# UNIT 2

## Object Oriented Analysis (OOA / Coad-Yourdon) and Object Oriented Design

# Object Oriented Analysis

In the system analysis or object-oriented analysis phase of software development, the system requirements are determined, the classes are identified and the relationships among classes are identified.

The three analysis techniques that are used in conjunction with each other for object-oriented analysis are object modelling, dynamic modelling, and functional modelling.

**Object Modelling**

Object modelling develops the static structure of the software system in terms of objects. It identifies the objects, the classes into which the objects can be grouped into and the relationships between the objects. It also identifies the main attributes and operations that characterize each class.

The process of object modelling can be visualized in the following steps −

- Identify objects and group into classes
- Identify the relationships among classes
- Create user object model diagram
- Define user object attributes
- Define the operations that should be performed on the classes
- Review glossary

**Dynamic Modelling**

After the static behavior of the system is analyzed, its behavior with respect to time and external changes needs to be examined. This is the purpose of dynamic modelling.

Dynamic Modelling can be defined as “a way of describing how an individual object responds to events, either internal events triggered by other objects, or external events triggered by the outside world”.

The process of dynamic modelling can be visualized in the following steps −

- Identify states of each object

- Identify events and analyze the applicability of actions
- Construct dynamic model diagram, comprising of state transition diagrams
- Express each state in terms of object attributes
- Validate the state–transition diagrams drawn

**Functional Modelling**

Functional Modelling is the final component of object-oriented analysis. The functional model shows the processes that are performed within an object and how the data changes as it moves between methods. It specifies the meaning of the operations of object modelling and the actions of dynamic modelling. The functional model corresponds to the data flow diagram of traditional structured analysis.

The process of functional modelling can be visualized in the following steps −

- Identify all the inputs and outputs
- Construct data flow diagrams showing functional dependencies
- State the purpose of each function
- Identify constraints
- Specify optimization criteria

Coad & Yourdon mention 7 key motivations and benefits in favor of OOA/OOD instead of using traditional analysis methods.

These motivations and benefits are:

* Tackle more challenging problem domains
* Improve analyst and problem domain expert interaction
* Increase the internal consistency across analysis, design and programming
* Explicitly represent commonality between classes and objects
* Build specifications resilient to change
* Reuse OOA, OOD and OOP results
* Provide a consistent underlying representation for analysis, design and programming

According to Coad & Yourdon the main result of Object Oriented Analysis and Design (OOA/OOD) comes from a reduction of complexity of a problem and the system's responsibilities within it. The OOA/OOD method is based on a number of general principles for managing complexity. According to Coad & Yourdon the Object Oriented approach is the only way to provide all these principles.

The Object Oriented approach is based on a uniform underlying representation (Classes and Objects) which according to Coad & Yourdon leads to a number of major implications:

* No major difference between analysis and design notations
* No 'transition' from analysis to design
* No waterfall model has to be followed, spiral and incremental are also possible
* There are still different skills and strategies needed for analysts and designers
* There is a uniform representation from OOA to OOD to OOP (OO programming)

# Object Oriented Design

Object-oriented design **(OOD)** is the process of creating a software system or application utilizing an object-oriented paradigm. This technique permits the creation of a software solution based on object notion. OOD is an implementation of the **object-oriented programming** (OOP) paradigm.

## What is Object Oriented Design?

In the object-oriented design method, the system is considered a **collection of objects** (i.e., entities). The state is shared among the objects, and each object is responsible for its own state data. Tasks designed for a specific purpose cannot refer to or update data from **other objects**. Objects have internal data that represents their current state. Similar objects form a class. In other words, **every object** belongs to a class.

- **Objects:** Objects are all the entities involved in the solution design. Persons, banks, companies, and users are all examples of objects. Every object has some properties associated with it, along with some methods for performing operations on those attributes.
- **Class:** Classes are generic descriptions of objects. An object is a class instance. A class defines all the properties an object can have and the methods that represent the object's functionality.
- **Messages:** Objects communicate by passing messages. Messages contain the target object's integrity, the name of the requested operation, and any other action required to complete the function. Messages are frequently implemented through procedure or function calls.
- **Abstraction:** Abstraction is used in object-oriented design to deal with complexity. Abstraction is the removal of the unnecessary and the amplification of the necessary.
- **Encapsulation:** It is also known as information concealing. The processes and data are tied to a single unit. Encapsulation not only groups together an object's vital information but also restricts access to the data and operations from the outside world.
- **Inheritance:** OOD allows similar classes to be stacked hierarchically, with lower or sub-classes being able to import, implement, and reuse variables and functions from their immediate superclasses. This OOD characteristic is known as inheritance. This facilitates the definition of specialized classes as well as the creation of generic classes.
- **Polymorphism:** OOD languages give a technique for assigning the same name to methods that perform similar functions but differ in arguments. This is referred to as polymorphism, and it allows a single interface to perform functions for multiple types. The relevant piece of the code is run depending on how the service is invoked.

## Stages of Object-Oriented Design

The object-oriented design process includes two main stages:

1. **System design**
2. **Object design**.

### System Design

The entire architecture of the intended system is designed at this stage. The system is envisioned as a collection ofinteracting subsystems, each comprising a hierarchy of interacting objects classified into classes. The system analysis model and the proposed system architecture are used to design the system. The emphasis here is on the system's objects rather than the system's processes.

