Inheritance:

It creates new classes from existing classes,so that the new classes will aquire all the fetures of the existing classes is called Inheritance.

Father and Child relationship is called Inheritance.

 Definition:

A class that is derived from another class is called a *subclass* (also a *derived class*, *extended class*, or *child class*). The class from which the subclass is derived is called a *superclass* (also a *base class* or a *parent class*).

IS-A relationship is also called inheritance.

By using extends keyword we can implement IS-A relation ship

Application Without inheritance:

- code duplication(Redundancy)
- Length of the code is increased.

Example:

```
class A
{ void m1(){}
 void m2(){}
}
class B
{ void m1(){}
 void m2(){}
        void m3(){}
        void m4(){}
}
class C
{ void m1(){}
 void m2(){}
 void m3(){}
 void m4(){}
        void m5(){}
        void m6(){}
}
```

Application With Inheritance/Advantage of Inheritance:

- To reduce the Redundancy
- Reduce the length of the code.

The syntax of inheritance is:

        class subclass extends superclass
        {
        }

Example:

class A   //class which providing is called Parent class or Super class or base class
{
        void m1(){}
        void m2(){}
}
class B extends A //class which deriving is called Child class or Sub class or derived class
{
        void m3(){}
        void m4(){}
}
class C extends B
{
        void m5(){}
        void m6(){}
}

Why super class members are available to sub class?
        Because,the sub class object contains a copy of super class object.
Q)What is the advantage of inheritance?
        In inheritance,a programmer reuses the super class code without rewriting it,in creation of sub classes.So,developing the classes becomes very easy.Hence,the programmer's productivity is increased.

Note:
- extends is a keyword which is used for inheriting the data members and methods from base class to the derived class and it also improves functionality of derived class.
- Final classes cannot be inherited.
- If the base class contains private data members then that type of data members will not be inherited into derived class.

Whenever we develop any inheritance application, it is always recommended to create an object of bottom most derived class. Since, bottom most derived class contains all the features from its super classes.

• One class can extend only one class at a time. Since, JAVA does not support multiple inheritances. Whenever we inherit the base class members into derived class, when we creates an object of derived class, JVM always creates the memory space for base class members first and later memory space will be created for derived class members.

Example1:
```
class Parent
{
public void m1()
{
System.out.println("Parent");
}
}
class Child extends Parent
{
public void m2()
{
System.out.println("Child");

}
}

class Test
{
public static void main(String args[])
{
//case 1
Parent p=new Parent();
p.m1();
}
}
```

output:
D:\Yellaswamy_ClassNotes\UNIT-2\Inheritance\Example1>javac Test.java
D:\Yellaswamy_ClassNotes\UNIT-2\Inheritance\Example1>java Test
Parent

Example2:
```
class Parent
{
public void m1()
{
System.out.println("Parent");
}
}
class Child extends Parent
{
public void m2()
{
System.out.println("Child");

}
}

class Test
{
public static void main(String args[])
{
//case 1
Parent p=new Parent();
p.m1();
p.m2();
}
}
```
output:

```
D:\Yellaswamy_ClassNotes\UNIT-2\Inheritance\Example2>javac Test.java
Test.java:24: cannot find symbol
symbol  : method m2()
location: class Parent
p.m2();
 ^
1 error
```

Example 3:

```java
class Parent
{
public void m1()
{
System.out.println("Parent");
}
}
class Child extends Parent
{
public void m2()
{
System.out.println("Child");

}
}

class Test
{
public static void main(String args[])
{
//case 2
Child c=new Child();
c.m1();
c.m2();
}
}
```

output:

```
D:\Yellaswamy_ClassNotes\UNIT-2\Inheritance\Example3>javac Test.java
D:\Yellaswamy_ClassNotes\UNIT-2\Inheritance\Example3>java Test
Parent
Child
```

Example 4:

```
class Parent
{
public void m1()
{
System.out.println("Parent");
}
}
class Child extends Parent
{
public void m2()
{
System.out.println("Child");

}
}

class Test
{
public static void main(String args[])
{
//case 3
Parent p=new Child();
p.m1();
p.m2();
}
}
```

output:

```
D:\Yellaswamy_ClassNotes\UNIT-2\Inheritance\Example4>javac Test.java
Test.java:24: cannot find symbol
symbol  : method m2()
location: class Parent
p.m2();
 ^
1 error
```

