**Multilevel Inheritance and calling of Immediate Superclass’s Methods and Constructors**

In the case of multilevel inheritance, there will be times when a subclass needs to refer to its immediate superclass. The subclass does so by use of the keyword **“super”.**

**super** has two general forms. The first calls the superclass’s (immediate superclass) constructor. The second one is used to access a member of the superclass that has been hidden by a member of a superclass.

**The first form:**

A subclass can call a constructor defined by its superclass by use the following form of **super**:-

super (args-list)

Here args-list specifies the arguments needed by the constructor in the superclass. super() must always be the first statement executed inside a subclass constructor (otherwise compilation error will be generated. Because, according to the rule, superclass constructor needs to be invoked before subclass’s constructor.

**The second form:**

The second form is used to access a data member (a public member or protected member. Because private members which cannot be inherited) or method member of the base class which is hidden (due to overriding) due to sub-class.

**Example of using two form of super:**

Import java.util.\*;

class Person

{

private String Name;

private int Age;

private char Gender;

public Person()

{

Name=null;

Age=0;

Gender=’\0’;

}

public Person(final String Name,final int Age, final char Gender)

{

this.Name=Name;

this.Age=Age;

this.Gender=Gender;

}

public void ReadData()

{

System.out.println(“Enter the name:”);

Scanner in= new Scanner(System.in);

Name=in.next();

System.out.println(“Enter the Age:”);

Age=in.nextInt();

System.out.println(“Enter the gender:”);

Gender=in.next().charAt(0);

in.close();

//closing the inputStream associated with the object in of Scanner

}

public void DisplayData()

{

System.out.println(“The Name is:”+Name);

System.out.println(“The age is:”+Age);

System.out.println(“The gender is :”+Gender);

}

}

class SalariedPerson extends Person()

{

private int Salary;

public SalariedPerson ()

{

super();

//calling of/invoking of Person class’s default constructor (taking no arguments)

//first form of super i.e. super(args-list) with no arguments

//Need to be first statement of subclass constructor’s definition

Salary=0;

}

public SalariedPerson(final String Name,final int Age, final char Gender, final int Salary)

{

super(Name,Age,Gender);

//calling of Person class’s parameterized constructor

//first form of super i.e. super(args-list) with 3 arguments

//Need to be first statement of subclass constructor’s definition

this.Salary=Salary;

}

public void ReadData()//overriding of its base class ReadData() method

{

super.ReadData();

//Second form of super which is used to call a base class method which is overridden

Scanner in=new Scanner(System.in);

System.out.println(“Enter the salary:”);

Salary=in.nextInt();

in.close();

}

public void DisplayData()//overriding of its base class ReadData() method

{

super.DisplayData();

System.out.println(“The Salary is:”+Salary);

}

class Mainsclass //driver program

{

public static void main(String args[])

{

//Implement the main of your own

}

}