



SOFTWARE ENGINEERING

```
<!--BEGIN #primary .hfeed-->  
<div id="primary" class="hfeed">  
<?php if (have_posts()) : while (have_posts())  
|  
<?php zilla_page_before(); ?>  
<!--BEGIN .hentry-->  
<div <?php post_class() ?> id="post-?<?php  
<?php zilla_page_start(); ?>  
<div id="contact-boxes" class="c  
<div class="box support">  
<h2>Need Theme Sup  
<p>Get a helping  
<a href="<?php  
</div>  
<div class="box f  
<div class=  
<h2>Freque  
answ
```

SYSTEM MODELLING

Presented by :

Mohamad Nor Hassan

**Faculty of Informatics and Computing,
Universiti Malaysia Terengganu**

Email : mohamadnor@umt.edu.my

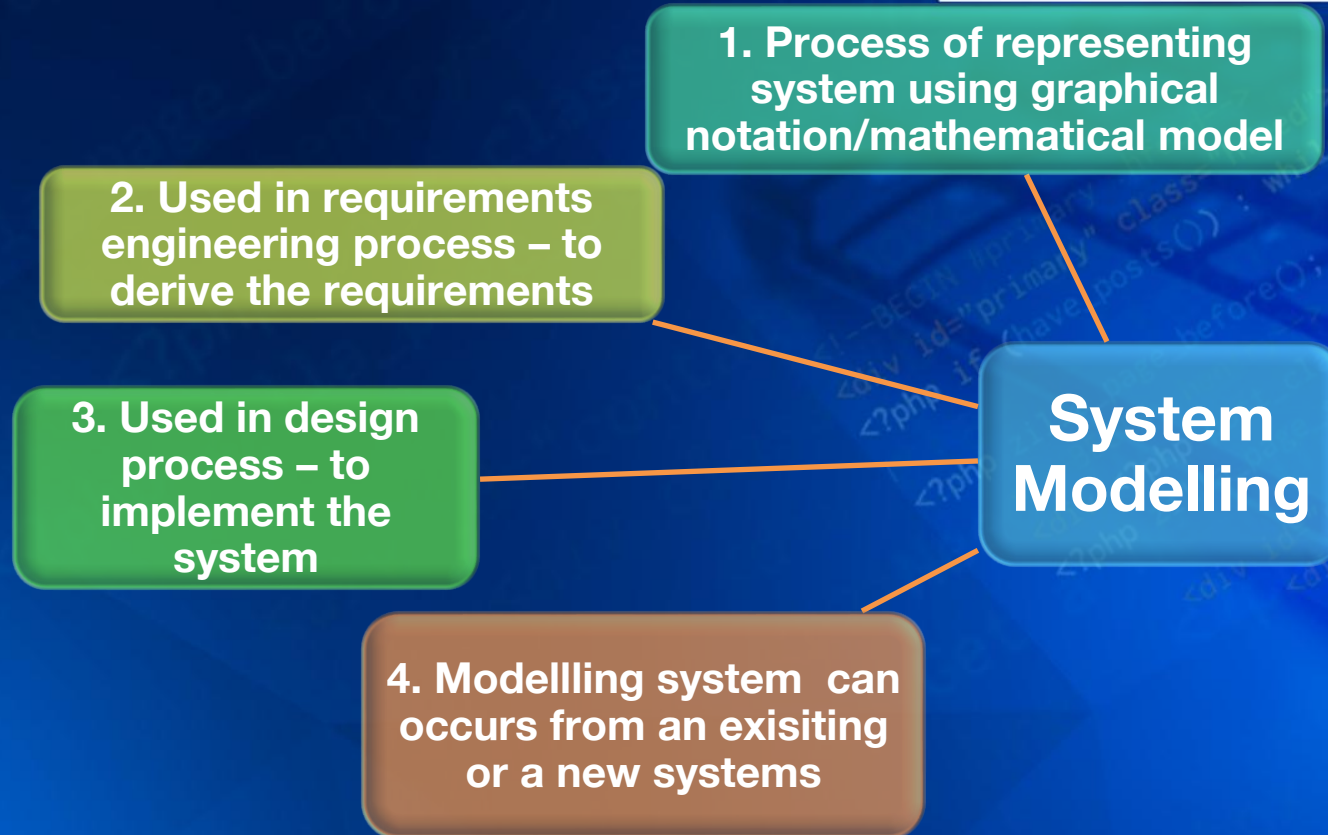
Learning Outcomes

The use of graphical models for representing the software systems as a whole using UML Notation.

