

SUPER KEYWORD IN JAVA

- The super keyword in Java is a reference variable which is used to refer immediate parent class object.
- Whenever you create the instance of subclass, an instance of parent class is created implicitly which is referred by super reference variable.
- **Usage of Java super Keyword -**
 - super can be used to refer immediate parent class instance variable.
 - super can be used to invoke immediate parent class method.
 - super() can be used to invoke immediate parent class constructor.

super is used to refer immediate parent class instance variable

We can use super keyword to access the data member or field of parent class. It is used if parent class and child class have same fields.

```
class Animal
{
    String color="white";
}

class Dog extends Animal
{
    String color="black";
    void printColor()
    {
        System.out.println(color);           //prints color of Dog class
        System.out.println(super.color);     //prints color of Animal class
    }
}

class TestSuper1
{
    public static void main(String args[])
    {
        Dog d=new Dog();
        d.printColor();
    }
}
```

super can be used to invoke parent class method

The super keyword can also be used to invoke parent class method. It should be used if subclass contains the same method as parent class. In other words, it is used if method is overridden.

```
class Animal
{
    void eat()
    {
        System.out.println("eating...");
    }
}

class Dog extends Animal
{
    void eat()
    {
        System.out.println("eating bread...");
    }

    void bark()
    {
        System.out.println("barking...");
    }

    void work()
    {
        super.eat(); bark();
    }
}

class TestSuper2
{
    public static void main(String args[])
    {
        Dog d=new Dog();
        d.work();
    }
}
```

super is used to invoke parent class constructor

The super keyword can also be used to invoke the parent class constructor. Let's see a simple example:

```
class Animal
{
    Animal()
    {
        System.out.println("animal is created");
    }
}

class Dog extends Animal
{
    Dog()
    {
        super();
        System.out.println("dog is created");
    }
}

class TestSuper3
{
    public static void main(String args[])
    {
        Dog d=new Dog();
    }
}
```

Note: super() is added in each class constructor automatically by compiler if there is no super() or this().