

WHAT IS A CLASS DIAGRAM?



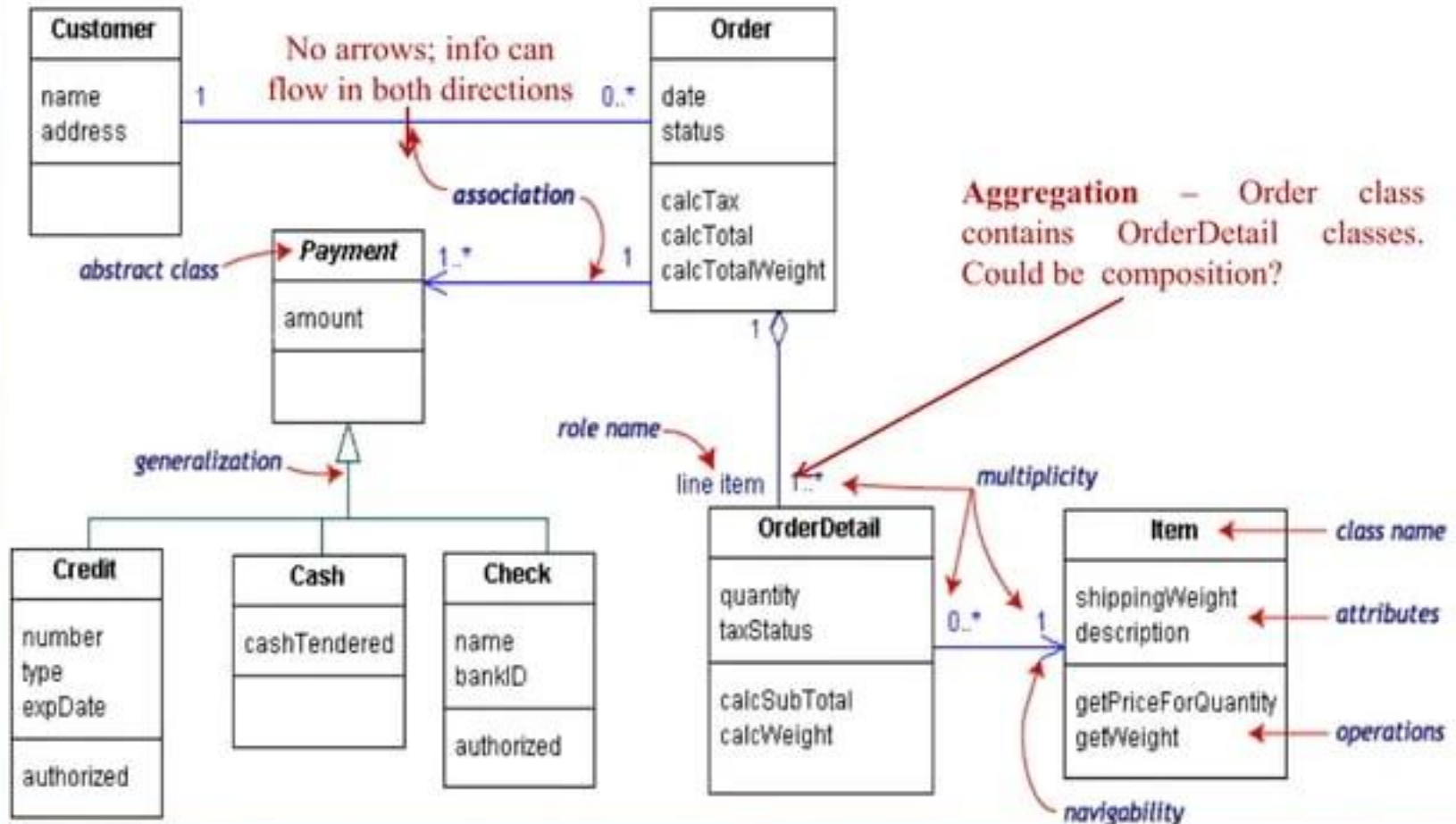
- ❑ The class diagram is a **static model** that shows the **classes** and the **relationships among classes** in the system.
- ❑ Class diagrams illustrates **classes**, **interfaces**, and **their associations**. They are used for static object modeling.
- ❑ A Class diagram gives an **overview of a system** by **showing its classes** and the **relationships among them**.
- ❑ Class diagrams display what interacts but not what happens when they do interact

WHAT IS A CLASS?



- ❑ The **main** building block of a class diagram is the **class**, which **stores** and **manages information** in the **system**.
- ❑ A blueprint that an object made from.
 - The class contains two major parts:
 - Attributes “data members”.
 - Operations “method members”.

EXAMPLE 1:



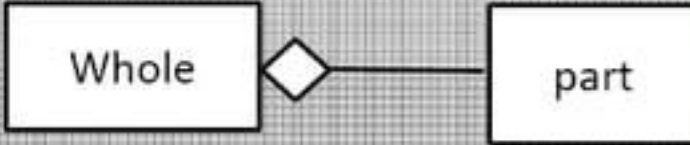

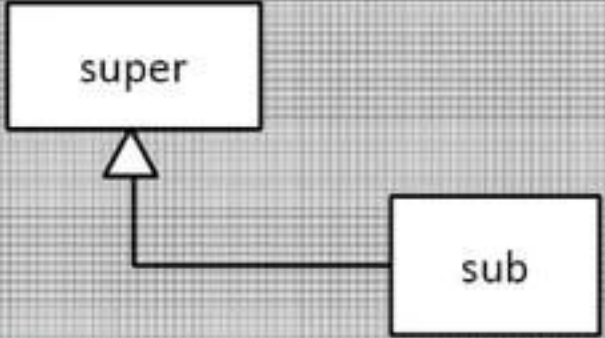
CLASS DIAGRAMS SYMBOLS



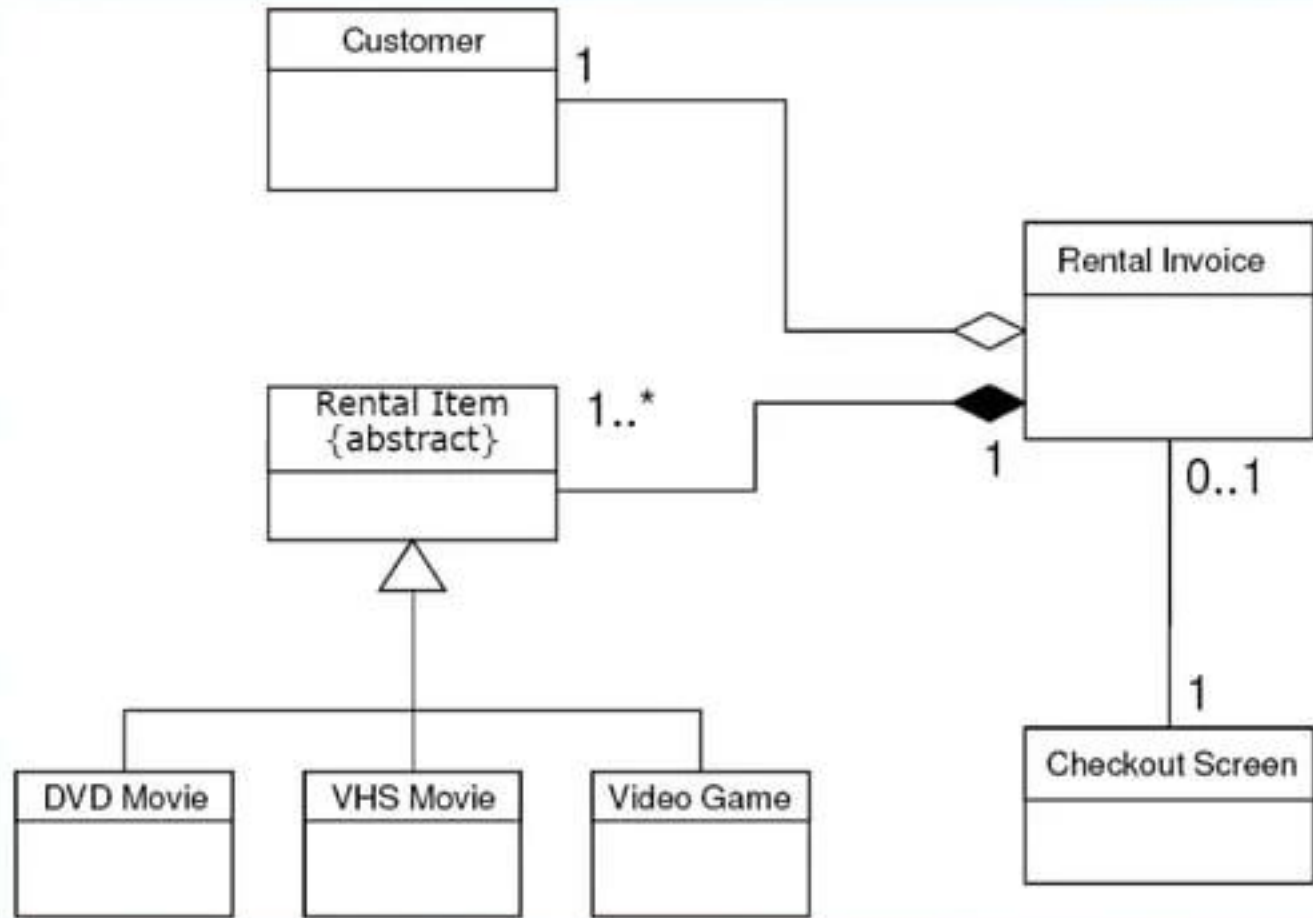
Description	symbol
<p>Class : represents a person, place, or thing about which the system will need to <u>capture</u> and <u>store</u> information.</p> <p>Attribute: Represents <u>properties</u> that <u>describe</u> the <u>state</u> of an <u>object</u>. Must be named with <u>noun</u>.</p> <p>Method “operation”: Represents the <u>actions</u> or <u>functions</u> that a <u>object</u> can <u>perform</u>. Must be named with <u>verb</u>.</p>	<div>Class name</div> <div>Attributes list</div> <div>Methods list</div>
<p>Association: Represents a <u>relationship</u> between <u>multiple</u> <u>classes</u>, or a class and <u>itself</u>. Must be labeled with <u>verb</u>.</p>	<u>association name</u>
<p>Multiplicity: Represents the <u>minimum</u> and <u>maximum</u> times a <u>class instance</u> can be <u>associated</u> with the related <u>class instance</u>.</p>	<div>0..* – 1 1 – 1</div> <div>1..* - 0..* 1 – m</div> <div>1..7 – 4..9 m - n</div>