Understand system modelling perspective from the context models, interaction models, structural models, and behavioural models.

How to model software systems via case study

Introduction



Type of Models

External perspective – model the content/environment of software system.

Interaction perspective – model interactions of system and its environment/ among components

Structural perspective – model structure of data of the software system.

Behavioural perspective – model dynamics behaviour of the system and reflect to an event.

Definition of UML

What is UML?

Is a set of diagrammatic techniques that being used to specify, visualize and document models of software system including their structure and design that meeting the requirements stated.
(OMG, 2015)

Why using UML?

Visualize an abstraction of each component of software system.

Impose a coherent structure of requirements gathered from clients.

UML Notation

Use case

- Show how the system interact with its users

Sequence diagram

- Show the interactions between objects, actors.

Activity diagram

- Show the sequence of activities that formed the process.

Class diagram

- Show data elements in the system and its relationships.

State diagram

- Representing dynamic model show how an objects change their states due to an events.

UML Symbols for Use Case

(System Modelling/External Perspective)

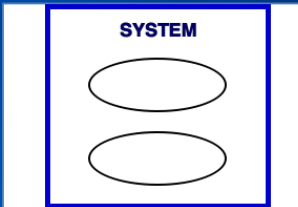


A use case
an ellipse label that represent major of system functionality.

An actor
an external entity that interact with the system.

An association relationship
- a link between an actor and a use case

System boundary
- represent the boundary of the system from an external actor and the system.



UML Symbols for Sequence Diagram

(Interaction Perspective)



anObject: aClass



An actor

– an external entity participate in a sequence of process via sending or receiving a message.

An object

– a representation of business object entity for software system that sending and receiving message.

A lifeline

– represent the life span of an object during a sequence.

UML Symbols for Sequence Diagram

(Interaction Perspective)



An execution occurrence

– denote when an object sending and receiving message.



A message

– convey information message denote when an object sending and receiving message.

UML Symbols for Activity Diagram

(Behavior Perspective)



An activities - represent the functionalities of the system derive from use case.



A flow – show the workflow of the process.



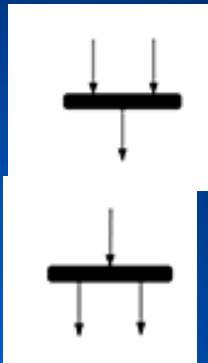
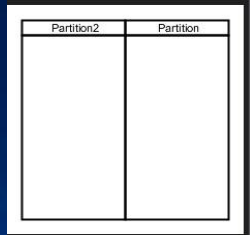
Start – the starting point of the process.



End – the end point of the process.

UML Symbols for Activity Diagram

(Behavior Perspective)



Swim lane Separator – show area of responsibilities for each activities.

Decision - accepts an input on an incoming process and presents them to multiple outgoing process.

A synchronization bar – a thick vertical/horizontal line that combines two concurrent processes and reintroduce them to one activity .

A fork – it splits a single process flow into two concurrent processes.

