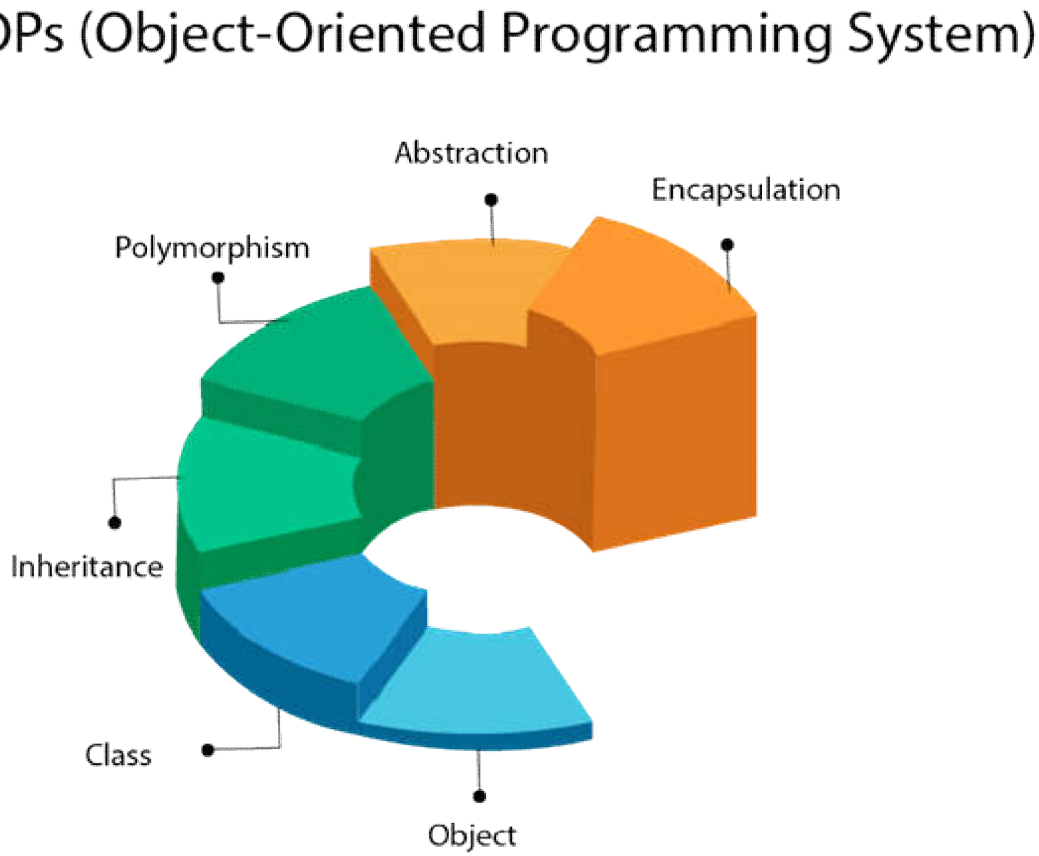
OOPs concept

- Object-oriented programming System(OOPs) is a programming paradigm based on the concept of "objects" that contain data and methods. The primary purpose of object-oriented programming is to increase the flexibility and maintainability of programs.



Class and Object

- **Class :** A class is a user defined blueprint or prototype from which objects are created.

- **Object :** Object is a bundle of data and its behaviour (often known as methods).

- *Every object contains three characteristics,*
  1) State : well defined condition of an item (instance variable/fields/properties)
  2) Behavior : effects on an item (methods/behavior)
  3) identity : identification number of an item(hash code)

Principles of OOP

- *The main building blocks of oops are,*
  1. ***Encapsulation***
  2. ***Inheritance***
  3. ***Polymorphism***
  4. ***Abstraction***

Encapsulation

- The process of binding the data and code as a single unit is called encapsulation.

- It also means to hide your data in order to make it safe from any modification.

- We can achieve encapsulation in Java by:
  - Declaring the variables of a class as private.
  - Providing public setter and getter methods to modify and view the variables values.

Advantages of Encapsulation

- **We can make a class read-only** or **write-only**: for a read-only class, we should provide only a getter method. For a write-only class, we should provide only a setter method.

- **Control over the data:** we can control the data by providing logic to setter methods, just like we restricted the stock keeper from assigning negative values in the above example.

- **Data hiding:** other classes can't access private members of a class directly.

