**Super Keyword in Java**

* Super keyword in java is a **reference variable** that used to refer to the **immediate parent class object.**
* Whenever you make an instance from parent class which is called Subclass ,all the features from parent class is built in in sub class referred to by the super reference variable.

**Java super keyword usage:**

* super can be used to refer to the parent class instance variable.
* super can be used to call the parent class method.
* super () can be used to invoke the parent class constructor

**Java super Keyword with variables:**

* This scenario occurs when there are data members similar to the derived class and base class.
* In that case there is a possibility of ambiguity for JVM.

**Java super Keyword with methods:**

* This is used when we want to call parent class method.
* So whenever parents and children have the same designated methods, we use super keywords to solve the ambiguity.

**Java super Keyword with the Constructors:**

* Super Keywords can also be used for the original Class Constructor.
* Another important thing is that, on the basis of “super” position, both the parametric and the non-parametric constructor can call.

**Example:  program without using super keyword:**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22 | class Employee\_Java {          float salary = 10000;    }  class HR extends Employee\_Java {          float salary = 20000;          void display() {                // print current class salary              System.out.println("Salary: " + salary);        }          public static void main(String[] args) {                HR obj = new HR();              obj.display();          }  } |

**Output:**

|  |  |
| --- | --- |
| 1 | Salary: 20000 |

**Example:  program using super keyword:**

* In the above example, both the employee and the hr class have a general property salary.
* Example variable of the current class is referred to by default by example, but to refer to the parent class frequency variable, we use the super keyword to differentiate between the parent class instance frequency variable and the current class instance variable.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | class Employee\_Java {          float salary = 10000;    }  class HR extends Employee\_Java {          float salary = 20000;          void display() {              // print base class salary              System.out.println("Salary: " + super.salary);        }          public static void main(String[] args) {              HR obj = new HR();              obj.display();        }  } |

**Output:**

|  |  |
| --- | --- |
| 1 | Salary: 10000 |