

MOD -02 -> OOPs



# super keyword



# Super Keyword

Super is a keyword in java which refers to the immediate super class object.

Use of super keyword in java

1. Super can be used to call immediate super class constructor (constructor chaining).
2. Super can be used to call immediate super class method on a subclass object from a subclass method.
3. Super can be used to access immediate super class instance variable.

# Assignment

Demonstrate the usage of the **super** keyword by writing programs.

1) Super can be used to call immediate super class constructor (constructor chaining).

If super is not used explicitly compiler will automatically add super as the first statement.

Super() (Constructor) call must be the first statement in a constructor

```
package superpack;

class Parent {
    Parent() {
        System.out.println("i am in super class constructor");
    }
}

class Child extends Parent {
    Child() {
        super();
        // super(); //If super is not used explicitly compiler will automatically add
        // super as the first statement.
        System.out.println("i am in child class constructor");
        // super(); //Constructor call must be the first statement in a constructor
    }
}

public class SuperClassConstructor {
    public static void main(String[] args) {
        Child child = new Child();
    }
}
```

2) super can be used to call immediate super class method on a subclass object from a subclass method.

```
package superpack;

//Superclass
class Employees {
void showDetails() {
System.out.println("Name: Arun");
System.out.println("Role: Employee");
}
}
```

```
//Subclass
class Managers extends Employees {
void showDetails() {
System.out.println("Name: Devu");
System.out.println("Role: Employee");
}
void show() {
super.showDetails(); // Calls Employee's method
-> immediate super class method
showDetails(); //calls Manager's method ->
subclass method itself
}
}
public class SuperClassSubClassSameMethod {

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

If superclass and subclass have the same method (i.e., overridden method): You can call the subclass method directly using the object of the subclass. You can call the superclass method using super.methodName() from within the subclass.

If super class and subclass not have same methods and method of super class is called from subclass method then super class method is called. There is no need of super keyword.

**3) super can be used to access immediate super class instance variable.**

**If superclass and subclass have same instance variable names:**

1. The subclass variable hides the superclass variable.
2. Accessing the variable directly in the subclass refers to the **subclass's variable**.
3. To access the superclass variable, you need to use `super.variableName`.

**If variable names are different:**

1. No conflict — you can access both directly, so no need for `super`.

```
package superpack;

class Person {
String name = "Arun";
int empCode=23;
}

class Student extends Person {
String name = "Devu";
float salary=4500.00f;

void showNames() {
System.out.println("Subclass variable: " + name); // Refers to subclass variable
System.out.println("Superclass variable: " + super.name); // Refers to superclass variable
using super
System.out.println("Superclass variable: " + empCode); //no need of super. compiler will
add super keyword
System.out.println("Subclass variable: " + salary);
}
}
```

```
public class SuperSubVariables {  
  
    public static void main(String[] args) {  
        Student s = new Student();  
        s.showNames();  
    }  
  
}
```

Difference between this and super keyword in java.

1. **this** is used for accessing variables and methods of current class.  
**super** is used for accessing variables and methods of immediate super class.
  
2. **this** is used in **constructor chaining** in current class.  
**super** is used in **constructor chaining** in immediate super class.

Thank you ☺ Happy learning ☺