Inheritance

- *The process of acquiring properties (variables) & methods (behaviors) from one class to another class is called inheritance.*

- *We are achieving inheritance concept by using 'extends' and 'implements' keyword. Inheritance is also known as is a relationship.*

- *Extends keyword is providing relationship between two classes.*

- *Implements keyword is providing relationship between class and interface.*

Inheritance

- *The main objective of inheritance is code extensibility whenever we are extending the class automatically code is reused.*
- *In inheritance one class providing properties & another class is acquiring the properties.*
- *In inheritance parent class is giving properties & Child is acquiring properties from Parent.*
- In java it is possible to create objects for both parent and child classes,

Inheritance

- If we are creating object for parent class it is possible to access only parent specific method.
- if we are creating object for child class it is possible to access both parent & child specific methods.
- *The root class of all java classes is "**object**" class.*
- *in java if we are extending the class then it will be parent class , if we are not extending the class then object class will become parent class.*

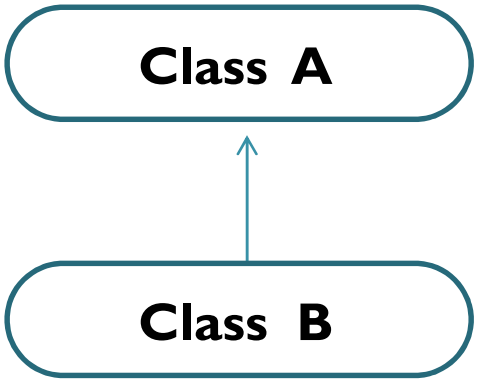
Types of Inheritance

- *There are five types of inheritance in java:*

  1. ***Single inheritance***
  2. ***Multilevel inheritance***
  3. ***Hierarchical inheritance***
  4. ***Multiple inheritance***
  5. ***Hybrid Inheritance***

Single Inheritance

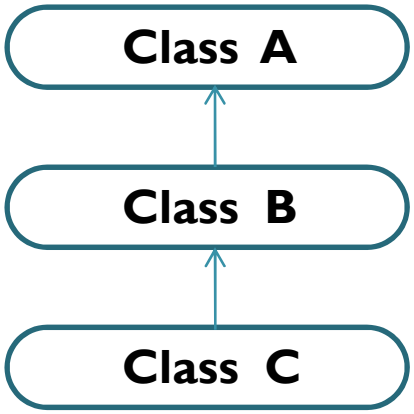
- *One class has only one direct super class is called single inheritance.*
- *In the absence of any other explicit super class, every class is implicitly a subclass of Object class.*



- *class B acquiring properties of A class.*

Multilevel Inheritance

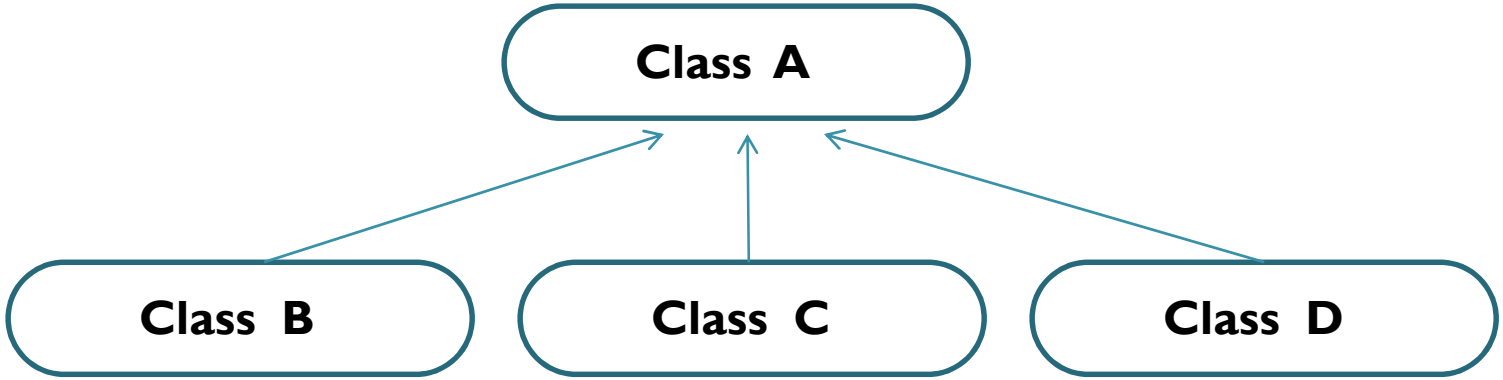
- *One Sub class is extending Parent class then that sub class will become Parent class of next extended class this flow is called multilevel inheritance.*



