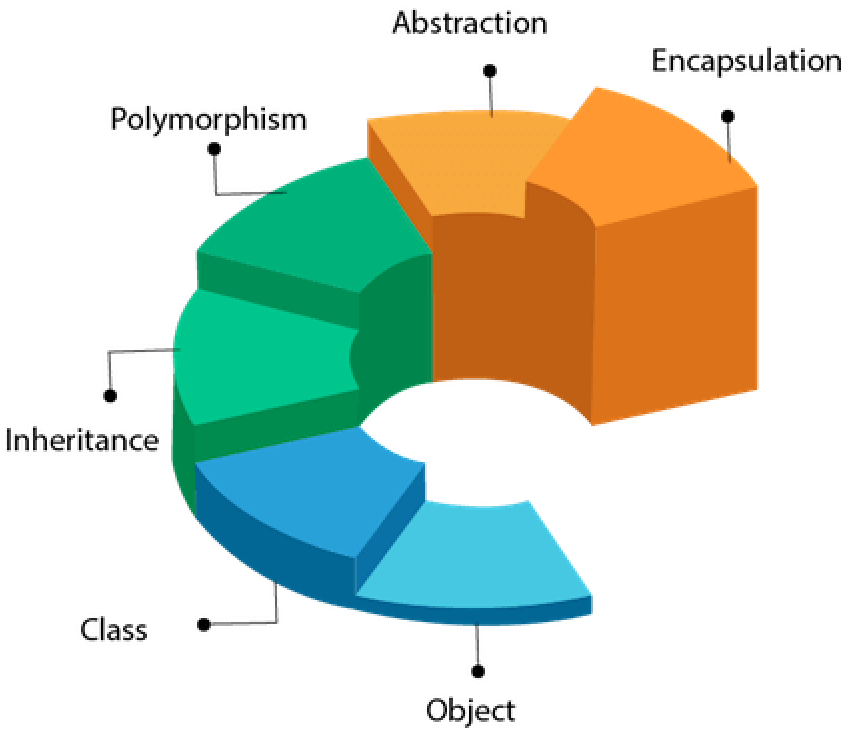
JAVA OOP’s Concepts
===================

OOPs (Object-Oriented Programming System)



1. Object
2. Class
3. Inheritance
4. Polymorphism
5. Abstraction
6. Encapsulation

—--------------

Inheritance
===========

Inheritance is one of the OOPs concepts, in this subclass acquires the properties of superclass by using “extents” keyword only by creating objects of subclass.

a. Super Class / Parent class

   —--------------------------------

```
public class Superclass
{
        public void method1()                           // 1st super class method
        {
                System.out.println("first parent method");
        }
        public void method2()                           // 2nd super class method
        {
                System.out.println("second parent method");
        }
}
```

b. Sub Class / Child class

   —---------------------------

```
public class SubClass extends Superclass
{

        public void method3()                           // sub class method
        {
                System.out.println("subclass method");
        }

        public static void main(String[] args)
        {
                SubClass s = new SubClass();        // creating object of subclass
                s.method3();                        // calling subclass method
                s.method1();                        // calling superclass method
                s.method2();                        // calling superclass method
        }
}
```

```
Output :        subclass method
                first parent method
                second parent method
```

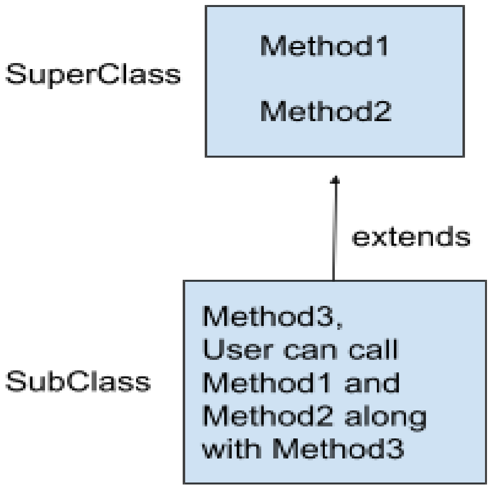
==========

Types of Inheritance

a. **Single level inheritance**
b. **Multilevel inheritance**
c. **Multiple inheritance**
d. **Hierarchical inheritance**

a. Single level Inheritance

In single level inheritance one subclass acquires the properties of superclass by using “extends” keyword only by creating objects of subclass.



E.x **Super class >>**

```
public class Superclass
{
        public void test1()                             // superclass property
        {
                System.out.println("super class property");
        }
}
```

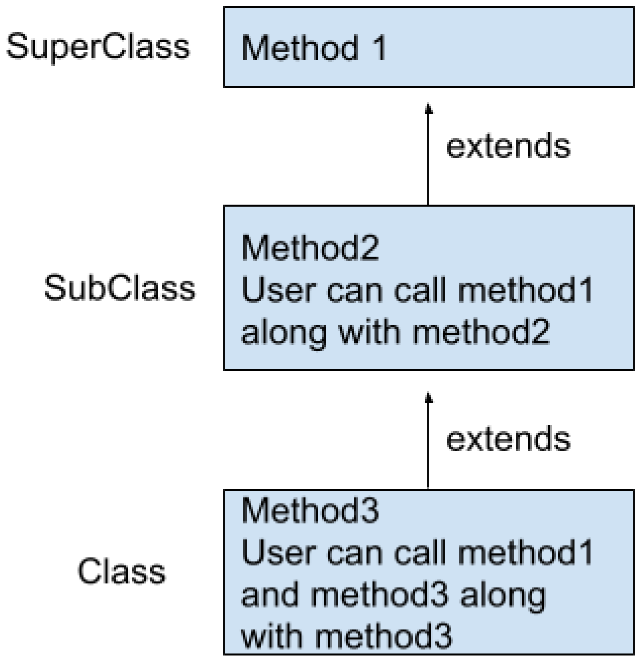
Subclass >>

```
public class Subclass extends Superclass
{
        public void test2()                             // subclass property
        {
                System.out.println("Subclass property");
        }
        public static void main(String[] args)
        {
                Subclass s = new Subclass();            // creating object of subclass
                s.test1();                              // // calling superclass method
                s.test2();
        }
}
```

—---------------

b. Multi level Inheritance

In multilevel inheritance one subclass acquires the property of superclass which also acquires the property of another super class.