CLASS DIAGRAMS SYMBOLS

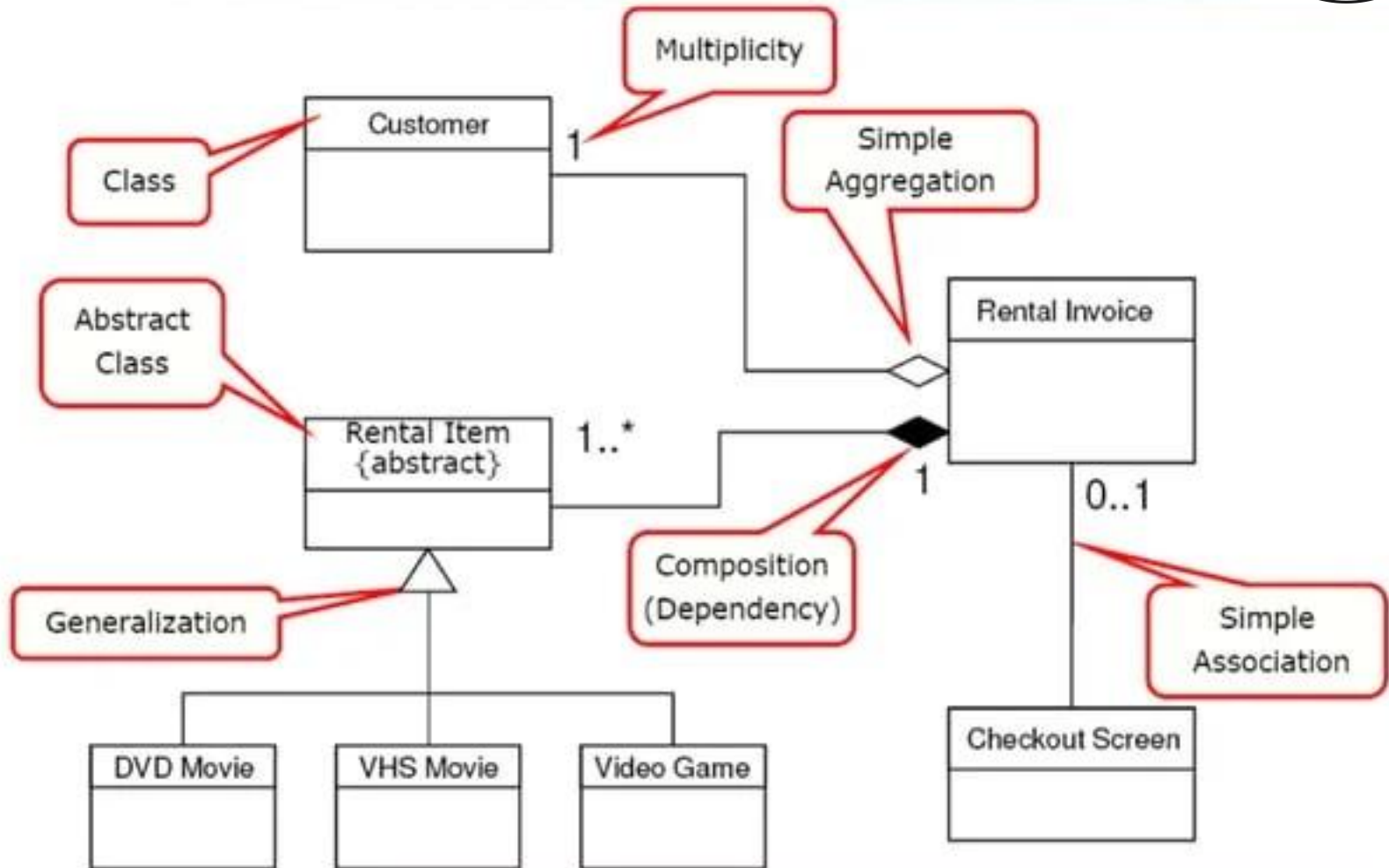


Description	Symbol
<p>Aggregation: is a <u>weak whole - part</u> relationship, <u>when an object is part of other object(s). (is part of).</u></p>	 <pre> graph LR Whole[Whole] o-- part[part] </pre>
<p>Composition: is a <u>strong whole – part</u> relationship, <u>when an object is part of another object.</u> If the <u>container (whole) object</u> is <u>destroyed</u> the <u>part object</u> is also <u>destroyed</u>.</p>	 <pre> graph LR Whole[Whole] *-- part[part] </pre>
<p>Generalization “ inheritance” : when a class (<u>subclass</u>) <u>inherits</u> from <u>another class (super class)</u>, meaning that the <u>properties</u> and <u>operations</u> of the <u>super class</u> are also <u>valid</u> for objects of the <u>subclass</u>. (is – a).</p>	 <pre> graph BT sub[sub] -- > super[super] </pre>

UML - CLASS DIAGRAM



UML - CLASS DIAGRAM



□ Class diagrams are great for:

- discovering related data and attributes
- getting a quick picture of the important entities in a system
- seeing whether you have too few/many classes
- seeing whether the relationships between objects are too complex, too many in number, simple enough, etc.
- spotting dependencies between one class/object and another

□ Not so great for:

- discovering algorithmic (not data-driven) behavior
- finding the flow of steps for objects to solve a given problem
- understanding the app's overall control flow (event-driven? web-based? sequential? etc.)



EXAMPLE

- Draw a class diagram for a information modeling system for a school.
 - School has one or more Departments.
 - Department offers one or more Subjects.
 - A particular subject will be offered by only one department.
 - Department has instructors and instructors can work for one or more departments.
 - Student can enrol in upto 5 subjects in a School.
 - Instructors can teach upto 3 subjects.
 - The same subject can be taught by different instructors.
 - Students can be enrolled in more than one school.

CLASS DIAGRAM - EXAMPLE



- School has one or more Departments.



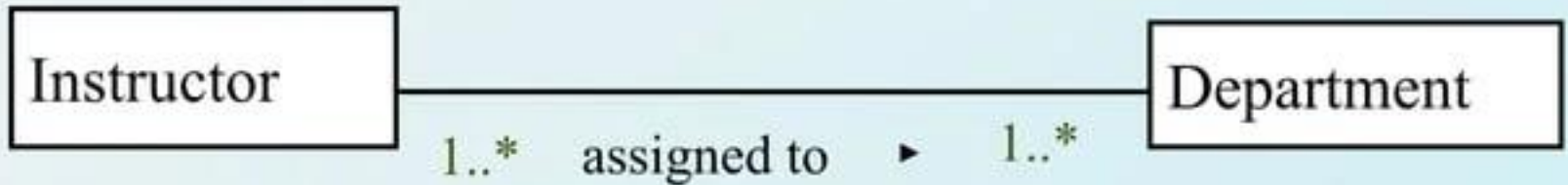
- Department offers one or more Subjects.
- A particular subject will be offered by only one department.



CLASS DIAGRAM - EXAMPLE



- Department has Instructors and instructors can work for one or more departments.



- Student can enrol in upto 5 Subjects.



CLASS DIAGRAM - EXAMPLE



- Instructors can teach up to 3 subjects.
- The same subject can be taught by different instructors.



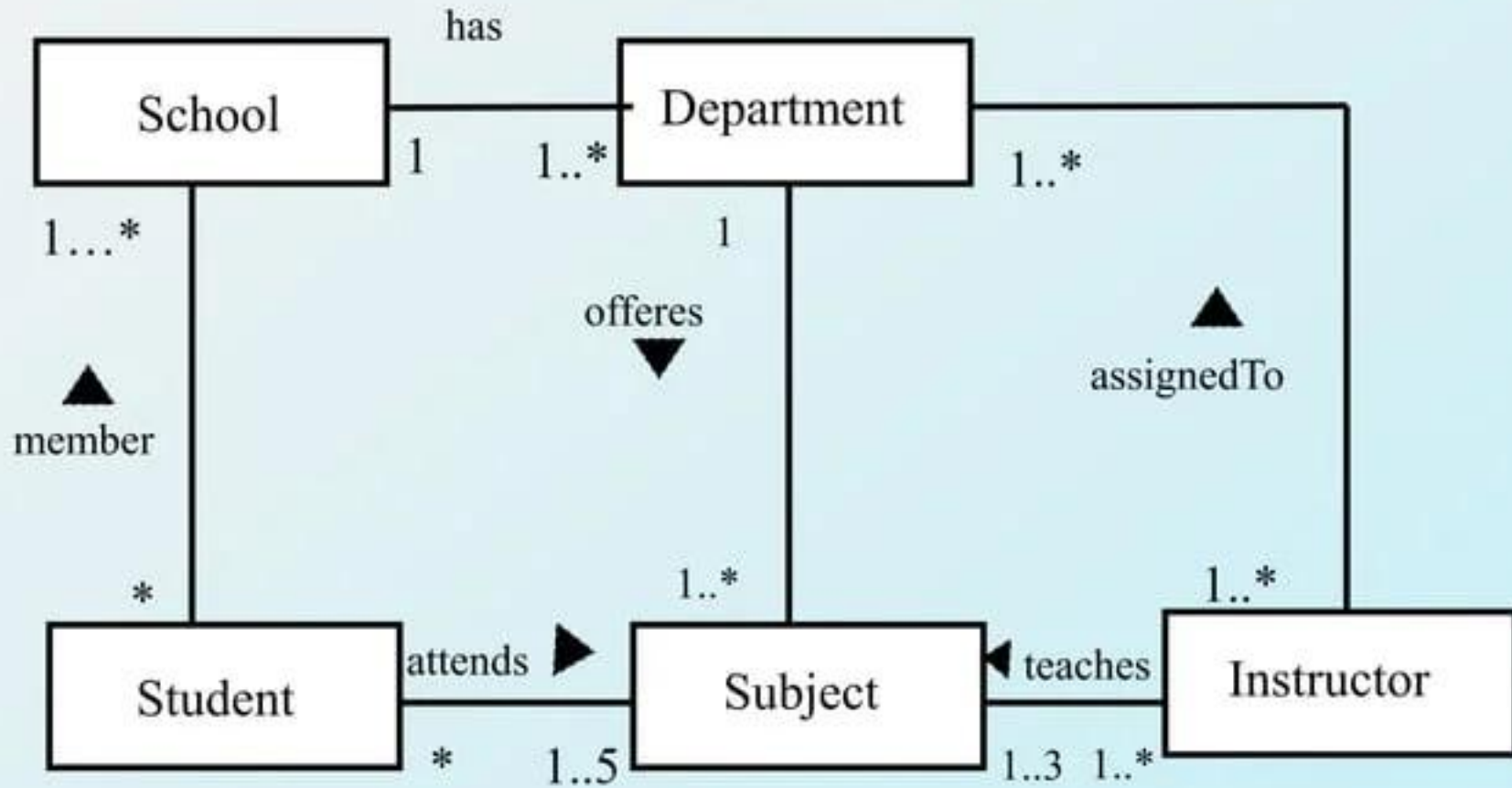
CLASS DIAGRAM - EXAMPLE



- Students can be enrolled in more than one school.