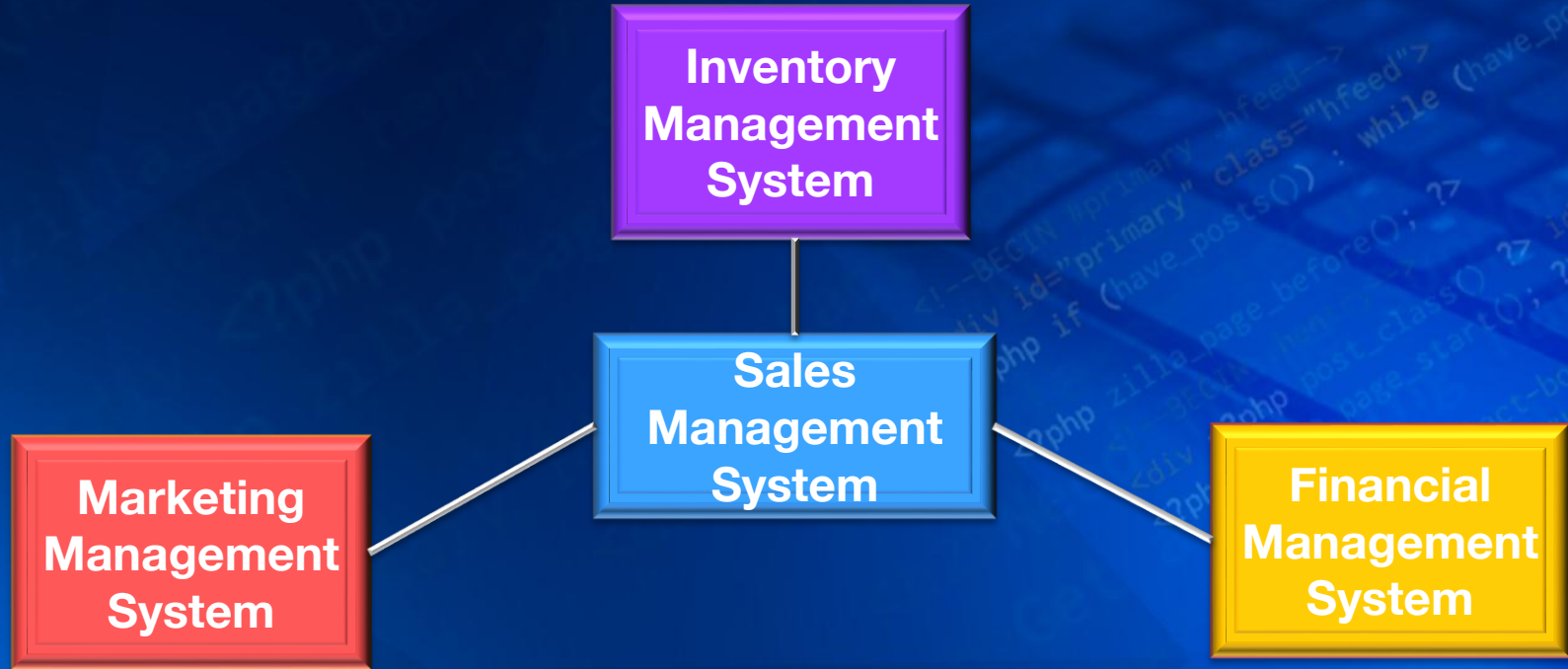
Context Model

This refer to the system boundaries of the software system.

The project team will work with stakeholder to identify functions include in the system and what provided by system's environment.

The context model will determine specifically what kind of requirements/specifications will be carried out by during requirement analysis.

Context Model



Interaction Models

All systems will interact in various ways to perform the task.

The interaction can be take place in several ways;

- **User inputs and outputs**
- **Interaction between the systems**
- **Interaction between the components of the system**

For interaction model, we focus on;

- **Use case modelling**
- **Sequence diagram**

Use Case Modelling

Use to support the process of requirement elicitation.

