# **BASIC OOP CONCEPTS**

## **Objects**

An object represent an entity in real world and each object has its own properties that describe what it is orders.

Now question arises what kind of things become object in OOPS the answer to this is limited only by your imagination.

like physical objects (Automobiles, Aircraft, Electrical Components) Programming construct (Arrays, Stacks, Linked list, Binary-tree) User defined data type (Time, Angle, Complex number).

	Person Object	Car Object
Data	Name	Model
	age	Colour
	weight	Year
Function	Actions	Action
	Walk	Start
	Sleep	Stop
	Talk	Accelerate

### Classes

The entire set of data and code of an object can be made a user-defined data type with the help of a class. Infact, objects are variables of type class. Once a class has been defined, we can create any number of objects belonging to that class.

A class serves as a plan or template. It specifies what data and what functions will be included in objects of that class. A class is thus a collection of objects of similar type. Defining the class doesn't create any variables.

e.g: Mango, apple, orange are members of the class fruit.

e.g: Polygon class.

Properties  $\longrightarrow$  Ch. of object vertices fill color border color  $\longrightarrow$  All objects of polygon has border color  $\longrightarrow$  Method: draw erase Move

## **Properties**

Actually class is a prototype, not an actual specimen entity. Each object has its own value for each of its properties but it shares the properties names or operations with other instances of the class.

**Method:** Each action of an object becomes a function in class that defines it and is referred to as a method.

### Abstraction

Abstraction is the process of examining certain aspect of a problem, without including the details or explanation. An abstract specification tells us that what an object does independent of how it works i.e. focus is on what an object is and does before deciding how it should be implemented. So OOPS implementing data abstraction in a clean way by using class concept. So class is known as Abstract Data type.

So data abstraction is used to identify properties and method of each object as relevant to the application. And this wrapping up of data and function into a single unit is known as encapsulation.

The data is not accessible to the outside world and only those functions which are wrapped in the class can access it. This insulation of the data from direct access by the program is called data hiding.

#### Inheritance

Inheritance is the property by which objects of one class acquire the properties of objects of another class.

It allows the reuse of an existing class to build a new class.

e.g: Class of animals can be divided into mammals, insects, reptiles etc.

The principle in this sort of division is that each subclass shares common properties with the class from which it is derived. e.g. off vehicles in a class can share similar properties of having wheels and a motor, while subclass bus may have its own characteristic like seats for people, while trucks may have space for carrying goods.

Thus this new class i.e. subclass inherits all the behaviour of the original class (parent class). Here subclass is said to be a speicalization of its superclass and conversely a superclass-generalization of its sub-classes.

#### Reusability

In OOP, the concept of inheritance provides the idea of reusability. Once a class has been written, created and debugged it can be distributed to other programmers for use in their own programs. This is called Reusability.

So, this is very similar to the way a library of functions in a procedural language can be incorporated

into different programs.

## Polynorphism

Poly (many) + morphism (form) [i.e. one thing with several distinct forms]

**e.g.** an operation may exhibit different behaviour in different instances. The behaviour depends upon the types of data used in operation. Say operation of addition for two number it will generate sum, while for strings, the operation would produce a third string by concatenation.

This is similar to a particular word having several different meanings depending on the context.

Say a single function name can be used to handle different number and different types of arguments.



### Extensibility

One of the difficulty of traditional languages is the difficulty of creating new data types. Computer language typically have several built in data types. Integers, float, char; Perhaps if we want to work with complex number. 2-D co-ordinates or dates, there quantities can't handle easily by built in type.

Being able to create your own types is called extensibility i.e. you can extend the capabilities of a language. Traditional languages are not usually extensible.

So here one of the benefits of objects is that they give programmer a way to construct new data type.

Position 1 = Position 2 + Origin

Object = oriented languages

The language should support several of the OOP concepts to claim that they are object oriented. Depending upon the features they support, they can be classified into the following two categories:

(a) **Object based programming language:** Language that supports programming with objects are said to be object based programming language. They don't support Inheritance and

dynamic binding.

e.g. Ada

(b) **Object Oriented Programming Language:** Object based features + Inheritance + dynamic binding C++, Smalltalk and Object Pascal, Java Eiffel.

Pure object oriented programming language: Smalltalk

Dynamic Binding means that the code associated with a given procedure call is not known until the time of call at run-time. It is associated with polymorphism and inheritance.

Binding refers to the linking of a procedure call to the code to be executed in response to the call.