CLASS DIAGRAM EXAMPLE

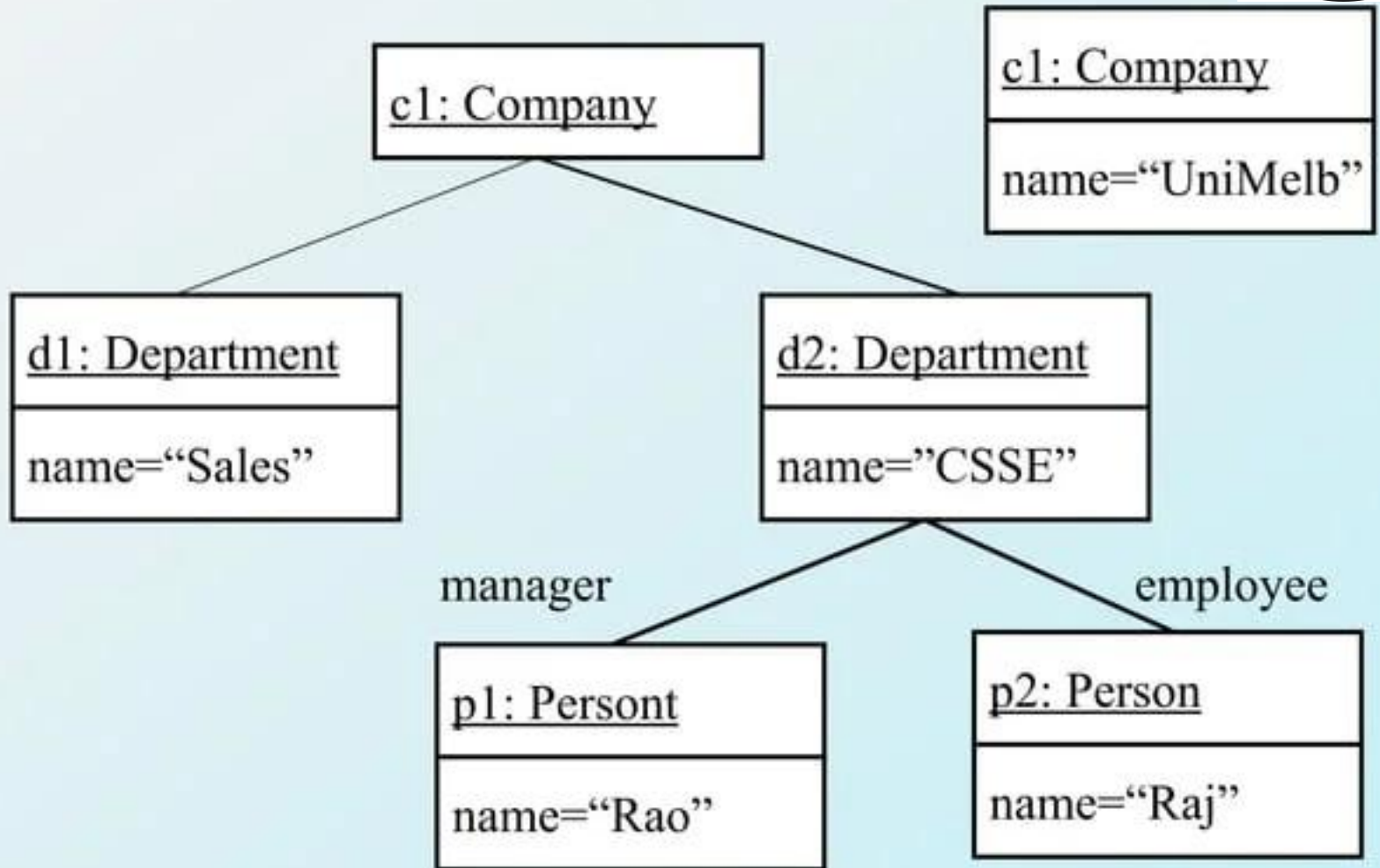


OBJECT DIAGRAM



- ❑ Object Diagram shows the relationship between objects.
- ❑ Unlike classes objects have a state.