### Object Design

A design model is created in this phase based on the models created in the system analysis phase and the architecture built in the system design phase. All of the requisite classes have been recognized. The relationships between the specified classes are established, and class hierarchies are identified. In addition, the developer creates the internal details of the classes and their relationships, such as the data structure for each attribute and the algorithms for the operations.

## Design of Object

The object design process comprises the following tasks:

- Object recognition
- Object representation, or the creation of design models
- Operation classification
- Design of algorithms
- Relationship design
- Control implementation for external interactions
- Modularize classes and connections.

## Relationships Designing

During the object design phase, the strategy for implementing the relationships must be developed. **Associations**, **aggregations**, and **inheritances** are some of the common relationships. The designer should address things like identifying whether an association is unidirectional or bidirectional, etc.

## Packaging Classes in OOD

A package is a namespace that organizes a set of related classes and interfaces. The thorough partitioning of an implementation into modules or packages is critical in any major project. Classes and objects are bundled into packages during object design to allow several groups to collaborate on

a project.

## Optimizing the Design

Before implementing a design, it should be optimized to make the implementation more efficient. Optimization aims to reduce costs in time, space, and other metrics. However, excessive design optimization should be avoided because ease of implementation, **maintainability**, and extensibility are also significant considerations. A perfectly optimized design is often more efficient but less understandable and reusable. As a result, the designer must find a happy middle ground between the two.

## Documentation of Design

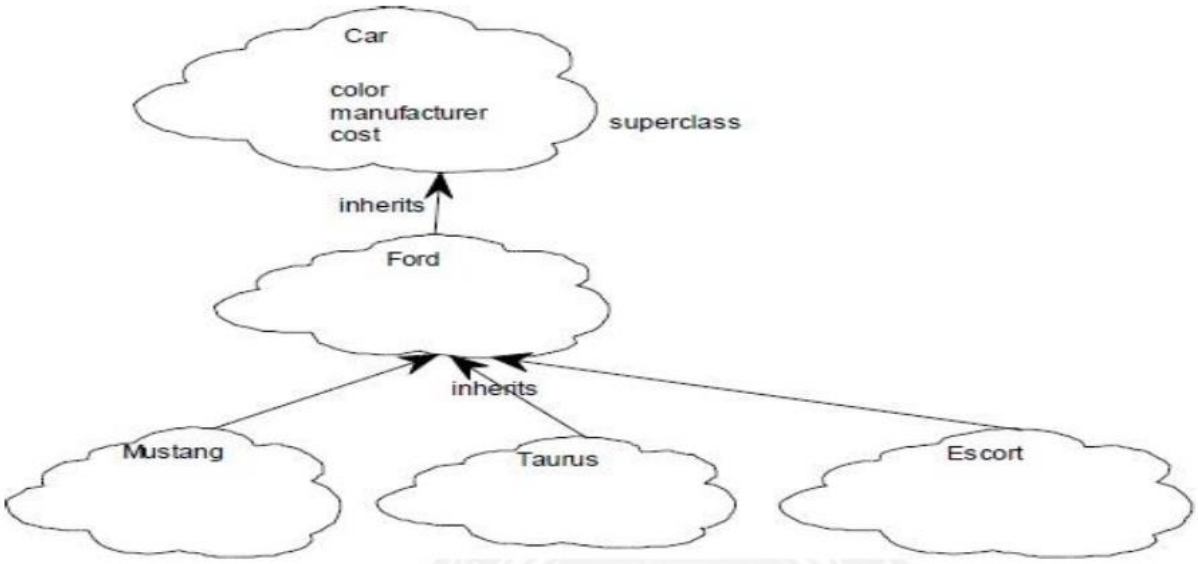
Documentation is an essential aspect of any software development process since it records the steps involved in creating the product. For any non-trivial software system, design decisions must be documented in order to be transmitted to others.

# The Booch Methodology

•It is a widely used object oriented method that helps us to design the system using object paradigm.
• The Booch methodology covers the analysis and design phases of systems development.
• Booch sometimes is criticized for his large set of symbols.
• You start with class and object diagram in the analysis phase and refine these diagrams in various steps.

The Booch method consists of the following diagrams:
– Class diagrams
– Object diagrams
– State transition diagrams
– Module diagrams
– Process diagrams
– Interaction diagrams

**Object Modeling using Booch Notation**




The Booch methodology prescribes
– A macro development process serve as a controlling framework for the micro process and can take weeks or even months. The primary concern of the macro process is technical management of the system
– A micro development process.

## Macro development process

The macro development process consists of the following steps:

***1. Conceptualization :***

- you establish the core requirements of the system
- You establish a set of goals and develop a prototype to prove the concept

***2. Analysis and development of the model.***

Use the class diagram to describe the roles and responsibilities objects are to carry out in performing the desired behavior of the system .Also use the Object diagram to the the desired behavior of the system in terms of scenarios or use the interaction diagram.

***3. Design or create the system architecture***.

In this phase, You use the class diagram to decide what class exist and how they relate to each other .Object diagram to used to regulate how objects collaborate. Then use module diagram to map out where each class and object should be declared. Process diagram – determine to which processor to allocate a process.

***4. Evolution or implementation***. – refine the system through many iterations

***5. Maintenance***. - make localized changes the the system to add new requirements and eliminate bugs.

## Micro Development Process

Each macro development process has its own micro development process

• The micro process is a description of the day to- day activities by a single or small group of
s/w developers

• The micro development process consists of the following steps:

1.Identify classes and objects.
2. Identify class and object semantics.
3. Identify class and object relationships.
4. Identify class and object interfaces and implementation.