



## UNIT-I-Part-III

**Object oriented thinking:** A way of viewing world – Agents, responsibility, messages, methods, Classes and instances, class hierarchies – inheritance, method binding, overriding and exceptions.

### Object Oriented Thinking:

Way of thinking where we consider each component of a problem in terms of object.

Here we break down the problem into different components and think each component as an object.

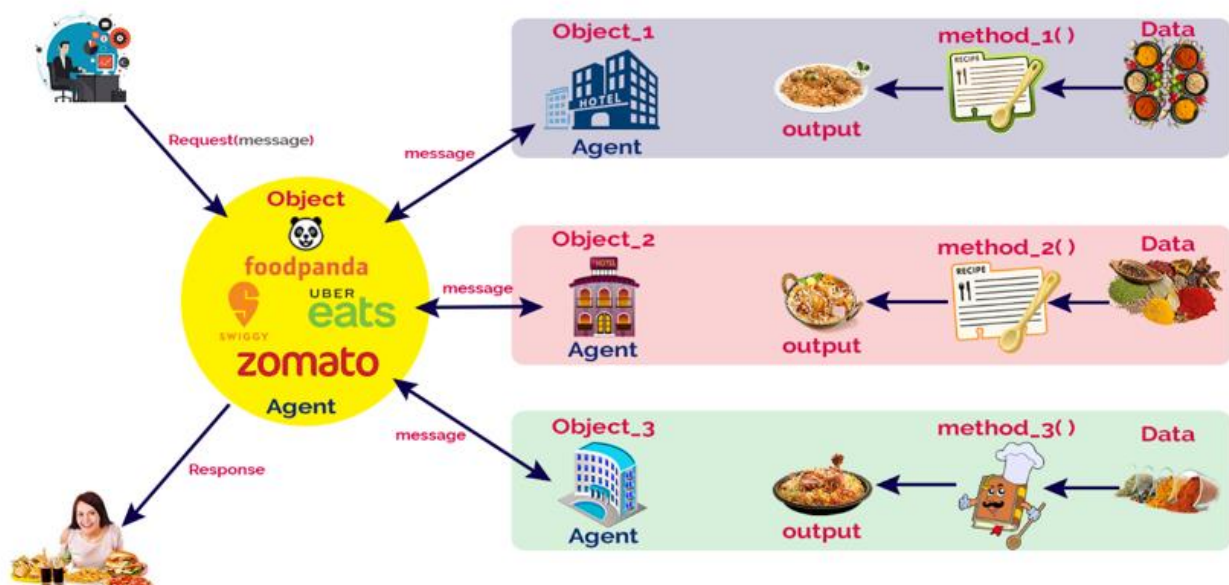
### A way of viewing world: -

A way of viewing the world is an idea to illustrate the object-oriented programming concept with an example of a real-world situation.

Let us consider a situation, I am at my office and I wish to get food to my family members who are at my home from a hotel. Because of the distance from my office to home, there is no possibility of getting food from a hotel myself. So, how do we solve the issue?

To solve the problem, let me call zomato (an **agent** in food delivery community), tell them the variety and quantity of food and the hotel name from which I wish to deliver the food to my family members. Look at the following image.

## A way of viewing world with OOP



## Agents and Communities

To solve my food delivery problem, I used a solution by finding an appropriate agent (Zomato) and pass a message containing my request. It is the responsibility of the agent (Zomato) to satisfy my request. Here, the agent uses some method to do this. I do not need to know the method that the agent has used to solve my request. This is usually hidden from me.

So, in object-oriented programming, problem-solving is the solution to our problem which requires the help of many individuals in the community. We may describe agents and communities as follows.

**An object-oriented program is structured as a community of interacting agents, called objects. Where each object provides a service (data and methods) that is used by other members of the community.**

In our example, the online food delivery system is a community in which the agents are zomato and set of hotels. Each hotel provides a variety of services that can be used by other members like zomato, myself, and my family in the community.

## Messages and Methods

To solve my problem, I started with a request to the agent zomato, which led to still more requests among the members of the community until my request has done. Here, the members of a community interact with one another by making requests until the problem has satisfied.

**In object-oriented programming, every action is initiated by passing a message to an agent (object), which is responsible for the action. The receiver is the object to whom the message was sent. In response to the message, the receiver performs some method to carry out the request. Every message may include any additional information as arguments.**

In our example, I send a request to zomato with a message that contains food items, the quantity of food, and the hotel details. The receiver uses a method to food get delivered to my home.

## Responsibilities

In object-oriented programming, behaviors of an object described in terms of responsibilities.

In our example, my request for action indicates only the desired outcome (food delivered to my family).

The agent (zomato) free to use any technique that solves my problem.

By discussing a problem in terms of responsibilities increases the level of abstraction. This enables more independence between the objects in solving complex problems.

## Classes and Instances

In object-oriented programming, all objects are instances of a class.

The method invoked by an object in response to a message is decided by the class. All the objects of a class use the same method in response to a similar message.

In our example, the zomato a class and all the hotels are sub-classes of it.

For every request (message), the class creates an instance of it and uses a suitable method to solve the problem.

## Classes Hierarchies

A graphical representation is often used to illustrate the relationships among the classes (objects) of a community.

This graphical representation shows classes listed in a hierarchical tree-like structure. In this more abstract class listed near the top of the tree, and more specific classes in the middle of the tree, and the individuals listed near the bottom.

**In object-oriented programming, classes can be organized into a hierarchical inheritance structure.**

**A child class inherits properties from the parent class that higher in the tree.**

## Method Binding, Overriding, and Exception

In the class hierarchy, both parent and child classes may have the same method which implemented individually.

Here, the implementation of the parent is overridden by the child. Or a class may provide multiple definitions to a single method to work with different arguments (overloading).

The search for the method to invoke in response to a request (message) begins with the class of this receiver.

If no suitable method is found, the search is performed in the parent class of it.

The search continues up the parent class chain until either a suitable method is found or the parent class chain is exhausted.

If a suitable method is found, the method is executed. Otherwise, an error message is issued.