- *class B acquiring properties of A class*
- *class C acquiring properties of B class*
- *indirectly class C using properties of A & B classes*

Hierarchical Inheritance

- *More than one sub class is extending single Parent is called hierarchical inheritance.*

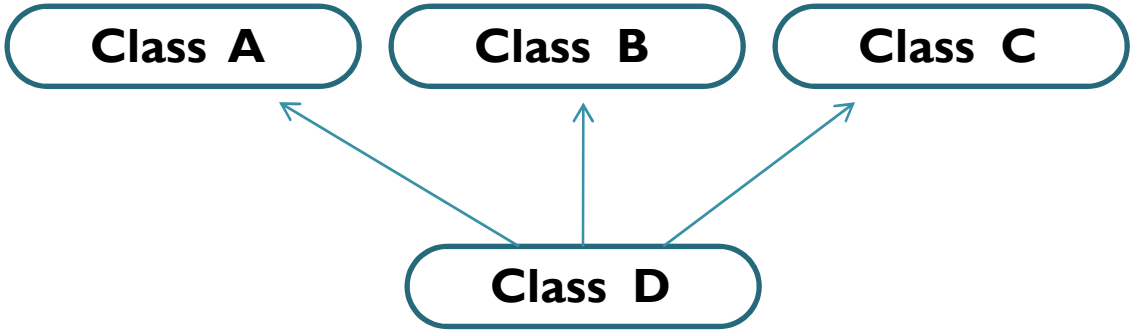


- *class B acquiring properties of A class*
- *class C acquiring properties of A class*
- *class D acquiring properties of A class*

Multiple Inheritance

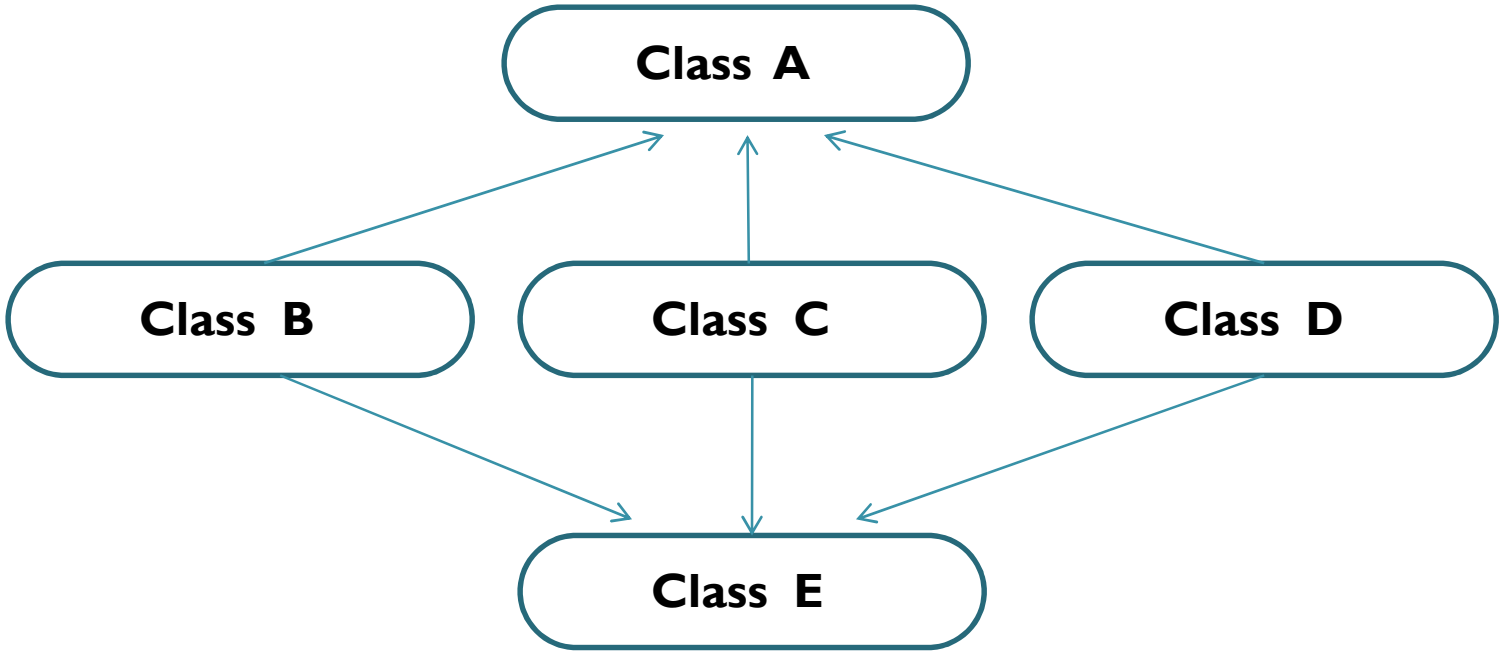
- *One sub class is extending more than one super class is called Multiple inheritance and java not supporting multiple inheritance because it is creating ambiguity problems.*

- *Java not supporting multiple inheritances hence in java one class able to extends only one class*
  at a time but it is not possible to extends more than one class.



