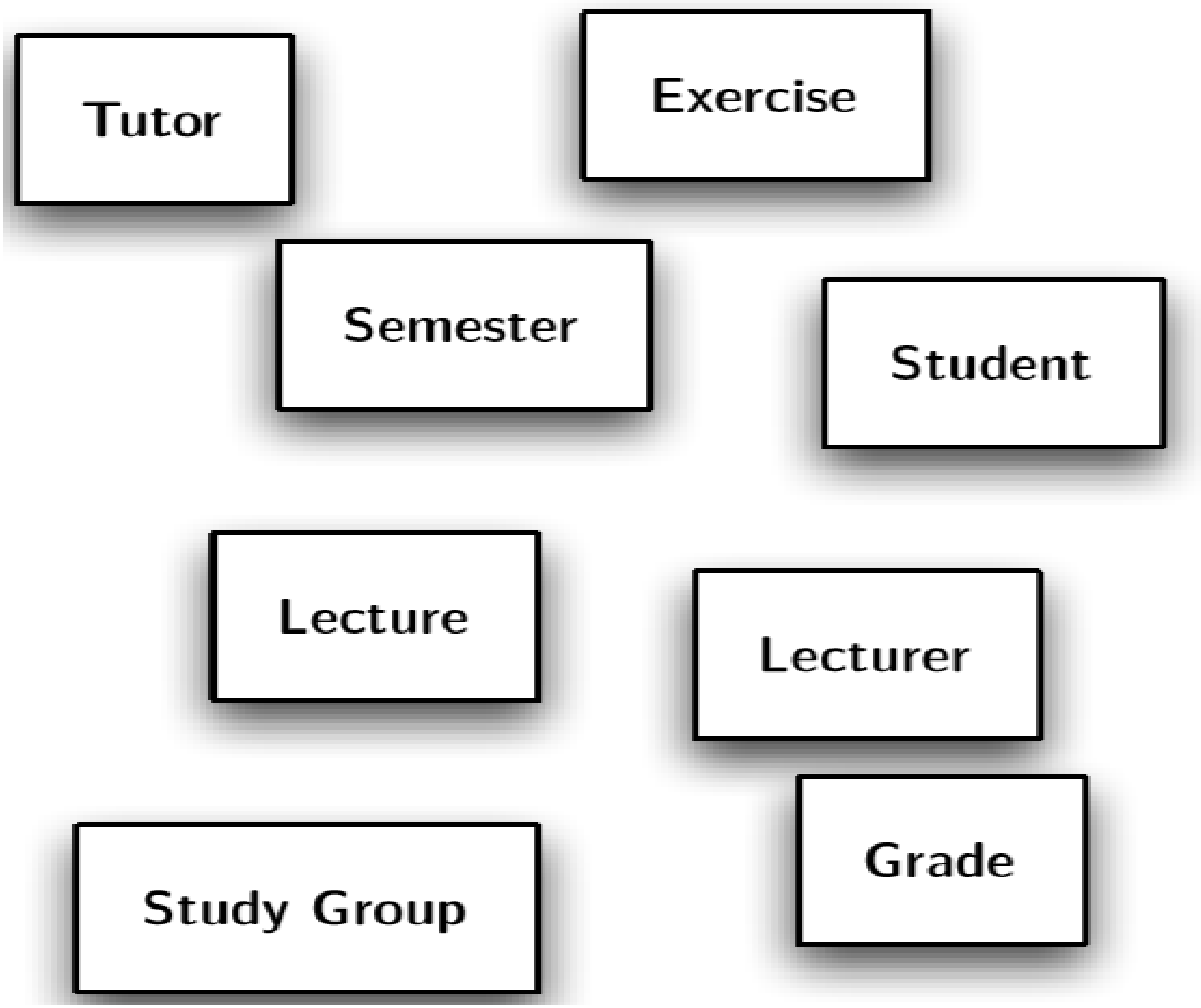
Example of finding conceptual Class of POS

- **Sale –** Business Transaction
- **Item –** Product or Service Related to transaction
- **CashPayment –** Business Transaction
- **Store –** Container or Place of Store
- **Register –** Where the transaction Recorded
- **Ledger –** Where the transaction are Recorded
- **Cashier –** Actor
- **Customer –** Actor
- **ProductDescription –** Description of Thing
- **ProductCatalog –** Catalog

Course Registration System Concept

- During a semester a **lecturer** reads one or more **lectures.**
- A **Student** usually attends one or more **lectures.**
- During the **semester** there will be several **exercises.**
- Each Student is assigned to one particular **study group** for the whole semester.
- Result is graded by **Tutor.**

Course Registration System Concept



Tutor

Exercise

Semester

Student

Lecture

Lecturer

Grade

Study Group

Why Description Class

- It is such class which contains information that describes something else or provide information about other class.
- For eg. Productdescription that record the price, text description of an Item.
- It represents the description of information about item.