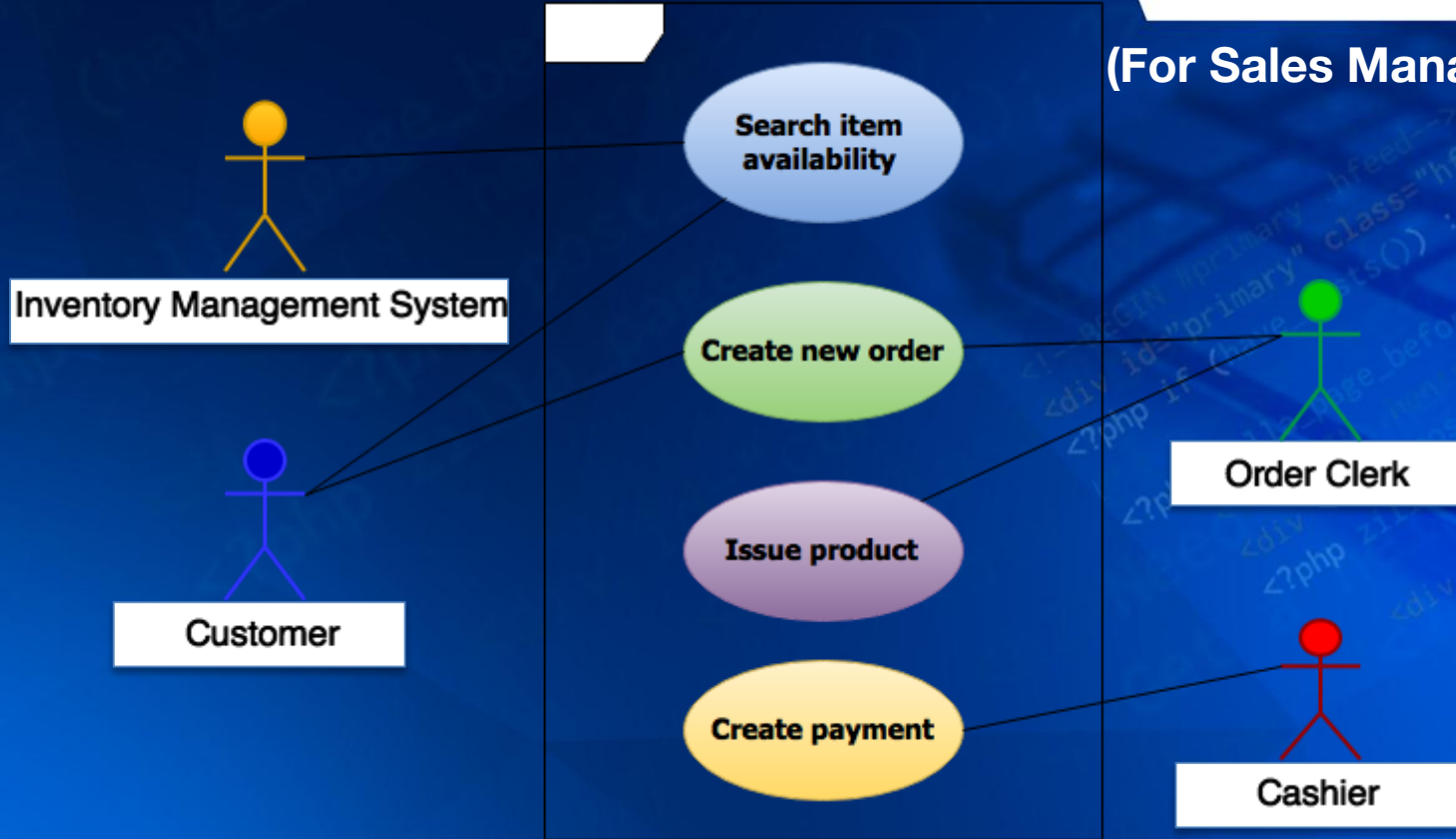
System analysts use the use case to present the functionality of the system at a very high level.

Inside the system boundary, is a part of the functionalities of the system.

Outside the system boundary, is an actors that act upon the system.

Use Case Diagram

(For Sales Management System)



Sequence Diagram

Use to model the interactions between the actors and the processes in a system.