OBJECT DIAGRAM - EXAMPLE





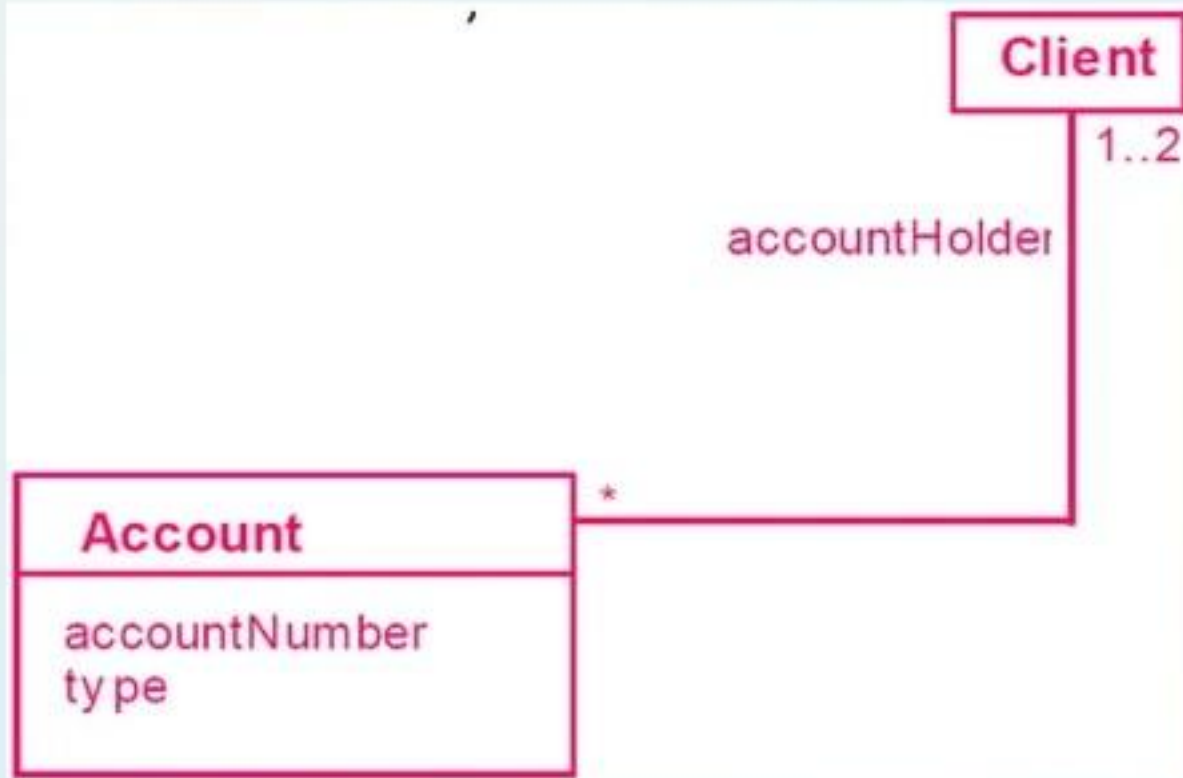
EXAMPLE

- This system provides the basic services to manage bank accounts at a bank called OOBank. OOBank has many branches, each of which has an address and branch number. A client opens accounts at a branch. Each account is uniquely identified by an account number; it has a balance and a credit or overdraft limit. There are many types of accounts, including: A mortgage account (which has a property as collateral), a chequing account, and a credit card account (which has an expiry date and can have secondary cards attached to it). It is possible to have a joint account (e.g. for a husband and wife). Each type of account has a particular interest rate, a monthly fee and a specific set of privileges (e.g. ability to write cheques, insurance for purchases etc. OOBank is divided into divisions and subdivisions (such as Planning, Investments and Consumer), the branches are considered subdivisions of the Consumer Division. Each division has a manager and a set of other employees. Each customer is assigned a particular employee as his or her 'personal banker'.

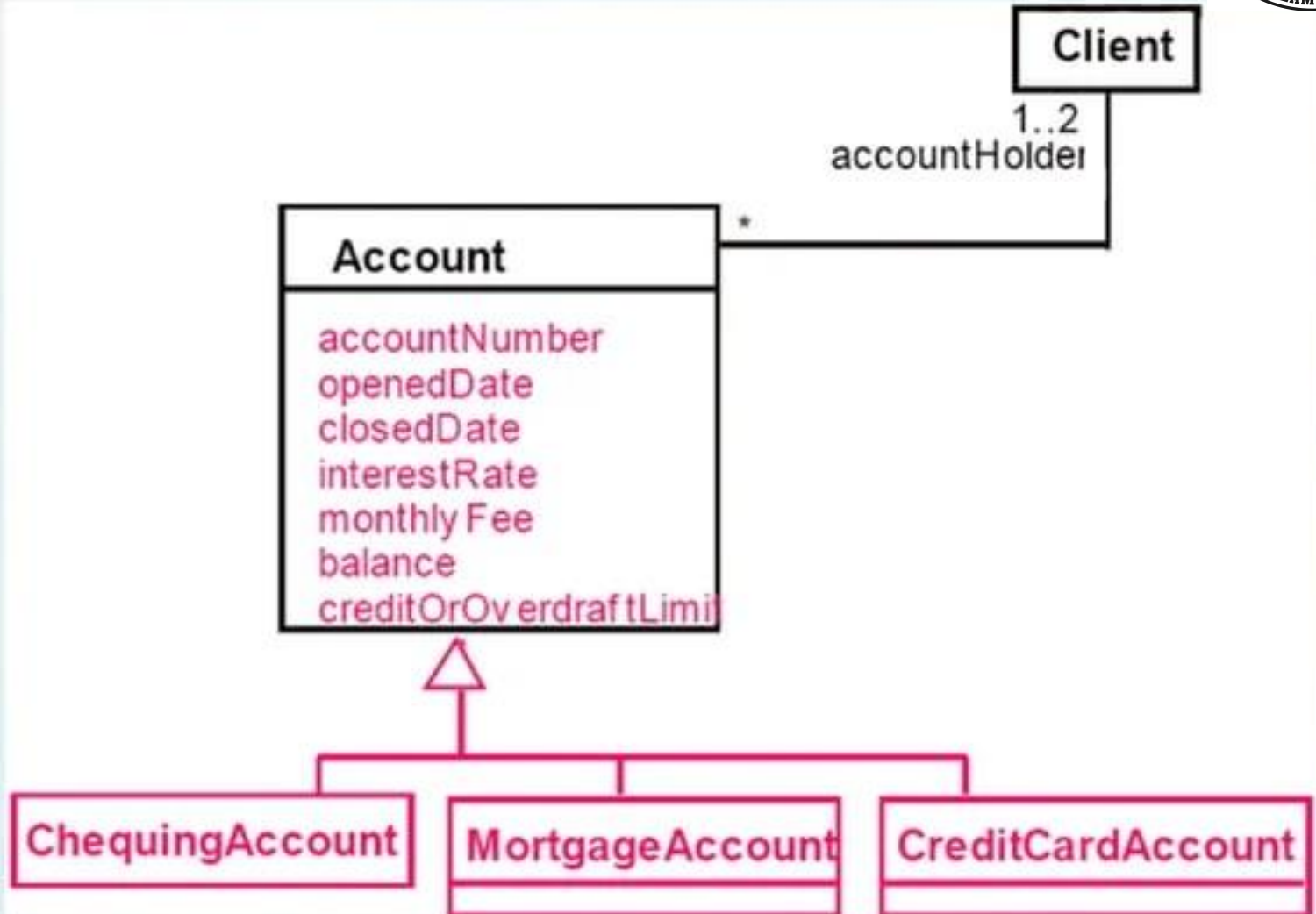
Marking nouns: potentially good classes, definitely *bad* classes, and classes we are unsure about

- This *system* provides the basic services to manage bank accounts at a bank called *OOBank*. OOBank has many branches, each of which has an address and *branch number*. A client opens accounts at a branch. Each account is uniquely identified by an account number; it has a balance and a credit or overdraft limit. There are many types of accounts, including: A mortgage account (which has a property as collateral), a chequing account, and a credit card account (which has an expiry date and can have secondary cards attached to it). It is possible to have a joint account (e.g. for a *husband* and *wife*). Each type of account has a particular *interest rate*, a *monthly fee* and a specific set of privileges (e.g. ability to write cheques, insurance for purchases etc. OOBank is divided into divisions and *subdivisions* (such as *Planning*, *Investments* and *Consumer*), the branches are considered subdivisions of the Consumer Division. Each division has a manager and a set of other employees. Each customer is assigned a particular employee as his or her ‘personal banker’.