Common Mistake in Finding Conceptual Class

- When certain things required physical space it become class not attributes.
- Object can be represent as class because it require physical space.
- An Event can be recognized as Class not as attributes.
- Real Existing things are recognized as class not as attributes.

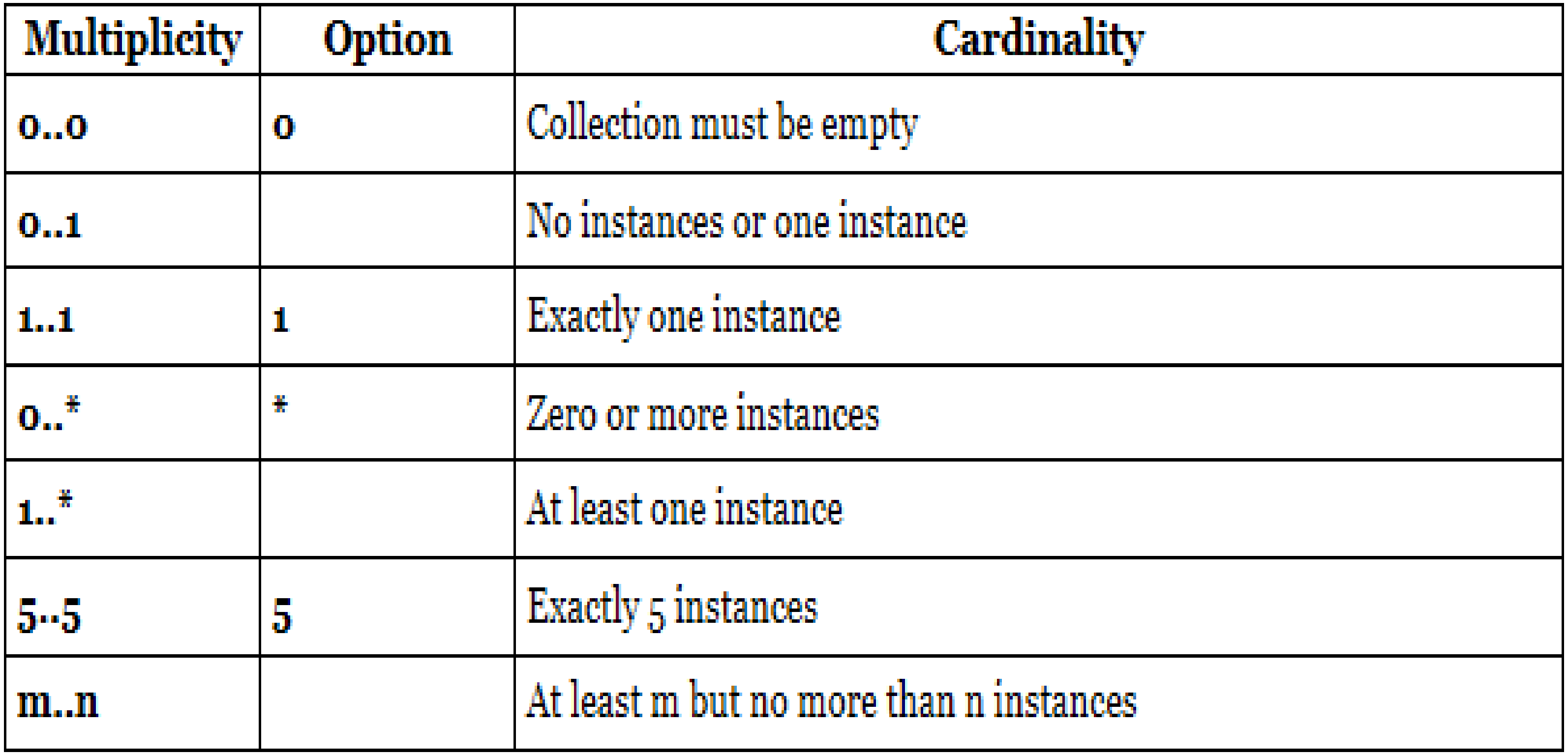
Add Association and Attributes

- It is a relationship between classes that indicates some meaningful connection.
- In UML, association are defined as “ the semantic relationship between two or more classifiers that involve connection among their instance”.
- It is represented by straight line.

Multiplicity

- It defines how many instance of one class can be associated with one instance of another class.
- For eg. One instance of store can be associated with many instance of item instances.

Multiplicity



Multiplicity	Option	Cardinality
0..0	0	Collection must be empty
0..1		No instances or one instance
1..1	1	Exactly one instance
0..*	*	Zero or more instances
1..*		At least one instance
5..5	5	Exactly 5 instances
m..n		At least m but no more than n instances

Multiple Association

- It is such type of association in which the meaning of differ from one instance to another class.
- For eg. Flight flies to Airport and Flight Flies from Airport indicate different relationship.