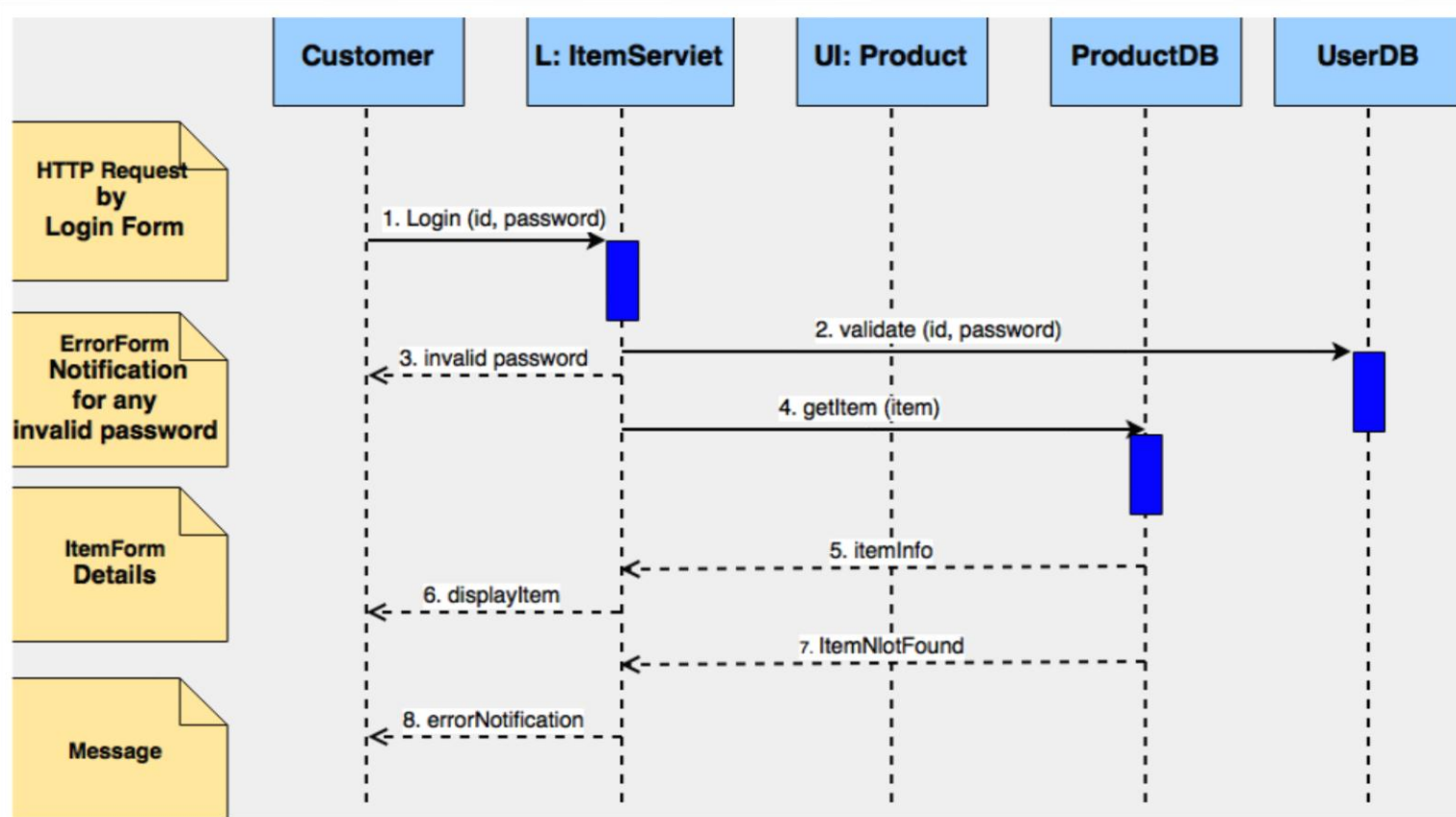
Sequence diagram show a sequence of interactions that take place for specific use case.

The objects and actors are displayed on top of the diagram with dotted line drawn vertically from these.

Interaction between objects are represented by annotated arrows.

The rectangle on dotted lines indicated the lifeline of the object.