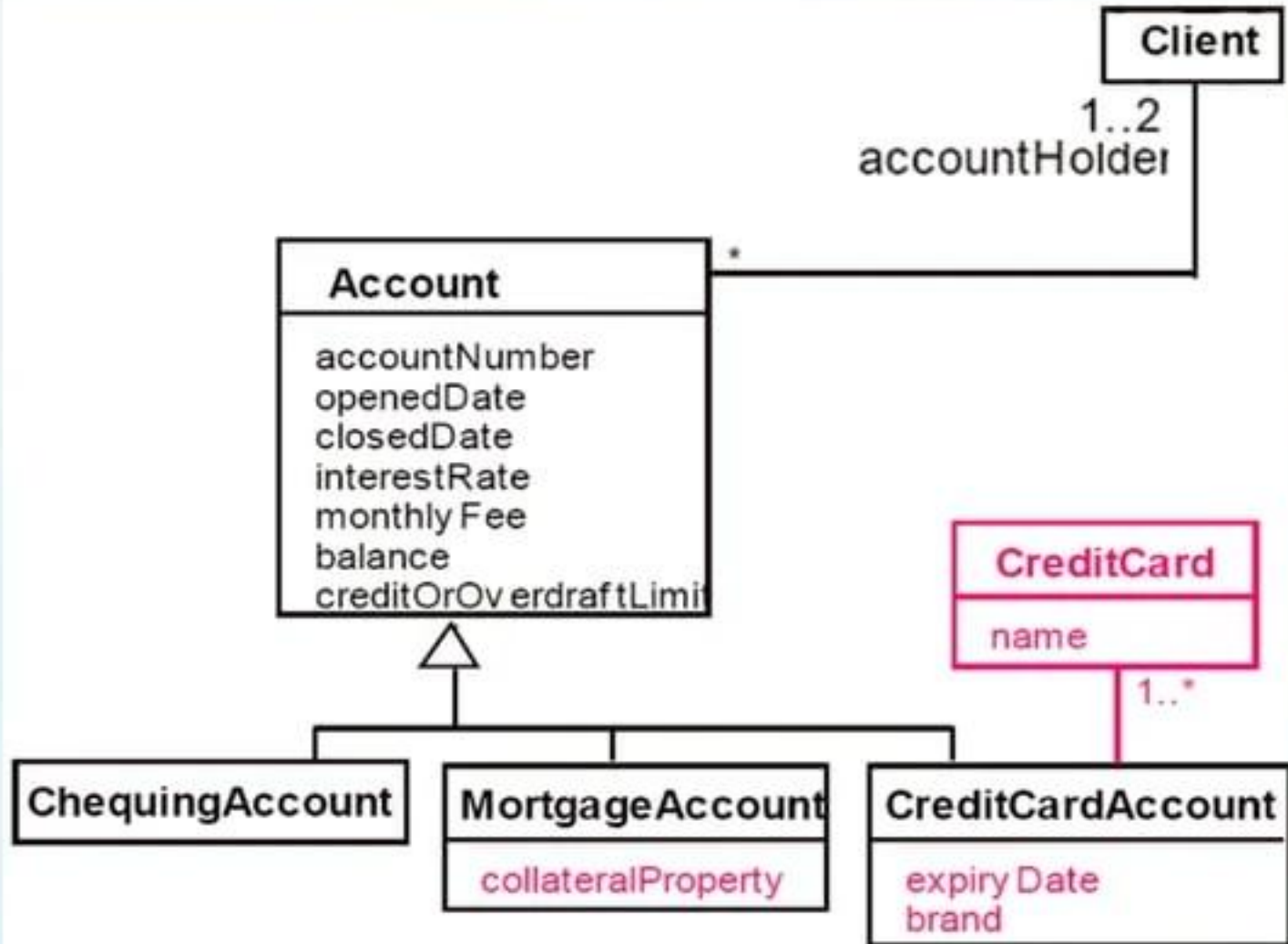
DRAFT CLASS DIAGRAM WITH CLASSES CLIENT AND ACCOUNT, AND THEIR ASSOCIATION



DRAFT CLASS DIAGRAM WITH ACCOUNT ATTRIBUTES AND THEIR SUBCLASSES



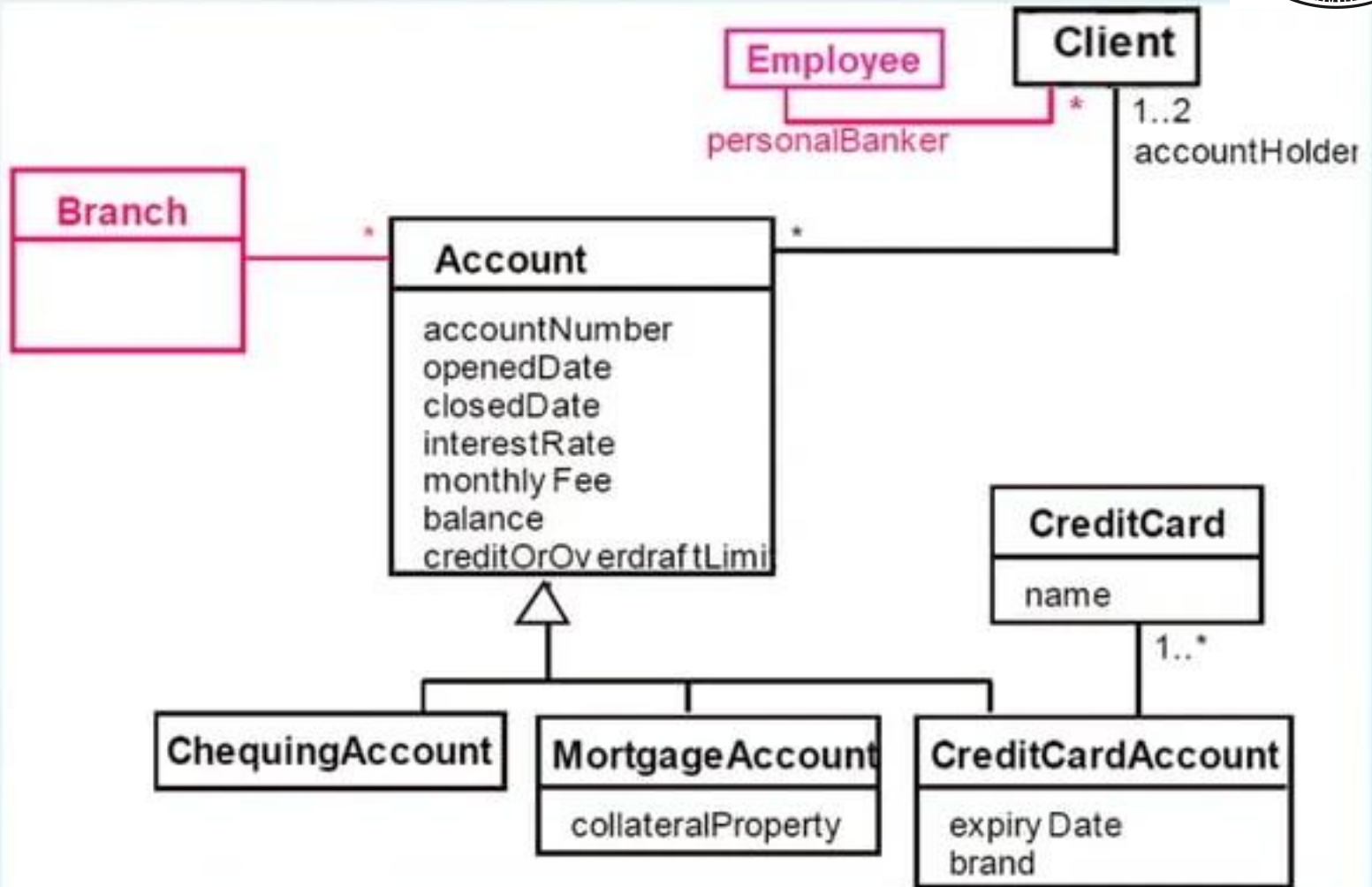
DRAFT CLASS DIAGRAM WITH CREDIT CARD AND ATTRIBUTES OF ACCOUNT SUBCLASSES



Problem statement - checking off what we have done (Classes, associations, attributes, generalizations)