When to Show Relationship

- Association are shown if a relationship between the classes needs to be preserved for some duration or if we need some memory of relationship between the objects.
- The following association Should be included in domain model.
  - Association for which relationship needs to be preserved form some duration (“Need to remember”) Association.
  - Association derived from the common Association list.

Attributes

- An Attribute is a logical data value of an object.
- It is necessary to identify those attributes of conceptual class that are needed to satisfy the information requirements of the current scenario.
- In UML Notation attributes are shown in the second compartment of the class box.

Domain Model

- Domain Model is a visual representation of conceptual class or real situation in a domain. It is also called conceptual model , domain model or analysis object model.
- According to unifiied process domain model is a part of business model and also called as business model.
- The refined domain model represents
  - Conceptual Class
  - Association
  - Attributes

Domain Model

- Domain Model is a visual representation of conceptual class or real situation in a domain. It is also called conceptual model , domain model or analysis object model.
- According to unifiied process domain model is a part of business model and also called as business model.
- The refined domain model represents
  - Conceptual Class
  - Association
  - Attributes

Domain Model refinement

- The domain models are created to understand the requirements of the system.
- A useful domain model captures the essential abstraction and information required for understanding the domain of the targeted system.
- The domain model can be refined by adding the attributes to the class and showing the associations among these classes.

Things not seen before domain Model

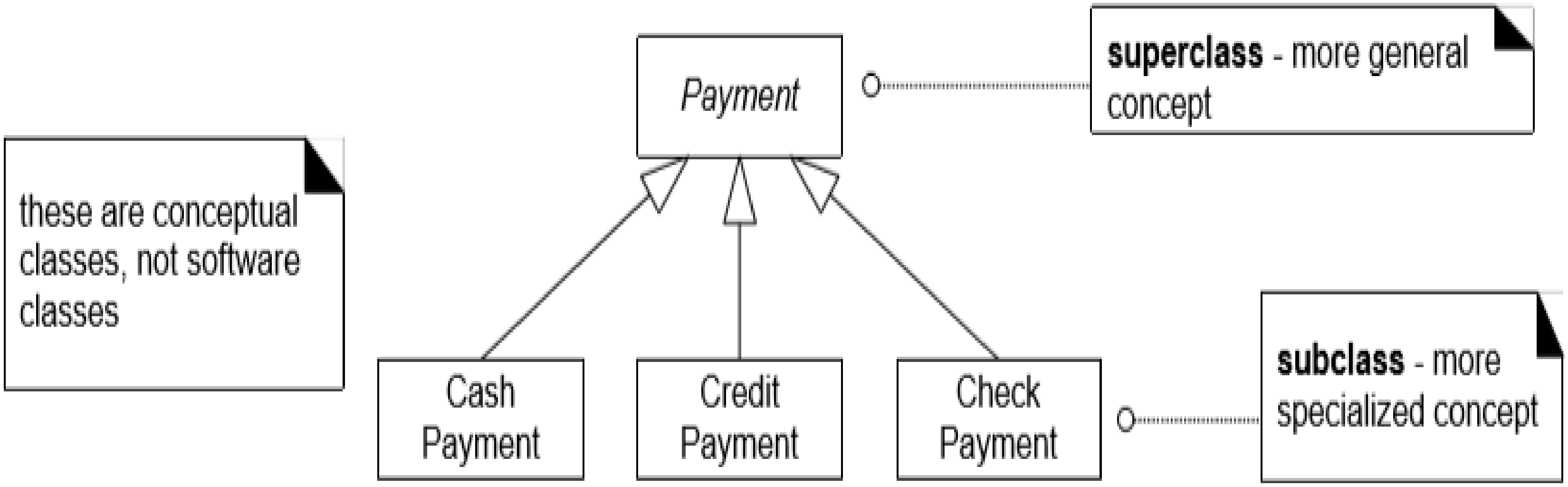
- Similar to the concepts in the object model.
- Generalization and Specialization.
- Conceptual Class Hierarchies
- Association class that capture information about the association.
- Time intervals
- Packages as a means of organization

Things not seen before domain Model

- Similar to the concepts in the object model.
- Generalization and Specialization.
- Conceptual Class Hierarchies
- Association class that capture information about the association.
- Time intervals
- Packages as a means of organization

