OBJECT-ORIENTED PROGRAMMING

Object-oriented programming (OOP) is a programming paradigm based on the concept of“objects”, which may contain data, in the form of fields, often known as attributes; and code,in the form of procedures, often known as methods.

List of object-oriented programming languages

Ada 95	Fortran 2003	PHP since v4,
BETA	Graphtalk	Python
C++	IDLscript	Ruby
C#	J#	Scala
COBOL	Java	Simula
Cobra	LISP	Smalltalk
ColdFusion	Objective-C	Tcl
Common Lisp	Perl since v5	

Abstraction

Abstraction is one of the key concepts of object-oriented programming (OOP) languages. Its main goal is to handle complexity by hiding unnecessary details from the user. This enables the user to implement more complex logic on top of the provided abstraction without understanding about all the hidden complexity.

A powerful way to manage abstraction is through the use of hierarchical classifications. This allows us to layer the semantics of complex systems, breaking them into more manage-able pieces.

➢ Hierarchical abstractions of complex systems can also be applied to computer programs.

➢ The data from a traditional process oriented program can be transformed by abstraction into its component object

➢ A sequence of process steps can become a collection of messages between these objects.

➢ Thus, each of these objects describes its own unique behavior.

- We can treat these objects as concrete entities that respond to messages telling them to do something.

Object

Objects have states and behaviors. Example: A dog has states - color, name, breed as wellas behaviors – wagging the tail, barking, eating. An object is an instance of a class.

Class

A class can be defined as a template/blueprint that describes the behavior/state that theobject of its type support.

Objects in Java

If we consider the real-world, we can find many objects around us, cars, dogs, humans, etc. All these objects have a state and a behavior.

If we consider a dog, then its state is - name, breed, color, and the behavior is - barking, wagging the tail, running.

If we compare the software object with a real-world object, they have very similar characteristics.

Software objects also have a state and a behavior. A software object's state is stored in fields and behavior is shown via methods.

So in software development, methods operate on the internal state of an object and theobject-to-object communication is done via methods.

Classes in Java

A class is a blueprint from which individual objects are created.

```
public class Dog
{
String breed;
int age;
String color;
void barking()
   {
   }
}
```

A class can contain any of the following variable types.

- ***Local variables*** − Variables defined inside methods, constructors or blocks are called local variables. The variable will be declared and initialized within the method and the variable will be destroyed when the method has completed.
- ***Instance variables*** − Instance variables are variables within a class but outside any method. These variables are initialized when the class is instantiated. Instance variables can be accessed from inside any method, constructor or blocks of that particular class.
- ***Class variables*** − Class variables are variables declared within a class, outside any method, with the static keyword.

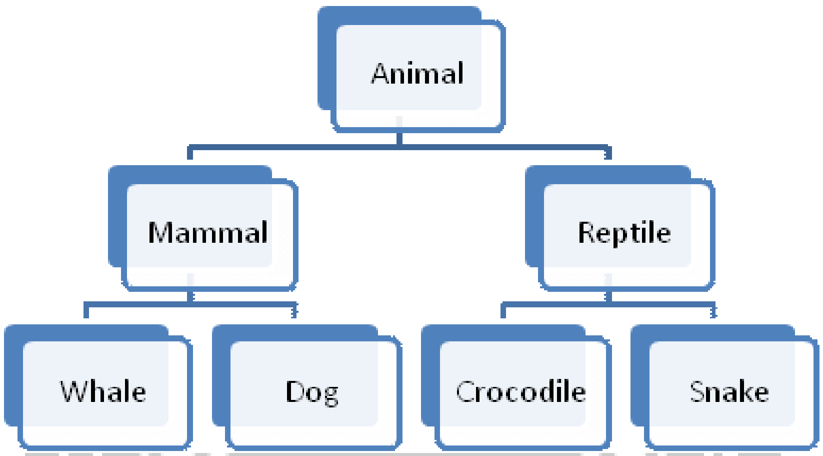
A class can have any number of methods to access the value of various kinds of methods. In the above example, barking(), hungry() and sleeping() are methods.

Encapsulation

Encapsulation is the mechanism that binds together code and the data it manipulates, and keeps both safe from outside interference and misuse.

- In Java, the basis of encapsulation is the class. There are mechanisms for hiding the complexity of the implementation inside the class.
- Each method or variable in a class may be marked private or public.
- The public interface of a class represents everything that external users of the classneed to know, or may know.
- The private methods and data can only be accessed by code that is a member of theclass.
- Therefore, any other code that is not a member of the class cannot access a private method or variable.
- Since the private members of a class may only be accessed by other parts of programthrough the class' public methods, we can ensure that no improper actions take place.

Inheritance



Inheritance is the process by which one object acquires the properties of another object.

For example a Dog is part of the classification mammal which in turn is part of the animal class.Without the use of hierarchies, each object would need to define all of its characteristics explicitly. However, by use of inheritance, an object need only define those qualities that make it unique within its class. It can inherit its general attributes from its parent. Thus, inheritance makes it possible for one object to be a specific instance of a more general case.

Polymorphism

Polymorphism (from Greek, meaning "many forms") is a feature that allows one interface to be used for a general class of actions. The specific action is determined by the exact natureof the situation.

For eg, a dog's sense of smell is polymorphic. If the dog smells a cat, it will bark and runafter it. If the dog smells its food, it will salivate and run to its bowl. The same sense of smellis at work in both situations. The difference is what is being smelled, that is, the type of databeing operated upon by the dog's nose.

Consider a stack (which is a last-in, first-out LIFO list). We might have a program requires three types of stacks. One stack is used for integer values, one for floating-point values,and one for characters. The algorithm that implements each stack is the same, even though the data being stored differs.