

# CHAPTER 9

## OBJECT-ORIENTED CONCEPT AND PRINCIPLES

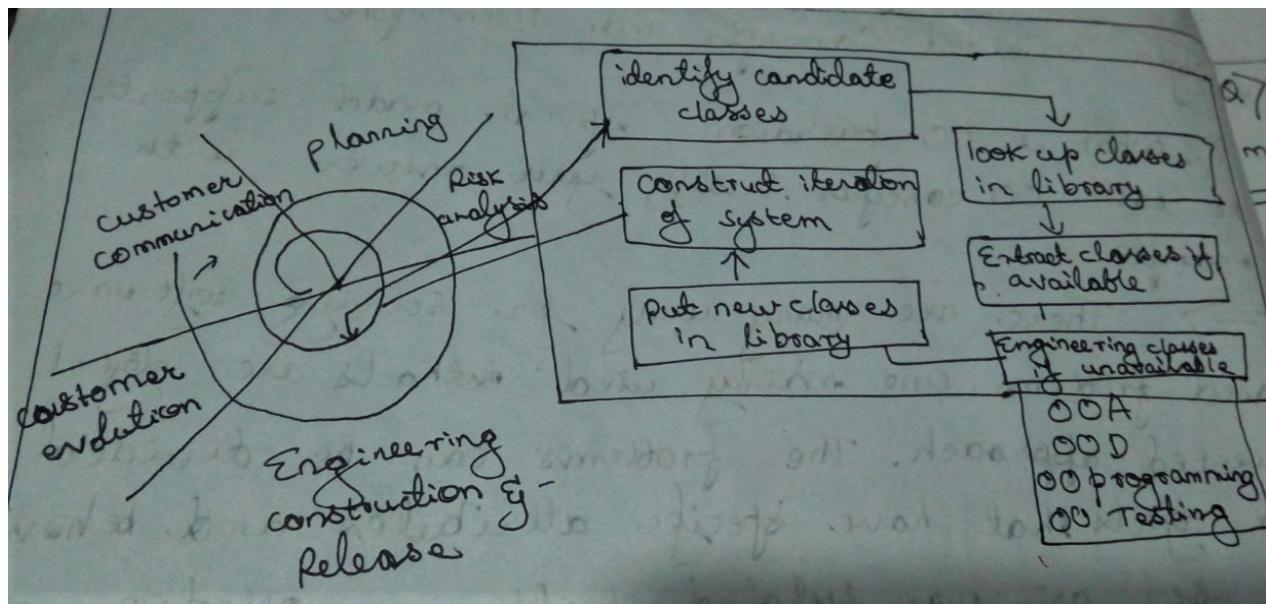
### ❖ OBJECT ORIENTED PARADIGM

**Q. What is OO paradigm? Spiral model supports the basic OO concept. Justify your answer with an example.**

There are many ways for solving software based problems. One mainly used method is object oriented approach. The problems can be divided into objects that have specific attributes and behavior. The objects are manipulated with a collection of function (methods, operations) and communicate with another through a message.

OO paradigm encompasses a complete view of software engineering. Paradigm is a way of conceptualizing what it means perform computation and how task to be carried out on a computation and how task to be carried out on a computer should be structured and organized. Software engineer and manager must consider such items as OO required analysis, OO Domain (OOD), OO Database and OO Computer Aided s/w engineering.

OO systems are designed using an evolutionary model. An evolutionary model make reuse of components in the best paradigm, for OO s/w engineering. Figure below shows the component based development process model change for OO s/w engineering.



The OO process moves through an evolutionary spiral that starts with customer communication. Planning and risk analysis establish for the OO project plan. The technical work of OO s/w engineering follows the iterative path shown in the box. Classes are looked up in library, the s/w engineering applies OOA, OOD, OO programming and OO testing to create the class and object is derived from the class. The new class is then put into the library so that it may be reused in the future.

❖ **OBJECT ORIENTED CONCEPT/ IDENTIFYING THE ELEMENTS OF AN OBJECT MODEL:**

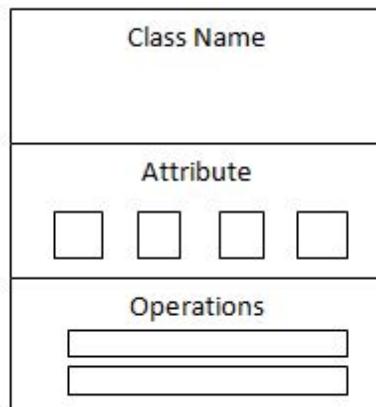
**Q. List out the different elements of an object model with suitable example.**

**1. OBJECT:**

Objects are runtime entities in an OO system. They may represent person, place or anything and also user defined data such as vector, time etc. Object should be chosen such that they represent real life entities. The characteristics of an object are represented by data and its behavior is represented by function.