- This *system* provides the basic services to manage bank accounts at a bank called *OOBank*. OOBank has many branches, each of which has an address and *branch number*. A client opens accounts at a branch. Each account is uniquely identified by an account number; it has a balance and a credit or overdraft limit. There are many *types of accounts*, including: A mortgage account (which has a property as collateral), a chequing account, and a credit card account (which has an expiry date and can *have* secondary cards attached to it). It is possible to *have* a joint account (e.g. for a *husband* and *wife*). Each type of account has a particular *interest rate*, a *monthly fee* and a specific set of privileges (e.g. ability to write cheques, insurance for purchases etc. OOBank is divided into divisions and *subdivisions* (such as *Planning*, *Investments* and *Consumer*), the branches are considered subdivisions of the Consumer Division. Each division has a manager and a set of other employees. Each customer is assigned a particular employee as his or her 'personal banker'.

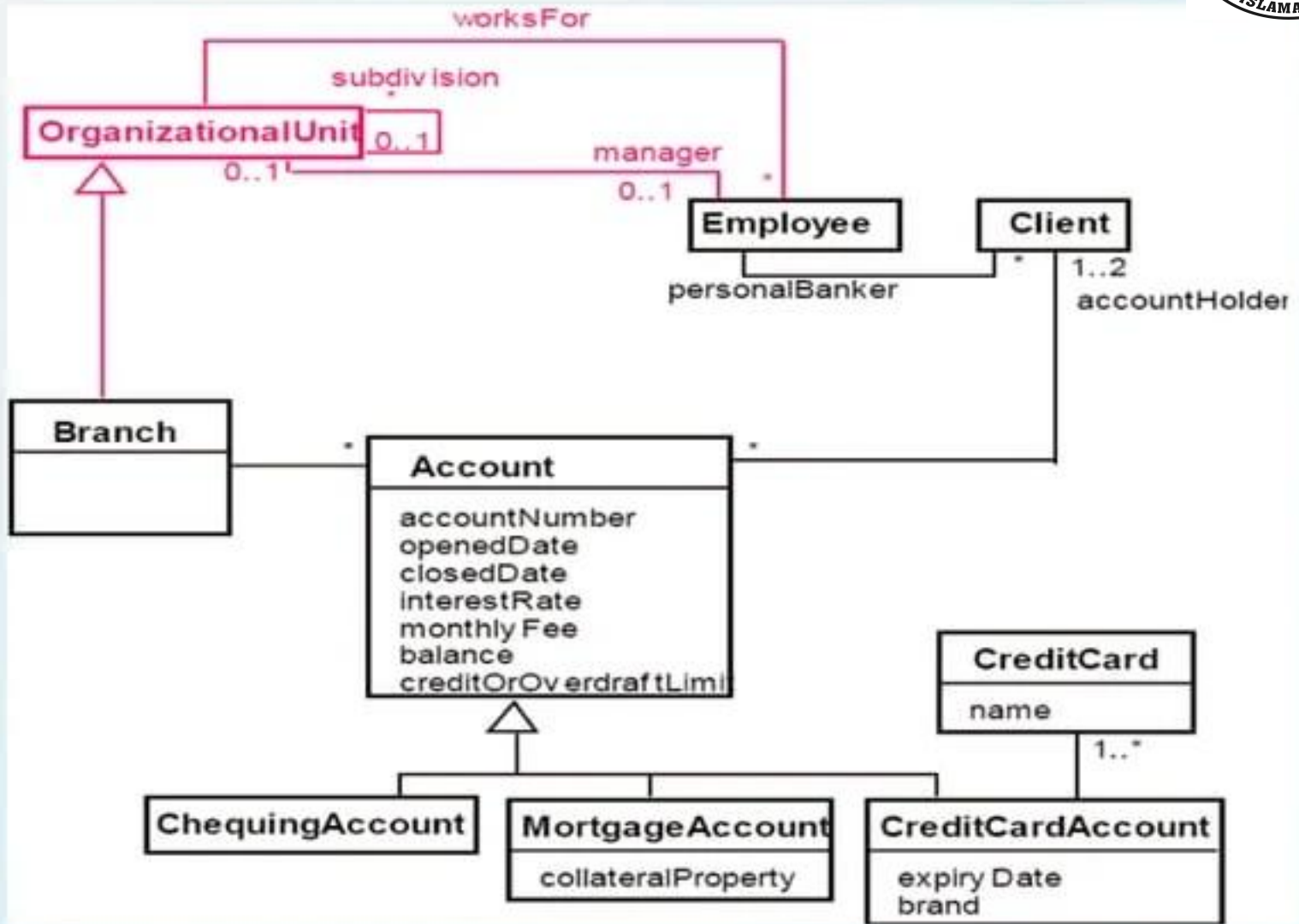
DRAFT CLASS DIAGRAM WITH BRANCH AND EMPLOYEE

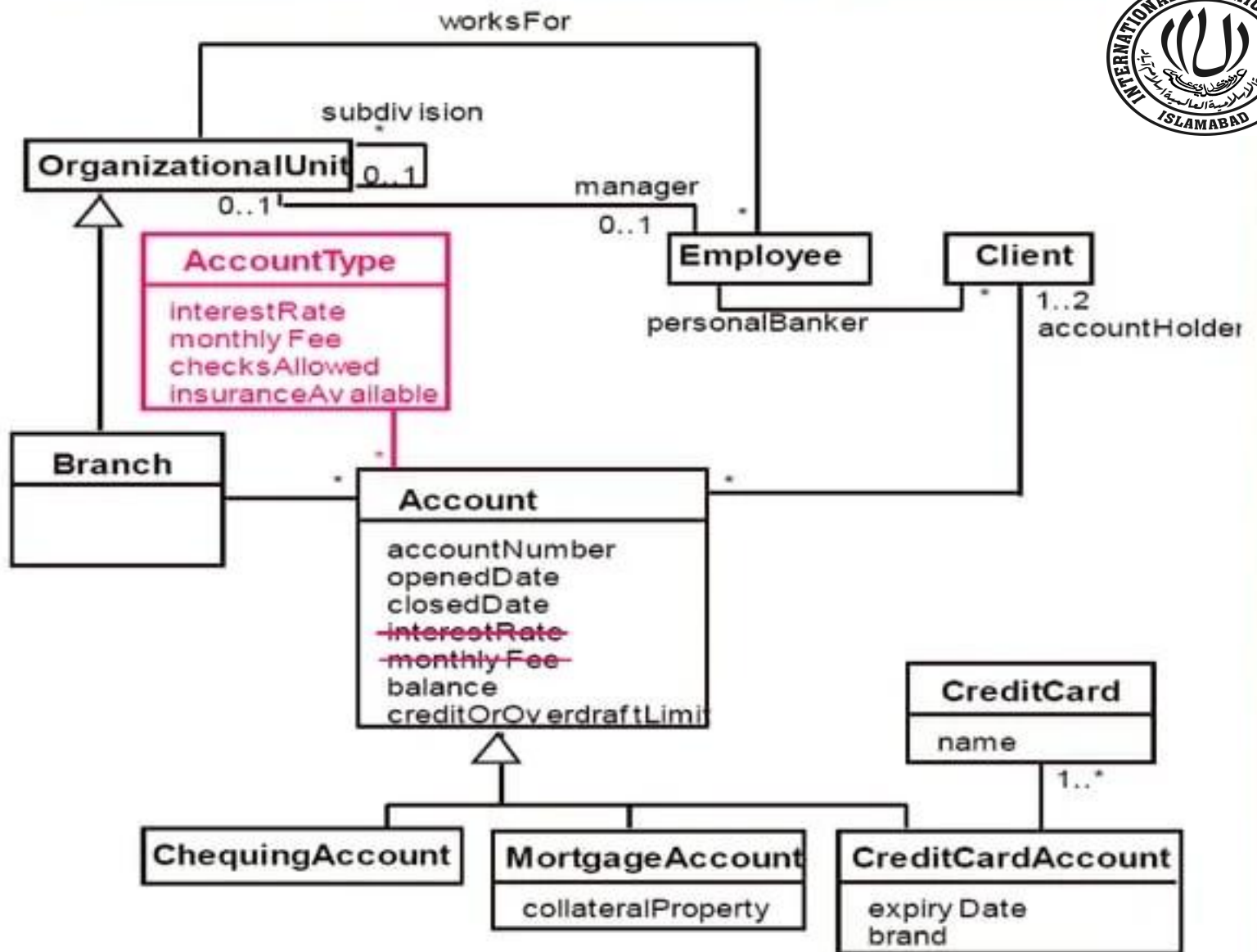


Looking at what we know about Branches (Classes, associations, attributes, generalizations)

