

A flock of approximately 12 birds is flying in a loose circular pattern around the central text. The birds are dark in color with lighter, possibly golden-brown, wingtips. The background is a solid, muted teal color.

Super Keyword

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

1. super can be used to refer immediate parent class instance variable.
2. super can be used to invoke immediate parent class method.
3. super() can be used to invoke immediate parent class constructor.

1) 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();
    }
}
```

OUTPUT:

black
white

In the above example, Animal and Dog both classes have a common property color. If we print color property, it will print the color of current class by default. To access the parent property, we need to use super keyword.

2) 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(); }}
```

Output:
eating...
barking...

In the above example Animal and Dog both classes have eat() method if we call eat() method from Dog class, it will call the eat() method of Dog class by default because priority is given to local.

To call the parent class method, we need to use super keyword.

3) super is used to invoke parent class constructor.

In Java, constructor of base class with no argument gets automatically called in derived class constructor.

// filename: Main.java

```
class Base {  
    Base() {  
        System.out.println("Base Class Constructor Called ");  
    }  
}  
  
class Derived extends Base {  
    Derived() {  
        System.out.println("Derived Class Constructor Called ");  
    }  
}
```

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

But, if we want to call parameterized constructor of base class, then we can call it using `super()`. The point to note is **base class constructor call must be the first line in derived class constructor**. For example, in the following program, `super(_x)` is first line derived class constructor.

// filename: Main.java

```
class Base {  
  
    int x;  
  
    Base(int _x) {  
  
        x = _x;  
  
    }  
  
}
```

```
class Derived extends Base {  
  
    int y;  
  
    Derived(int _x, int _y) {  
  
        super(_x);  
  
        y = _y;  
  
    }  
  
    void Display() {  
  
        System.out.println("x = "+x+", y = "+y);  
  
    }  
  
}
```

```
public class Main {  
  
    public static void main(String[] args) {  
  
        Derived d = new Derived(10, 20);  
  
        d.Display();  
  
    }  
  
}
```

Output:

x = 10, y = 20



you'll grow beautifully in your own way.

- D h i m a n