

Object Oriented Programming using Java

Prepared By:
Suyel, PhD
Assistant Professor
Dept. of CSE, NIT Patna



Outline

1. super Keyword
2. Usage of super Keyword

super Keyword

- ❑ The concept of **super** keyword comes with the idea of inheritance in Java.
- ❑ Suppose we have a member variable or method of the same name in the derived class as we do in the base class. When referring to that variable/method, how would the program know which one are we referring to; the base class or the derived class?
- ❑ To solve the above mentioned problem, super keyword acts like a reference variable of the immediate parent class.
- ❑ It is mainly used when we want to access a variable, method or constructor of the base class from the derived class.

Usage of super Keyword

- ❑ Accessing immediate parent class instance variable
 - ❖ When we have the data members of the same name in both the parent and child class, we can use the super keyword to access the parent class's data member or field from the child class.
- ❑ Invoking immediate parent class method
 - ❖ When the name of a method is same in both the parent and child class, the super keyword can be used to invoke the parent class method from the child class.
- ❑ Invoking immediate parent class constructor
 - ❖ super() can also be used to invoke the parent class's constructor (parameterized and no argument) from in the derived class.
 - ❖ super() must be the first statement inside the subclass constructor.

Usage of super Keyword (Cont...)

❑ Accessing immediate parent class instance variable

```
class Animal
{
    String color = "white";
    int age = 4;
}

class Dog extends Animal
{
    String color = "Black";
    int age = 1;

    void getColor()
    {
        System.out.println("Color of Child: " + color + " and age of Child dog: " + age);
        System.out.println("Color of Parent: " + super.color + " and age of Parent dog: " + super.age);
    }
}

class Super1
{
    public static void main(String args[])
    {
        Dog obj = new Dog();
        obj.getColor();
    }
}
```

Usage of super Keyword (Cont...)

- Accessing immediate parent class instance variable (Cont...)

- ❖ Output

```
Color of Child: Black and age of Child dog: 1  
Color of Parent: White and age of Parent dog: 4
```

Usage of super Keyword (Cont...)

❑ Invoking immediate parent class method

```
class Animal
{
    void getColor()
    {
        System.out.println("color of the parent dog is while");
    }
}

class Dog extends Animal
{
    void getColor()
    {
        System.out.println("color of the child dog is Black");
    }

    void show()
    {
        getColor();
        super.getColor();
    }

    public static void main(String args[])
    {
        Dog obj = new Dog();
        obj.show();
    }
}
```

Usage of super Keyword (Cont...)

□ Invoking immediate parent class method (Cont...)

❖ Output

```
Color of the Child dog is Black  
Color of the parent dog is While
```


Usage of super Keyword (Cont...)

❑ Invoking immediate parent class method (Cont...)

```
class Animal
{
    void getColor()
    {
        System.out.println("Color of the parent dog is while");
    }
}

class Dog1 extends Animal
{
    void getColor1()
    {
        System.out.println("Color of the child dog is Black");
    }

    void show()
    {
        getColor1();
        super.getColor();
    }

    public static void main(String args[])
    {
        Dog1 obj = new Dog1();
        obj.show();
    }
}
```

Usage of super Keyword (Cont...)

□ Invoking immediate parent class method (Cont...)

❖ Output

```
Color of the Child dog is Black  
Color of the parent dog is While
```

//In the last example, we have used super keyword unnecessarily. The program would work perfectly without super keyword.

Usage of super Keyword (Cont...)

❑ Invoking immediate parent class constructor

```
class Animal
{
    Animal()
    {
        system.out.println("It is the parent class of Animal");
    }
}

class Dog extends Animal
{
    Dog()
    {
        system.out.println("It, i.e. Dog is the child class of Animal");
    }
}

class Super2
{
    public static void main(String args[])
    {
        Dog obj=new Dog();
    }
}
```

Usage of super Keyword (Cont...)

□ Invoking immediate parent class constructor (Cont...)

❖ Output

```
It is the parent class of Animal  
It, i.e. Dog is the Child class of Animal
```

Usage of super Keyword (Cont...)

❑ Invoking immediate parent class constructor (Cont...)

```
class Parent
{
    Parent()
    {
        System.out.println("Constructor of the parent class");
    }
}

class Child extends Parent
{
    Child()
    {
        System.out.println("Constructor of the child class");
    }

    Child(int x)
    {
        System.out.println("Parameterized Constructor of the child class");
    }

    void disp()
    {
        System.out.println("Hello, how are you?");
    }

    public static void main(String args[])
    {
        Child obj1 = new Child();
        obj1.disp();
        Child obj2 = new Child(10);
        obj2.disp();
    }
}
```

Usage of super Keyword (Cont...)

□ Invoking immediate parent class constructor (Cont...)

❖ Output

```
Constructor of the parent class  
Constructor of the child class  
Hello, how are you?  
Constructor of the parent class  
Parameterized Constructor of the child class  
Hello, how are you?
```

//In the last example, compiler adds super() in both the constructors of the child class.

Usage of super Keyword (Cont...)

❑ Invoking immediate parent class constructor (Cont...)

```
class Animal
{
    Animal()
    {
        System.out.println("It is the parent class of Animal");
    }
}

class Dog extends Animal
{
    Dog()
    {
        super();
        System.out.println("It, i.e. Dog is the child class of Animal");
    }
}

class Super3
{
    public static void main(String args[])
    {
        Dog obj = new Dog();
    }
}
```

Usage of super Keyword (Cont...)

□ Invoking immediate parent class constructor (Cont...)

❖ Output

```
It is the parent class of Animal  
It, i.e. Dog is the Child class of Animal
```


Usage of super Keyword (Cont...)

□ Invoking immediate parent class constructor (Cont...)

```
class Animal
{
    Animal()
    {
        System.out.println("It is no argument constructor of the Animal class");
    }

    Animal(String str)
    {
        System.out.println(str);
        System.out.println("It is Parameterized Constructor of the Animal class");
    }
}

class Dog2 extends Animal
{
    Dog2()
    {
        super("We are at NIT Patna");
        System.out.println("Now, we are in the constructor of the child class");
    }

    void disp()
    {
        System.out.println("Hello, how are you?");
    }

    public static void main(String args[])
    {
        Dog2 obj = new Dog2();
        obj.disp();
    }
}
```

Usage of super Keyword (Cont...)

□ Invoking immediate parent class constructor (Cont...)

❖ Output

```
We are at NIT Patna  
It is Parameterized Constructor of the Animal class  
Now, we are in the Constructor of the child class  
Hello, how are you?
```

Usage of super Keyword (Cont...)

❑ Invoking immediate parent class constructor (Cont...)

Company.java

```
class Company
{
    int id;
    String name;

    company(int id, String name)
    {
        this.id = id;
        this.name = name;
    }
}
```

Super4.java

```
class Super4
{
    public static void main(String args[])
    {
        Emp obj = new Emp(1010, "Anjali", 40000f);
        obj.disp();
    }
}
```

Emp.java

```
class Emp extends Company
{
    float salary;
    Emp(int id, String name, float salary)
    {
        super(id, name);
        this.salary = salary;
    }

    void disp()
    {
        System.out.println("Identity: " + id + ", Name: " + name + ", and salary: " + salary);
    }
}
```

Usage of super Keyword (Cont...)

- Invoking immediate parent class constructor (Cont...)

- ❖ Output

Identity: 1010, Name: Anjali, and Salary: 40000.0



**Slides are prepared from various sources,
such as Book, Internet Links and many
more.**