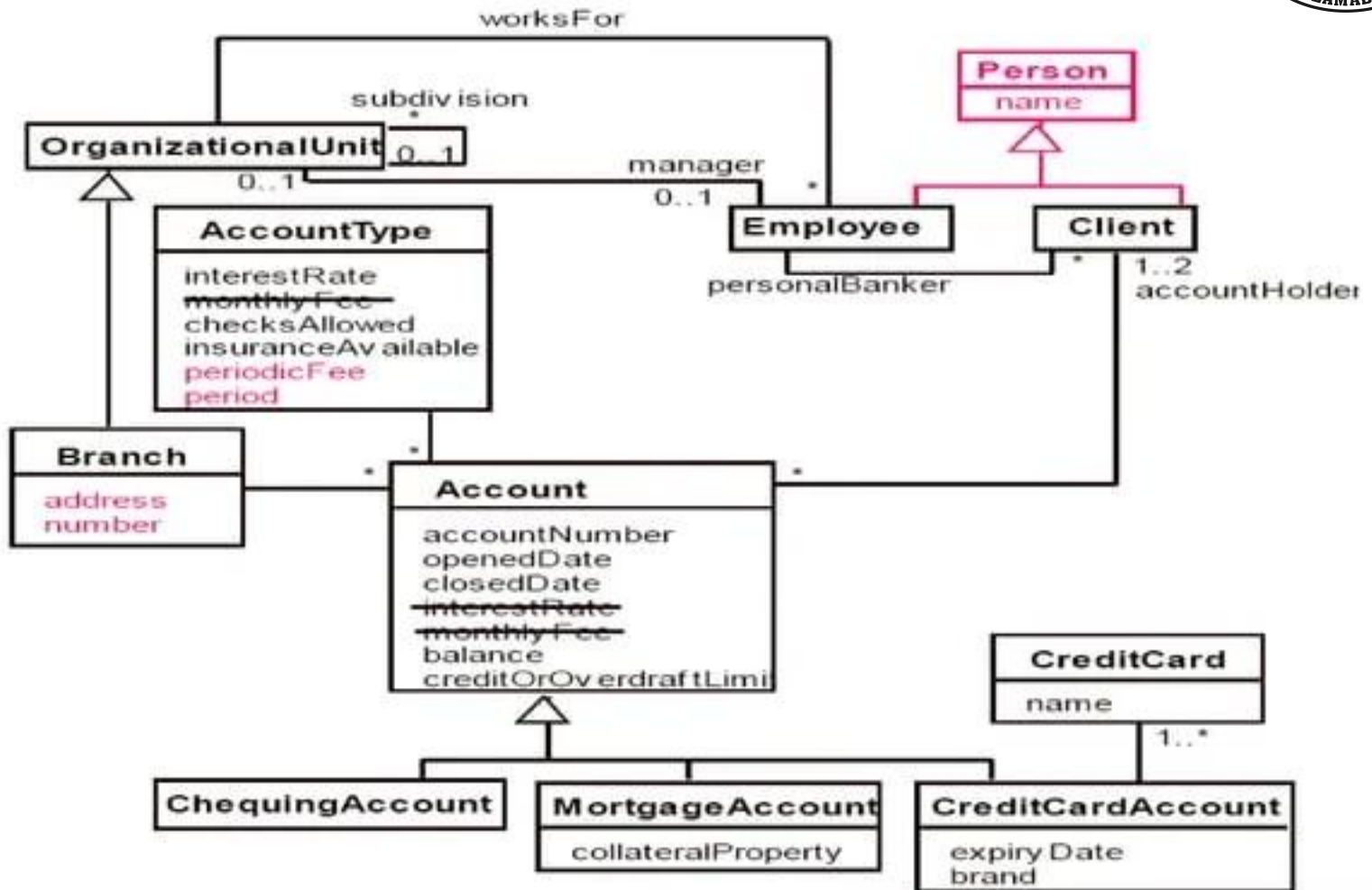
- This *system* provides the basic services to manage bank accounts at a bank called *OOBank*. OOBank has many branches, each of which has an address and *branch number*. A client opens accounts at a branch. Each account is uniquely identified by an account number; it has a balance and a credit or overdraft limit. There are many *types of accounts*, including: A mortgage account (which has a property as collateral), a chequing account, and a credit card account (which has an expiry date and can *have* secondary cards attached to it). It is possible to *have* a joint account (e.g. for a *husband* and *wife*). Each type of account has a particular *interest rate*, a *monthly fee* and a specific set of privileges (e.g. ability to write cheques, insurance for purchases etc. OOBank is divided into divisions and *subdivisions* (such as *Planning*, *Investments* and *Consumer*), the branches are considered subdivisions of the Consumer Division. Each *division* has a manager and a set of other employees. *Each customer is assigned* a particular employee as his or her ‘personal banker’.

DRAFT CLASS DIAGRAM WITH ORGANIZATIONAL UNIT





Bank Account System Class diagram with final touches



**Any Question
???**

Any Question