Sequence diagram for Search Item Availability



Structural Models

Structural models represent the organization of components of the system and its relationships.

Can be divided into two categories;

- **Static model** – show the structure of system design.
- **Dynamics model** – show the structure of system when executing.

Structure model is created when designing system architecture.

For structural model, we focus on;

- **Class diagram**
- **Generalization/ Specialization**

Class Diagrams

The class diagram is used to show the classes of an objects for the system.

Objects representing a real world artifact such as customer, product, order, etc.

Class diagram can be represented in different levels;

- Represent the object with the class name in a box.
- Represent the object as a semantic data model.

The relationship between the two classes will be implemented using an association between the.

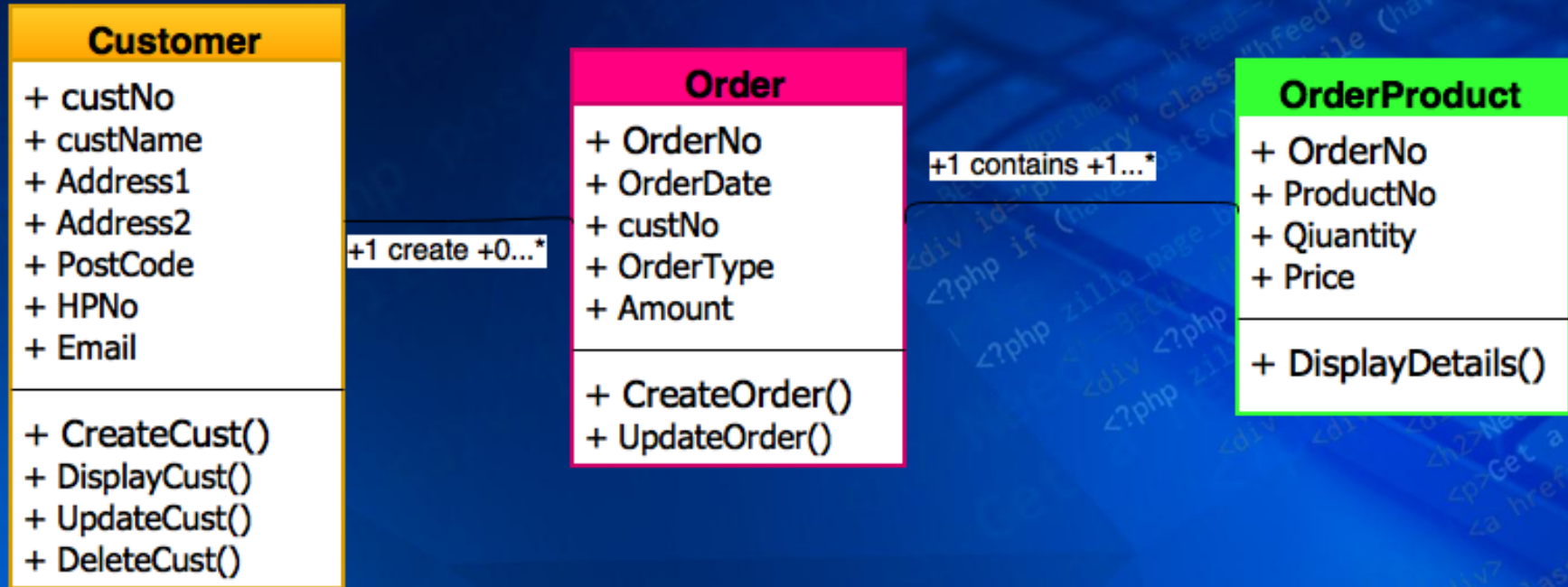
Class Diagrams

Simple class diagram with class name and relationship



Class Diagrams

Details class diagram with class name and relationship



Generalization

Generalization is the concept where people classify things based on similarities and differences.