Example 5:

```java
class Parent
{
public void m1()
{
System.out.println("Parent");
}
}
class Child extends Parent
{
public void m2()
{
System.out.println("Child");

}
}

class Test
{
public static void main(String args[])
{
//case 4
Child c=new Parent();

}
}
```

output:

```
D:\Yellaswamy_ClassNotes\UNIT-2\Inheritance\Example5>javac Test.java
Test.java:22: incompatible types
found   : Parent
required: Child
Child c=new Parent();
     ^
1 error
```

Conclusions:

> ➢ what ever methods parent has by default available to the child and hence on the child reference we can call both parent and child class methods.
> ➢ what ever methods child has by default not available to the parent and hence,on the parent reference we can not call child specific methods.

> ➤ Parent reference can be used to hold child object but using that reference we can't call child specific methods but we can call the methods present in parent class.
> ➤ parent reference can be used to hold child object but child reference can't be used to hold parent object.

Total Java API is implemented is based on inheritance concept.

The most common methods which are applicable for any java object are defined in Object class.hence every class in java is the child class of Object either directly or indirectly.so that Object class methods by default available to every java class with out re writing due to this Object class acts as root for all java classes.
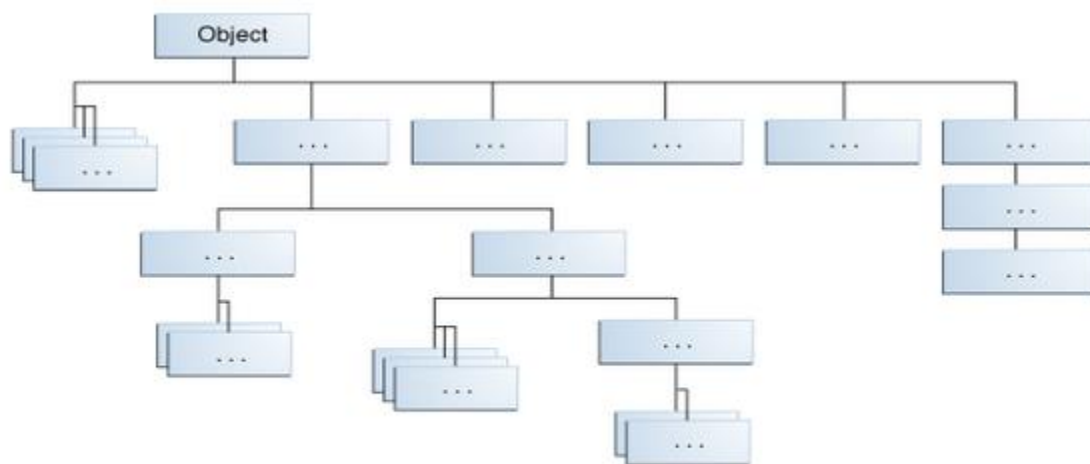
All Classes in the Java Platform are Descendants of Object

Types of Inheritance:

There are 5 Types of Inheritance.
1. Single Inheritance
2. Multilevel Inheritance
3. Hierarchical Inheritance
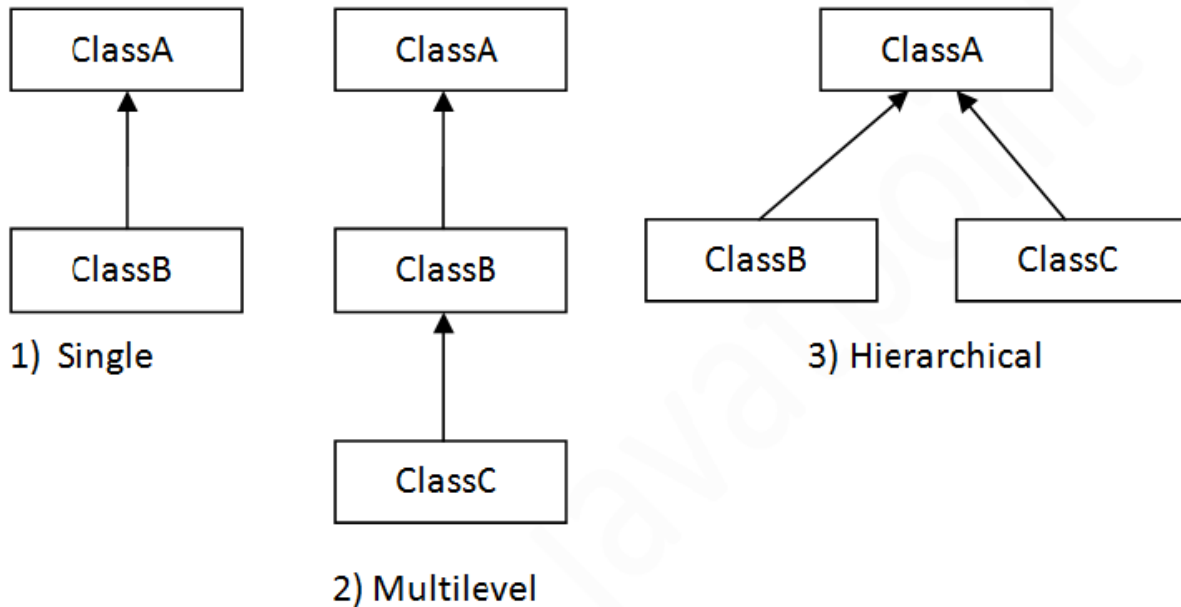4. Multiple Inheritance
5. Hydrid Inheritance

