OOP CONCEPTS IN JAVA

OOP concepts in Java are the main ideas behind Java's Object Oriented Programming.

They are:

Object

Any entity that has state and behavior is known as an object. It can be either physical or logical.

For example: chair, pen, table, keyboard, bike etc.

Class & Instance

Collection of objects of the same kind is called class. It is a logical entity.

A Class is a 3-Compartment box encapsulating data and operations as shown in figure.

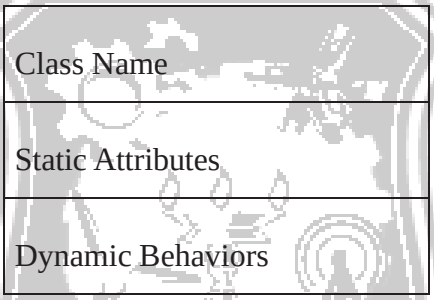


Figure: Class Structure

The followings figure shows two classes 'Student' and 'Circle'.

Name (Identifier)	Student	Circle
Variables (Static Attributes)	name, gender, dept, marks	radius, color
Methods (Dynamic Behaviors)	getDetails() calculateGrade()	getRadius() printArea()

Figure: Examples of classes

A class can be visualized as a three-compartment box, as illustrated:

1. Name (or identity): identifies the class.
2. Variables (or attribute, state, field): contain the static attributes of the class.
3. Methods (or behaviors, function, operation): contain the dynamic behaviors of theclass.

An instance is an instantiation of a class. All the instances of a class have similar properties, as described in the class definition. The term "object" usually refers to instance.

For example, we can define a class called "Student" and create three instances of the class"Student" for "John", "Priya" and "Anil".

The following figure shows three instances of the class Student, identified as "John","Priya" and "Anil".

John : Student	Priya : Student	Anil : Student
name = “John” gender = “male”	name = “Priya” gender = “female”	name = “Anil” gender = “male”
dept = “CSE” mark = 88	gender = “female” dept = “IT”	gender = “male” dept = “IT”
getDetails() calculateGrade()	getDetails() calculateGrade()	getDetails() calculateGrade()

Figure: Instances of a class ‘Student’

Abstraction

Abstraction refers to the quality of dealing with ideas rather than events. It basically dealswith hiding the details and showing the essential things to the user.

We all know how to turn the TV on, but we don’t need to know how it works in order to enjoy it.

Abstraction means simple things like objects, classes, and variables represent more complex underlying code and data. It avoids repeating the same work multiple times. In java, weuse abstract class and interface to achieve abstraction.

Abstract class:

Abstract class in Java contains the ‘abstract’ keyword. If a class is declared abstract, it cannot be instantiated. So we cannot create an object of an abstract class. Also, an abstract class can contain abstract as well as concrete methods.

To use an abstract class, we have to inherit it from another class where we have to provide implementations for the abstract methods there itself, else it will also become an abstract class.

Interface:

Interface in Java is a collection of abstract methods and static constants. In an interface, each method is public and abstract but it does not contain any constructor. Along with abstraction, interface also helps to achieve multiple inheritance in Java.

So an interface is a group of related methods with empty bodies.

Encapsulation

Binding (or wrapping) code and data together into a single unit is known as encapsulation.

It means to hide our data in order to make it safe from any modification.

The best way to understand encapsulation is to look at the example of a medical capsule, where the drug is always safe inside the capsule. Similarly, through encapsulation the method and variables of a class are well hidden and safe.

A java class is the example of encapsulation. Encapsulation can be achieved in Java by:

- Declaring the variables of a class as private.
- Providing public setter and getter methods to modify and view the variables values.

Inheritance

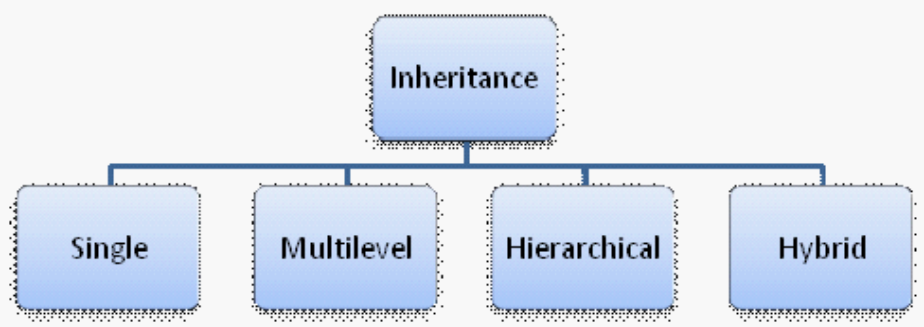
This is a special feature of Object Oriented Programming in Java. It lets programmerscreate new classes that share some of the attributes of existing classes.