Domain Class

Domain classes



Payment

superclass - more general concept

these are conceptual classes, not software classes

Cash Payment

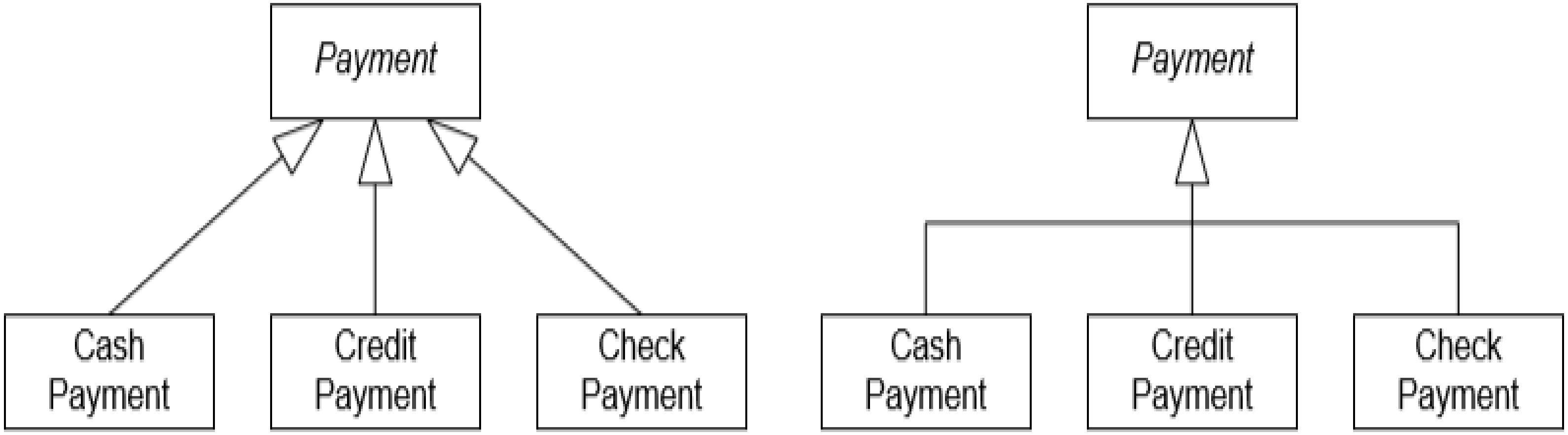
Credit Payment

Check Payment

subclass - more specialized concept

Domain Class

They each show the same thing



Conceptual Superclass and Subclass

- *Conceptual* superclass is more general than subclass
  - *All members* of subclass must be members of the superclass
  - *100% of superclass definition shall apply* to the subclass
- Subclass —is-all superclass
  - Woman —is-all human
  - Man —is-all human
  - Anything else that is a human?
  - All humans have a heart and a brain.

When to use Subclass

- Start with the super-class and look at differences to find the sub-classes that make sense.
  - Subclass has additional attributes
  - Subclass has additional associations
  - Subclass usage differs from superclass
  - Subclass represents an animate entity that behaves differently

When to use Superclass

- Start with a set of sub-classes and look for commonalities that help the model.
  - Potential subclasses represent variations of concept
  - Subclasses meet —is-all rule and 100% rule
  - All subclasses have common attributes that can be *factored* out
  - All subclasses have the same association that can be *factored* out

Aggregation

- Aggregation is a special case of association when an object ‘has-a’ another object, which you can have an aggregation between them.
- The direction between them helps you to specify which object contains the other object. Aggregation is also known as a “Has-a” relationship.
- It is a subtype of an association relationship in UML.

Aggregation

- For example, a car consists of wheels, engine, gearbox, steering, main body, etc. A car is an assembly, and the other parts are its constituents.
- Car to wheels is one aggregation, car to engines is another aggregation, and so on.
- Each individual pairing is defined to specify the multiplicity of each constituent part within the assembly as an outcome the number of objects can also be depicted.

Composition

- The **composition** is also a type of association but a more restrictive form.
- It is represented in UML by a tiny solid diamond adjacent to the assembly class.
- In this type of association, the constituent part can be linked to at most one assembly. Additionally, when a constituent part collaborates with assembly, it will have the concurrent lifetime with the assembly.

Aggregation	Composition
Association that show ‘has-a’ relationship.	Specific aggregation which implies ownership and show ‘part-of’ relationship.
It is indicated using a straight line with an empty arrowhead at one end.	It is indicated using a straight line with a filled arrowhead at any one of the ends.
It is weak type of association.	It is strong type of association.
Linked objects are not dependent upon the others.	Objects are highly dependent upon each other.
It is denoted by a filled Diamond.	It is denoted by a empty diamond.

UML Activity Diagram and Modelling

- A UML activity diagram shows sequential and parallel activities in a process.
- Activity Diagram are graphical representation of the work flows of stepwise activities and action.
- They are useful for modeling business processes workflows, data flows and complex algorithms.
- The Diagram shows the sequence of action some of which may be parallel and it also shows both control flow and data flow.

Activity Diagram Notation

- Initial State or Starting point
- Activity or action state
- Action flow
- Object flow
- Decision and Branching
- Guards
- Synchronization Bar
- Time Event
- Swimlanes
- Rake
- Final State or end point