When invoking a superclass version of an overridden method the **super** keyword is used.

**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.

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();    
}}    
The output:black  
white

**2. super keyword can be used to invoke immediate parent class method.**

**Eg\_1:**

import java.io.\*;

**class Person {**

int id;

String name;

void set\_data()

{

try{

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter the Id:");

id=Integer.parseInt(br.readLine());

System.out.println("Enter the Name");

name=br.readLine();

}catch(Exception ex){ex.printStackTrace();}

}

void display()

{

System.out.print(id+"\t"+name+"\t");

}

**}**

**class Employee extends Person{**

int sal;

String desgn;

void set\_data()

{

try{

super.set\_data();

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter the Designation:");

desgn=br.readLine();

System.out.println("Enter the Salary:");

sal=Integer.parseInt(br.readLine());

}catch(Exception ex){ex.printStackTrace();}

}

void display ()

{

super.display();

System.out.println(desgn+"\t"+sal);

}

public static void main(String args[])

{

Employee e1=new Employee();

e1.set\_data();

e1.display();

}

**}**

**Eg\_2:**

**class Person1 {**

int id;

String name;

void set\_data(int id,String name)

{

try{

this.id=id;

this.name=name;

}catch(Exception ex){ex.printStackTrace();}

}

void display()

{

System.out.print(id+"\t"+name+"\t");

}

}

class Employee1 extends Person1 {

int sal;

String desgn;

void set\_data(int id,String name,String desgn,int sal)

{

try{

super.set\_data(id,name);

this.desgn=desgn;

this.sal=sal;

}catch(Exception ex){ex.printStackTrace();}

}

void display()

{

super.display();

System.out.print(desgn+"\t"+sal);

}

public static void main(String args[])

{

Employee1 e1=new Employee1();

e1.set\_data(1001,"Manjeet","AP",20000);

e1.disp\_data();

}

**}**

**3. super() can be used to invoke immediate parent class constructor.**

**Eg\_1:**

import java.io.\*;

**class Person {**

int id;

String name;

public Person()

{

try{

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter the Id:");

id=Integer.parseInt(br.readLine());

System.out.println("Enter the Name");

name=br.readLine();

}catch(Exception ex){ex.printStackTrace();}

}

void display()

{

System.out.print(id+"\t"+name+"\t");

}

**}**

**class Employee extends Person{**

int sal;

String desgn;

public Employee()

{

try{

super();

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter the Designation:");

desgn=br.readLine();

System.out.println("Enter the Salary:");

sal=Integer.parseInt(br.readLine());

}catch(Exception ex){ex.printStackTrace();}

}

void display ()

{

super.display();

System.out.println(desgn+"\t"+sal);

}

public static void main(String args[])

{

Employee e1=new Employee();

e1.display();

}

**}**

**Eg\_2:**

**class Person1 {**

int id;

String name;

public Person1(int id,String name)

{

try{

this.id=id;

this.name=name;

}catch(Exception ex){ex.printStackTrace();}

}

void display()

{

System.out.print(id+"\t"+name+"\t");

}

}

class Employee1 extends Person1 {

int sal;

String desgn;

public Employee1(int id,String name,String desgn,int sal)

{

try{

super(id,name);

this.desgn=desgn;

this.sal=sal;

}catch(Exception ex){ex.printStackTrace();}

}

void display()

{

super.display();

System.out.print(desgn+"\t"+sal);

}

public static void main(String args[])

{

Employee1 e1=new Employee1(1001,"Manjeet","AP",20000);

e1.disp\_data();

}

**}**