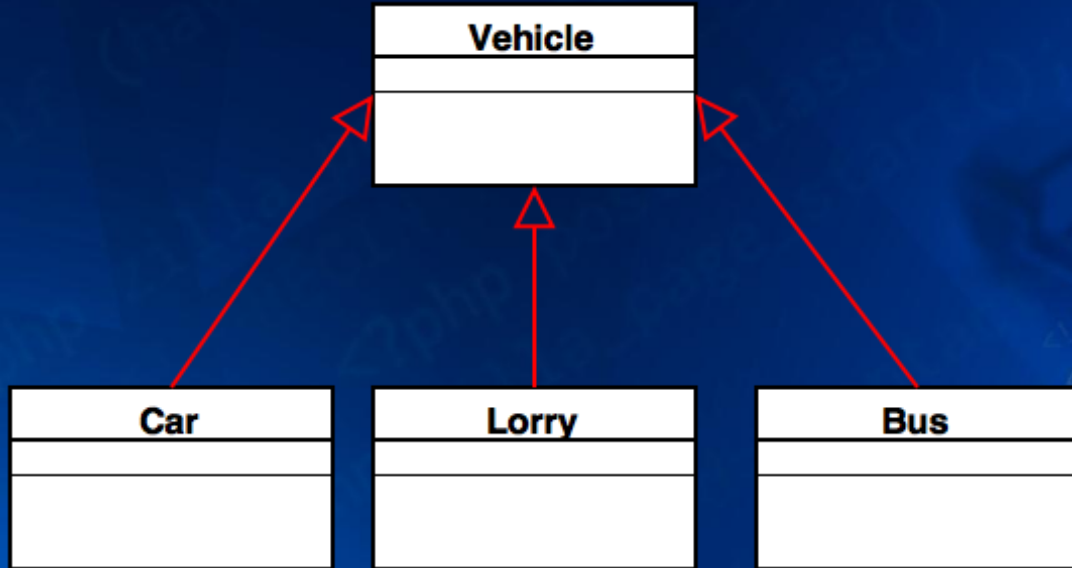
Generalization will group similar type of things together and to manage complexity.

For example, class vehicle consist of;

- **Car, Lorry, Bus**

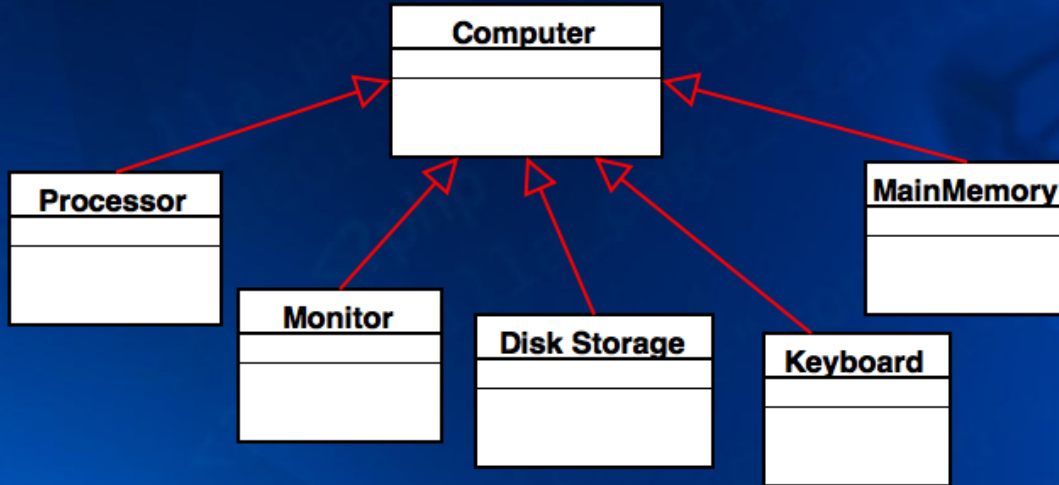
The purpose of generalisation is to structure common things from more general to more specific attributes.

Generalization



The UML has specific type of association that represent generalisation.

Aggregation





MINISTRY OF HIGHER EDUCATION



THANK YOU