For eg, a child inherits the properties from his father.
Similarly, in Java, there are two classes:
1. Parent class (Super or Base class)

2. Child class (Subclass or Derived class)

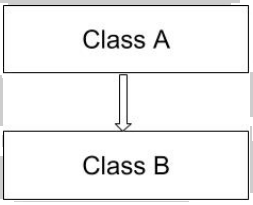
A class which inherits the properties is known as 'Child class' whereas a class whoseproperties are inherited is known as 'Parent class'.

Inheritance is classified into 4 types:



Single Inheritance

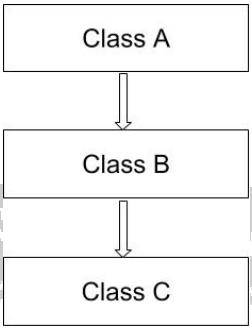
It enables a derived class to inherit the properties and behavior from a single parentclass.



Here, Class A is a parent class and Class B is a child class which inherits the propertiesand behavior of the parent class.

Multilevel Inheritance

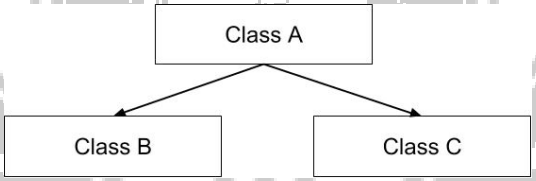
When a class is derived from a class which is also derived from another class, i.e. a classhaving more than one parent class but at different levels, such type of inheritance is called Multilevel Inheritance.



Here, class B inherits the properties and behavior of class A and class C inherits the properties of class B. Class A is the parent class for B and class B is the parent class for C. So, classC implicitly inherits the properties and methods of class A along with Class B.

Hierarchical Inheritance

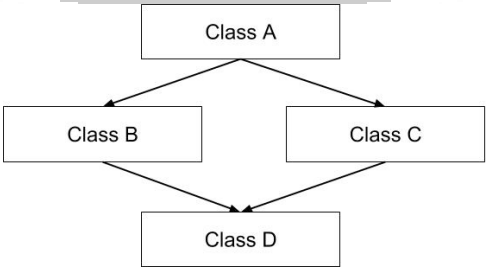
When a class has more than one child class (sub class), then such kind of inheritance is knownas hierarchical inheritance.



Here, classes B and C are the child classes which are inheriting from the parent class A.

Hybrid Inheritance

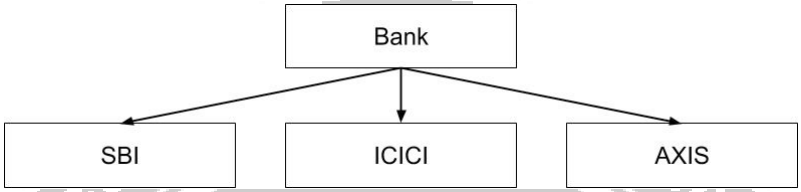
Hybrid inheritance is a combination of *multiple* inheritance and *multilevel* inheritance. Since multiple inheritance is not supported in Java as it leads to ambiguity, this type of inheritance can only be achieved through the use of the interfaces.



Here, class A is a parent class for classes B and C, whereas classes B and C are the parent classes of D which is the only child class of B and C.

Polymorphism

Polymorphism means taking many forms, where 'poly' means many and 'morph' means forms. It is the ability of a variable, function or object to take on multiple forms. In other words, polymorphism allows us to define one interface or method and have multiple implementations.



For eg, Bank is a base class that provides a method rate of interest. But, rate of interest may differ according to banks. For example, SBI, ICICI and AXIS are the child classes that provide different rates of interest.

Polymorphism in Java is of two types:

- Run time polymorphism
- Compile time polymorphism

Run time polymorphism:

In Java, runtime polymorphism refers to a process in which a call to an overridden method is resolved at runtime rather than at compile-time. Method overriding is an example of run time polymorphism.

Compile time polymorphism:

In Java, compile time polymorphism refers to a process in which a call to an overloaded method is resolved at compile time rather than at run time. Method overloading is an exampleof compile time polymorphism.