Hybrid Inheritance

- *Hybrid is combination of hierarchical & multiple inheritance .*
- *Java is not supporting hybrid inheritance because multiple inheritance(not supported by java) is included in hybrid inheritance.*



Instanceof Operator

- *It is used check the type of the object and it returns Boolean value as a return value.*

- *To use the "**instanceof**" operator the class name & reference variable must have some relationship either parent to child or child to parent otherwise compiler will generate error message "**inconvertible types**"*

- *If the relationship is child to parent it returns true & the relationship is parent to child it return false.*

Super keyword

- *"this" keyword is used to represent current class object& "super" keyword is used to represent super class object.*
  1. Super class variables.
  2. Super class methods.
  3. Super class constructors.

Polymorphism

- *The ability to appear in more forms is called polymorphism.*
- *One functionality with different actions is called polymorphism.*
- *Polymorphism is a Greek word poly means many and morphism means forms.*
- There are two types of polymorphism in java :

  1) Compile time polymorphism or static binding or early binding

  2) Runtime polymorphism or dynamic binding or late binding.

Polymorphism

- *Compile time polymorphism like overloading.*

- *Types of overloading:*
  a. Method overloading
  b. Constructor overloading
  c. Operator overloading

Compile time Polymorphism

- ***Method Overloading:*** If java class allows more than one method with same name but different number of arguments or same number of arguments but different data types those methods are called
overloaded methods.

- To achieve overloading concept one java class sufficient

- It is possible to overload any number of methods in single java class

- *In method overloading it is possible to have different data types for overloaded method.*

- *While overloading the methods check the signature (methodname+parameters) of the method but not return type.*

Compile time Polymorphism

- ***Constructor Overloading:*** *If the class contains more than one constructors with same name but different arguments or*
  same number of arguments with different data types those constructors are called overloaded constructors

- ***Operator overloading:*** One operator with different behaviors is called Operator overloading.

- Java is not supporting operator overloading but only one overloaded in java language is '+'.
  - If both operands are number then "**+**" performs addition.
  - If at least one operand is String then "**+**" perform concatenation.

Runtime Polymorphism

- *Runtime polymorphism Method Overriding):*

  - To achieve method overloading one java class sufficient but to achieve method overriding we required two java classes with parent and child relationship.

- *In overriding parent class method is called overridden method*

- *Child class method is called overriding method*

Runtime Polymorphism

- ***While overriding methods must fallow these rules:-***

  1) Overridden method signature & overriding method signatures must be same.

  2) The return types of overridden method & overriding method must be same (at primitive level).

  3) While overriding it is possible to change return type by using co-variant return types concept.

  4) Final methods can't override.

  5) Static method can't override but method hiding possible.

  6) Private methods can't override.

  7) While overriding it is not possible to maintain same level permission or increasing order but not decreasing.

Abstraction

- Abstraction is a process of hiding the implementation details and showing only functionality to the user.

- Another way, it shows only essential things to the user and hides the internal details

- There are two ways to achieve abstraction in java :
  - Abstract class (0 to 100%)
  - Interface (100%)

- Abstract class is a java class which contains at least one abstract method(wrong definition).

- If any abstract method inside the class that class must be abstract.

Abstract modifier

- Abstract modifier is applicable for methods and classes but not for variables.
- To represent particular class is abstract class and particular method is abstract method to the compiler use abstract modifier.
- The abstract class contains declaration of methods it says abstract class partially implement class hence for partially implemented classes object creation is not possible. If we are trying to create object of abstract class compiler generate error message “class is abstract con not be instantiated”

Abstract Class

- Abstract class contains abstract methods for that abstract methods provide the implementation in child classes.
- if the child class is unable to provide the implementation of all parent class abstract methods at that situation declare that class with abstract modifier then take one more child class to complete the implementation of remaining abstract methods.
- It is possible to declare multiple child classes but at final complete the implementation of all methods.
- It is possible to override non-abstract as a abstract method in child class.