E.x **Super class >>**

```
public class Superclass
{
        public void test1()                              // superclass property
        {
                System.out.println("super class property");
        }
}
```

Subclass >>

```
public class Subclass extends Superclass
{
        public void test2()                              // subclass property
        {
                System.out.println("Subclass property");
        }
        public static void main(String[] args)
        {
                Subclass s = new Subclass();             // creating object of subclass
                s.test2();                               // calling subclass method
                s.test1();                               // calling superclass method
        }
}
```

Class >>

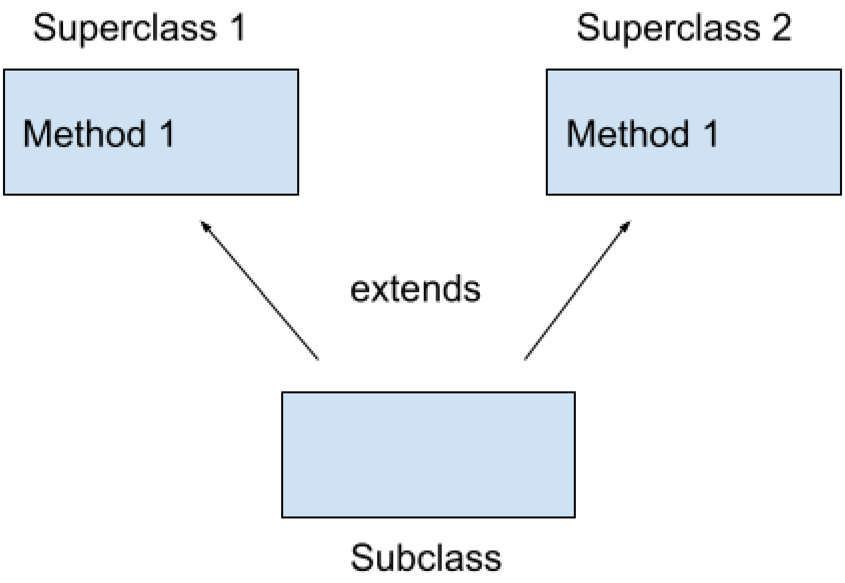
```
public class Class extends Subclass
{
        public void test3()
        {
                System.out.println("Class property");
        }

        public static void main(String[] args)
        {
                Class sc = new Class();                 // creating object of subclass
                sc.test3();                             // calling class method
                sc.test2();                             // calling subclass method
        }
}
```

—---------------

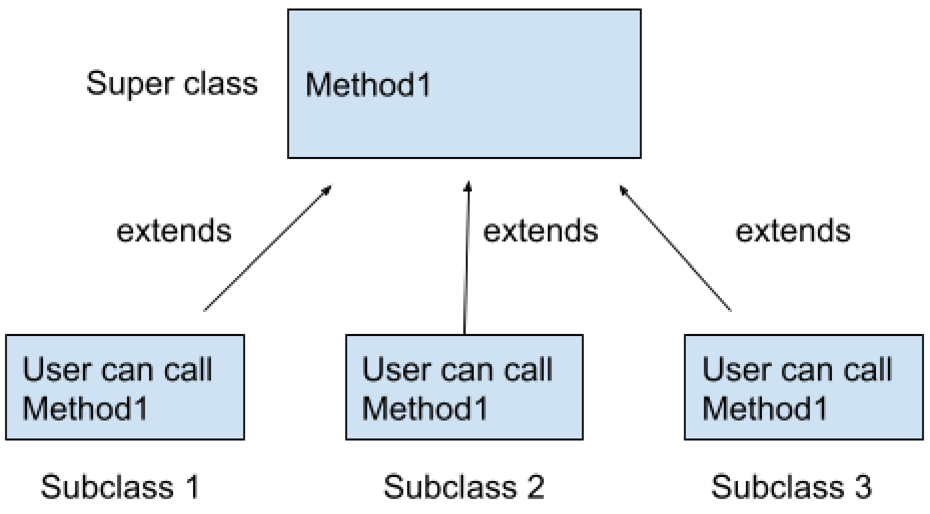
c. Multiple Inheritance

In multiple inheritance one subclass acquires the property of two different superclasses, but it is not possible in JAVA because it falls under diamond ambiguity, if the user tried to extends two classes simultaneously then the user get the compile time error.



d. Hierarchical Inheritance

In hierarchical inheritance multiple subclasses acquire the property of one super class.



E.x **Super class >>**

```
public class Superclass
{
        public void test1()                     // superclass property
        {
                System.out.println("super class property");
        }
}
```

Subclass >>

```
public class Subclass extends Superclass
{
        public void test2()                     // subclass property
        {
                System.out.println("Subclass property");
        }
        public static void main(String[] args)
        {
                Subclass s = new Subclass();    // creating object of subclass
                s.test2();                      // calling subclass method
                s.test1();                      // calling superclass method
        }
}
```

Class >>

```
public class Class extends Superclass
{
	public void test3()
	{
		System.out.println("Class property");
	}

	public static void main(String[] args)
	{
		Class sc = new Class();			// creating object of subclass
		sc.test3();				// calling class method
		sc.test1();				// calling subclass method
	}
}
```

===