**2. CLASSES:**

A class is a group of objects that share common properties and relationships. The entire set of data and code of an object can be made a user defined type with the help of a class. All objects exist within a class, inherit its attributes and the operations that are available to manipulate the attributes. A super class is a collection of classes and a sub class is an instance of a class.



**3. ATTRIBUTES (Data):**

It is used to describe the class or object in some way.

**4. OPERATIONS, METHODS AND SERVICES:**

An object represented as a collection of attributes and the algorithm that process the data. These algorithms are called operations, methods and services. Eg: The operation Getcolor for the object automobile will extract the color stored in the color attribute.

**5. MESSAGES**

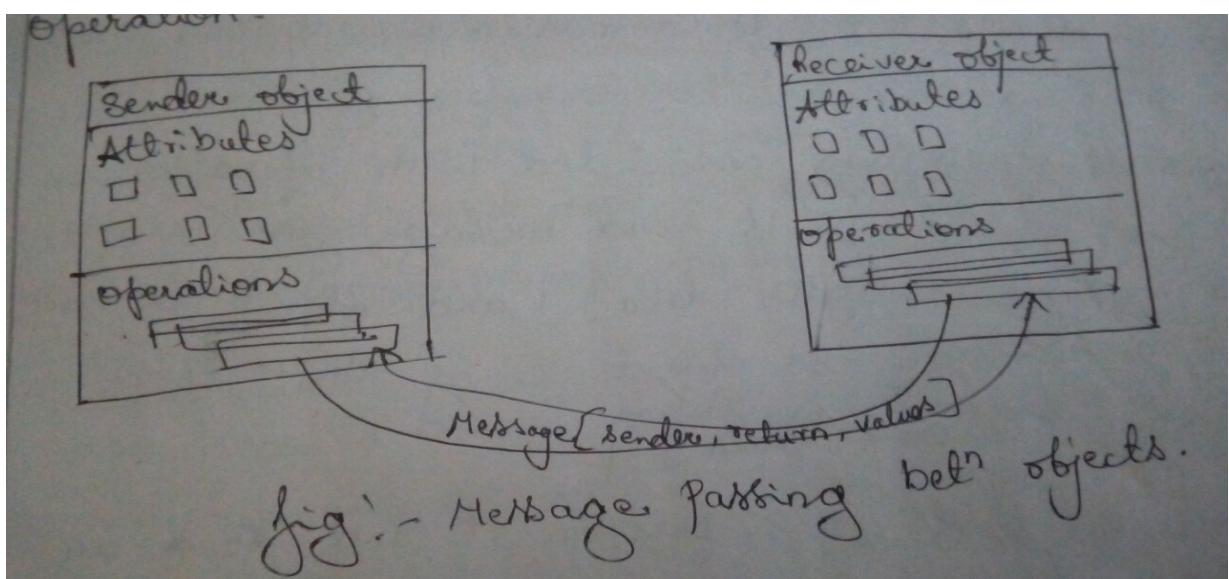
Messages are the means by which objects interact. An OOP consists of a set of objects interact. An OOP consists of a set of objects that communicate with each

other by sending and receiving information same as people pass message to one another. The message passing process can be performed in following ways:

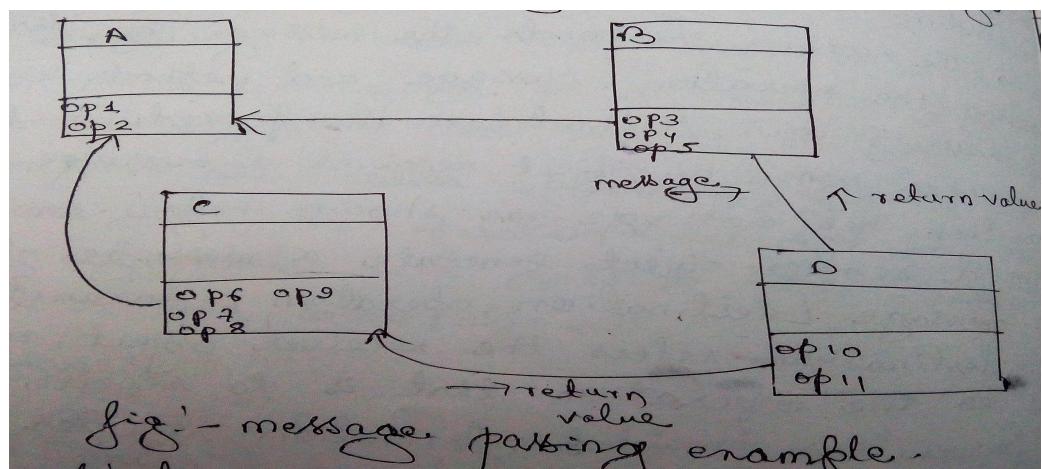
- Creating classes that defines object and their behavior.
- Creating object from class definitions.
- Establishing communication among objects.

An object is requested to perform one of its operations by sending it a message telling the object what to do. The receiver responds the message by first choosing the operation. Message and methods are two sides of coin.

Methods are the procedure that is invoked when an object receives a message. The interaction between messages is as shown below. An operation within a sender object, generate a message in the form [destination, operation, parameters], where destination refers to the operation that is to receive a message and parameters provide information required for the operation.



Consider the figure below that shows message passing within OO system.



Four objects A, B, C, D communicate with one another by passing message. For example if object B requires processing associated with operation op10 of object D, it sends message for D in the form [D, op10, {data}] and object D may send a message to object C in the form [C, op8, {data}].

Then C finds op8, performs it and send an appropriate return value to D. op10 completes and sends a return value to B.

## 6. ENCAPSULATION, INHERITANCE, POLYMORPHISM

### Encapsulation:

It is the most striking features of a class. The wrapping of data and function into a single class (unit) is known as encapsulation. In OOP data is not accessible to the outside program, so only these functions which are wrapped in that class can access it. This provides a number of important benefits.

It provides the way of information hiding i.e this reduce the side effect, when change occur.

It helps for components reuse.

An object that sends a message need not described the detail of internal data structures. Hence, interfacing is simplified and system coupling reduced.

Example: When using a “switch board” we need to known only its essential features that are what all switches are pressed and the background details like wiring and connection are hidden from us.

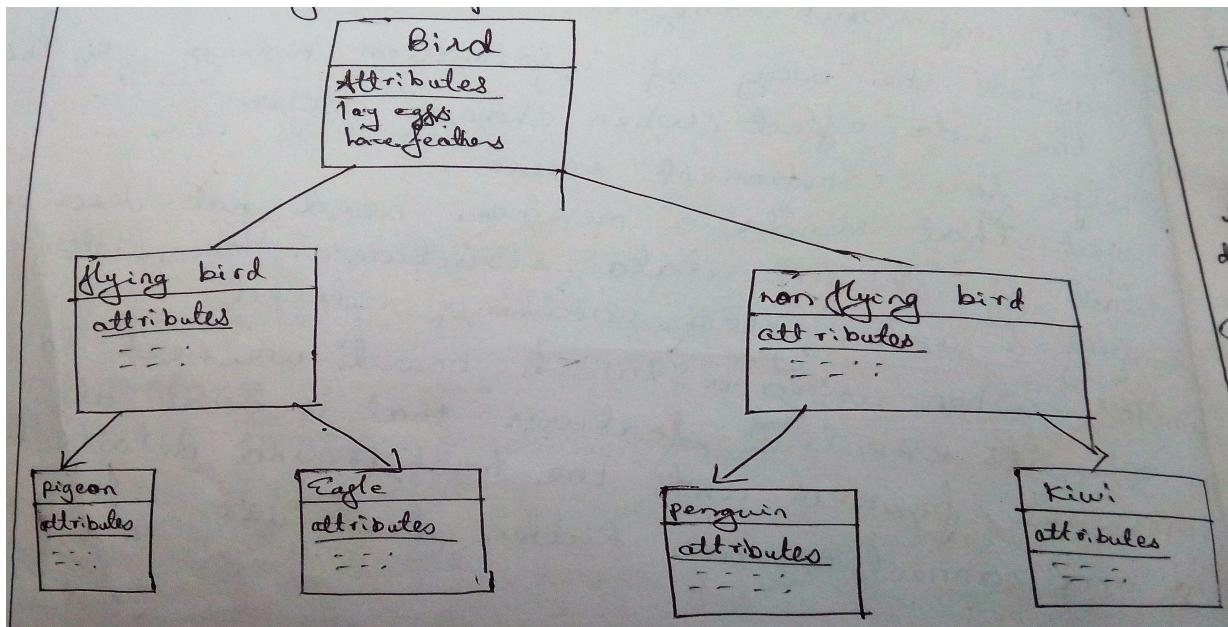
### Inheritance:

Inheritance is one of the key differentiators between conventional and OO systems. Inheritance provides a way of reuse subclasses to its super classes data and procedure.

It is the process by which objects of one class acquire the properties of objects of another class. It supports the hierarchical classification. In OOP, this concepts provides the idea of reusability.

A subclass Y inherits all of the attributes and operations associated with its subclass X. This means all the data structure and algorithm design and implemented for X are immediately available for Y, no further work need to be done.

Any changes to the data or operation of a super class are immediately inherited by all subclass that have inherited from super class.



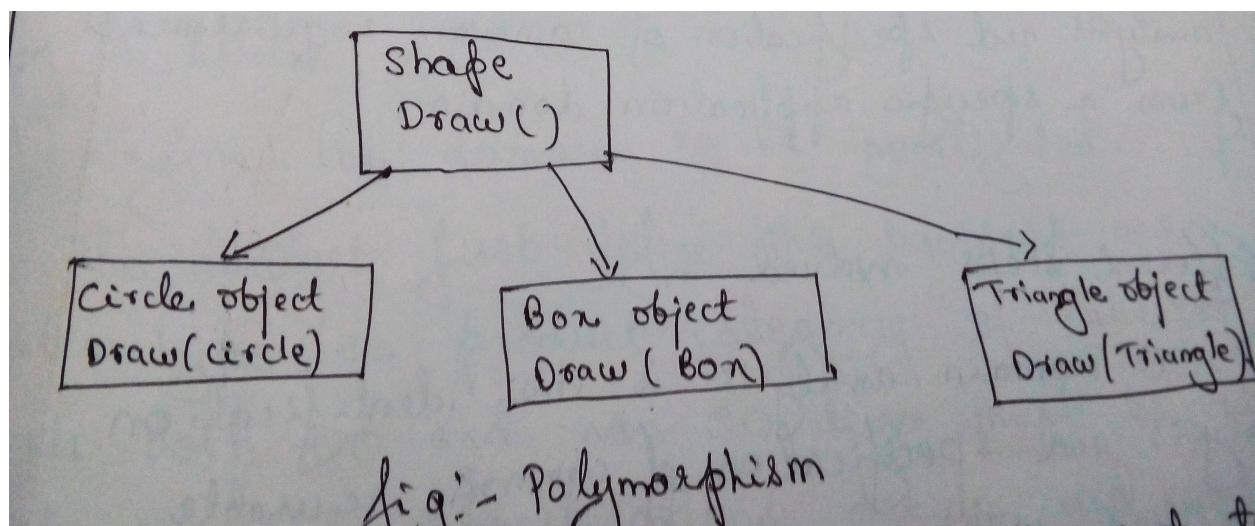
Hence, pigeon is a part of class “flying bird” which is again a part of class “Bird”.

### Polymorphism:

In Greek terms, it means the ability to take more than one form. Polymorphism is a mechanism that allows several objects in a class hierarchy to have different methods with the same name.

The main benefit of polymorphism is that it simplifies the programming interface. It supports the process of operation or operator overloading.

A single function name Draw () can be used to handle different numbers and types of argument using a single function name draw.



## ❖ MANAGEMENT OF OBJECT ORIENTED SOFTWARE PROJECTS

Modern software project management can be sub divided into:

- Establishing common process framework for a project.
- Using the framework and metrics to develop effort and time estimates.
- Establishing deliverables and milestones that will enable progress to be measured.
- Defining check points for risk management, quality assurance and control.
- Managing the changes that invariably occur as the project progresses.
- Tracking, monitoring and controlling progress.

A common process framework defines an organization approach to s/w engineering. It identifies the paradigm that us applied to build and maintain software and the tasks, milestones and deliverables that will be required.

Object Oriented Metrics are used to estimate the cost, time of the developing process.