Single Inheritance:
Producing sub classes from a single super class is called single inheritance.

Types of inheritance in java:

On the basis of class, there can be three types of inheritance in java: single, multilevel and hierarchical.

In java programming, multiple and hybrid inheritance is supported through interface only. We will learn about interfaces later.

Example:
```
class A
{
}
class B extends A
{
}
```

2.Multilevel Inheritance
Example:
```
class A
{
}
class B extends A
{
}
```

```
class C extends B
{
}
```

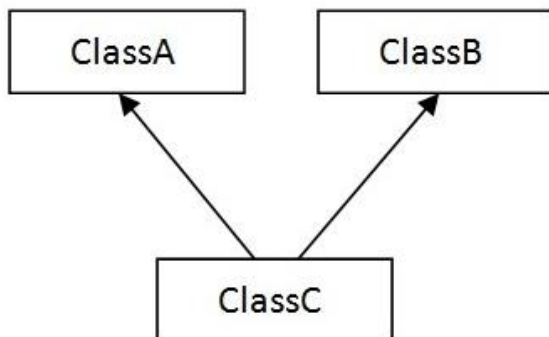3.Hierarchical Inheritance

Example:

```
class A
{
}
class B extends A
{
}
class C extends A
{
}
class D extends A
{
}
```
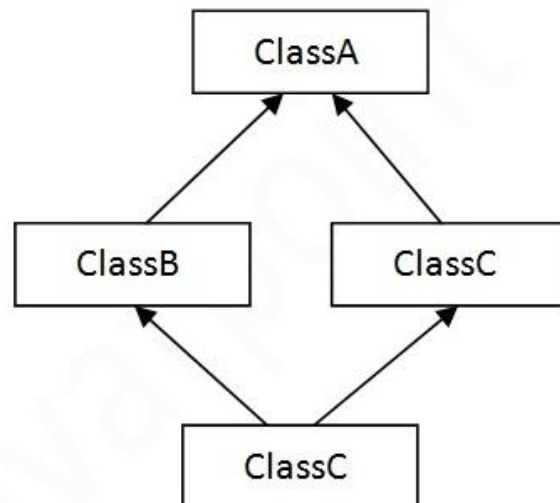
4. Multiple Inheritance

A java class can't extend more than one class at a time hence java would not provide support for multiple inheritance.

Note: Multiple inheritance is not supported in java through class.

When a class extends multiple classes i.e. known as multiple inheritance. For Example:

4) Multiple

5) Hybrid

Example:

```
class A
{
}
class B
{
}
class C extends A,B
{
}
```

5.Hybrid Inheritance.
combination of Multiple and Hirarchical inheritances is called Hybrid Inheritance.

Java is Not suporting Hybrid Inheritance.
Example:

```
class A
{
}
class B extends A
{
}
class C extends A
{
}
class C extends A,B   //invalid
{
}
```

Note:
1.if our class doesn't extend any other class then only our class is direct child class of Object.

Example:
```
class A
{
}
```

2.if our class extends any other class then our class is indirect child class of Object.
3.Cyclic Inheritance is not allowed in Java ofcoure it is not required.